Access to mathematics for visually disabled students through multi-modal interaction

Robert D. Stevens, Alistair D. N. Edwards
and
Philip A. Harling
The University of York, UK

February 19, 1997

Running head: Access to Mathematics

Robert Stevens is a Research Associate on the MATHS Project with responsibility for the design and implementation of auditory output and interaction design.

Alistair Edwards is a Lecturer in the Department of Computer Science at the University of York with research interests in multi-modal interaction, particularly in interfaces for people with visual disabilities.

Philip Harling is a Research Student who will shortly be submitting a dissertation on gestural interaction.

Abstract

Mathematics relies on visual forms of communication and is thus largely inaccessible to people who cannot communicate in this manner because of visual disabilities. This paper outlines the *Mathtalk* project, which addresses this problem based on the use of computers to produce multi-modal renderings of mathematical information. This example is unusual in that it is essential to use multiple modalities because of the nature and the difficulty of the application. In addition, the emphasis is on non-visual and hence novel modalities.

Contents

1 INTRODUCTION 4

2 RELATED WORK 7

3 FOUNDATIONS OF THE DESIGN 10

4 HOW TO SPEAK ALGEBRA? 13

4.1 Presenting Structural Information with Lexical and Prosodic Cues . . 14

4.2 Comparing the Two Presentations 18

5 CONTROLLING THE INFORMATION FLOW 20

5.1 Hiding Complex Objects 21

5.2 Elements of the Control 22

5.3 The Command Language 22

5.4 Feedback 25

5.5 Evaluating the Control 26

5.6 Improvements to the Control 29

6 PLANNING THE CONTROL OF INFORMATION: ALGEBRA EARCONS 30

6.1 Choice of Medium 31

6.2 Constructing Algebra Earcons 32

6.3 Evaluation of Algebra Earcons 36

6.4 Conclusions 40

7 THE INTEGRATED MATHTALK PROGRAM 41

7.1 Design 42

7.2 Results and Discussion 44

8 CONCLUSIONS 48

9 NOTES **49**

1 INTRODUCTION

Mathematics is a fundamental academic discipline as well as being important for employment. It is one subject which is compulsory in all school curricula. To be excluded from the study of mathematics is a major handicap since it is a pre-requisite for many other areas of study as well as mathematics itself. This paper describes the Mathtalk project which used synthetic speech and non-speech audio to explore design principles for presenting complex information non-visually.

Performing mathematics is essentially a cognitive activity, one carried out 'in the head' which suggests it should be an activity equally accessible to people with visual disabilities. However that is not in fact the case, because communicating mathematical information relies on external, visual representations (Larkin, 1989).

The objective of the work described in this paper is to develop non-visual representations of mathematics which are thus accessible to blind and visually impaired people. It has become evident that providing such representations of sufficient richness inevitably involves the use and development of multiple modalities.

Whenever two or more mathematicians gather together there will be a pencil and paper or chalkboard present. The concepts they wish to communicate are such that non-visual representations are usually not appropriate. Yet even one person working alone on a mathematical problem will use a visual form of *external memory* to keep track of intermediate results and ideas. Not having access to such representations makes the performance of mathematics very difficult and there is no simple alternative non-visual form available. Thus, few blind children study mathematics to a high level (Rapp and Rapp, 1992); the mechanics of reading and writing the notations are so difficult that they get in the way of the mathematical thinking and learning (Boormans and Cahill, 1994).

It takes only a little reflection on the part of a sighted mathematician to realize the extent of the problem; there are many different notations and levels of representation used in mathematics. The project described in this paper has concentrated on a tenable sub-set of the whole question. The notations tackled are what might generally be called *algebra*, that is to say written notations of letters (Roman and Greek and including words such as *sin* or *log*), numbers and other symbols. A typical example is Equation 1.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1}$$

In a sense, conventional mathematics has a complex visual interface. The interface is not simply static ink on paper because in order to 'do' mathematics, the person must be able to manipulate and create new instances of mathematical material such as equations, formulae etc. Mathematics is communicated almost invariably in written forms. The notations used are rich and complex – and hence very difficult to translate into other non-visual forms which will be accessible to blind people. The principle behind Mathtalk project is that in order to make such complex information accessible to blind people multiple modes of communication will have to be used and that computers make available the sorts of facilities required.

Currently blind mathematicians work with braille and/or computer-based linear textual representations. Braille has a number of limitations. Currently the most serious of these is not technical – but rather almost political. Whereas mathematics is to some extent an international language there is no one agreed international braille code for mathematics. Thus, even when braille versions of mathematical documents are available they may not be in the code known by the user. There are also practical limitations of braille. Many more symbols (cells) are required to represent mathematics and obviously it takes longer to scan through the extra symbols. Also braille is linear so that the useful second dimension of print is lost.

A common textual representation in use by blind mathematicians is Latex (Knuth, 1984; Lamport, 1985). This has the power to encode arbitrary mathematical material. In Latex, Equation 1might be rendered as:

```
x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
```

Equation 1 uses a notation which has been developed to facilitate the apprehension of its (mathematical) meaning. In a sense its Latex description embodies the same meaning (since it can be rendered into the identical print), but Latex was devised to facilitate typesetting *not* mathematical interpretation. The Latex is a linearization of the notation and hence can be generated by a conventional keyboard but in that transformation most of the 'usability' of the notation has been lost.

The development of techniques by which such notations can be transformed into non-visual forms is of interest both as a problem of presenting complex non-visual information and as an answer to a real problem of making mathematics accessible to blind people. The significance of the work thus goes beyond the specifics of the access to mathematics to informing the design of multi-media interfaces in general.

The research is aimed at transforming these visual notations into a non-visual form; it is not about devising novel non-visual representations. It may well be that there are other non-visual representations yet to be devised that might be more meaningful or accessible to blind people but it is *not* the objective of this work to explore those possibilities. The goal is to allow blind students access to the same material as their sighted peers, including the same text books. This will give the students the same references and vocabulary.

It has been decided to concentrate on the level of mathematics as studied by students at the upper end of their secondary schooling, just before they might go to college or university. This is a deliberate choice because it means that some assumptions can be made about the mathematical background of the intended users. It is recognized that ultimately comparable facilitates will be required by younger children, for without them they will never attain this level of mathematical ability, but at this stage it was safer to tackle the comparatively simpler challenge.

The advent of computers in work and education has been a mixed blessing to blind people. The early development of the personal computer was accompanied by the invention of technologies which made the computer accessible to blind people: screen reader software, synthetic speech and 'soft' braille displays (Edwards, 1991; Weber, 1994; Weber, 1995). Such solutions worked well as long as the information to be communicated was relatively simple, and textual. The advent of the graphical user interface represented a set-back, however, since the information to be communicated to the user was now more complex and inherently visual. This led to the development of interfaces making use of previously under-utilised non-visual modalities, including speech and non-speech sounds and braille. Examples are *Soundtrack* (Edwards, 1989) *Mercator* (Mynatt and Weber, 1994) and *Guib* (Mynatt and Weber, 1994). These projects share the objective of making complex visual information accessible through multiple media, but differ in the form of information.

The description of the Mathtalk program starts with a discussion of related work in presenting algebra to visually disabled readers. The foundations of the design of the user interface is built upon an analysis of the differences between visual reading and reading by listening. From this analysis emerge the twin themes of *external memory* and *control of information flow*. The following sections then break the design into four phases:

1. How to speak algebra;
2. how to control information flow;
3. how to gain an overview;
4. an evaluation of the integrated components of the Mathtalk program.

2 RELATED WORK

There has been one other significant development in the attempts to make technical material accessible via a synthetic speech presentation. The ASTER program was developed by T. V. Raman (Raman, 1994). ASTER is an audio previewer for documents written with the Latex typesetting language (Lamport, 1985). The ASTER program renders the Latex in a more human form, rather than directly speaking control sequences and other Latex symbols. Whilst not explicitly stated, ASTER is evidently aimed at mathematicians towards the upper end of the educational spectrum. This is in contrast to Mathtalk, where the context of use is in schools with pupils up to age 16.

The ASTER program has a recognizer that processes the whole of a Latex document and produces a tree that reflects the structure of the document and all its elements, one of which is a mathematical expression. Along-side this recogniser sits the audio formatting language (AFL) with which rules can be described that govern how the elements of the document are to be presented with either synthetic speech or non-speech audio. Rules can also be described for moving around the tree representing the document so that text can be read with other material making available all parts of a document, at all granularities. ASTER is not a screen reader, it is a document previewer, such as those previewers that provide a visual output from Latex, but in this case an audio presentation is given.

It is useful to describe some features of ASTER for purposes of comparison and contrast with the Mathtalk program. ASTER explicitly represents document structure, including expressions, as trees. Elements of the structure are reached by moving from parent node to child and from sibling to sibling. The following major moves are possible:

1. Go to next sibling;
2. go to previous sibling;
3. go to parent;
4. go to leftmost child;
5. go to rightmost child;
6. mark current node;
7. return to marked node.

To move from numerator of Expression 1 to the denominator, the user would have to move up to the numerator, along to the denominator and down into the sub-tree of that denominator.

More fundamentally, an expression is presented in prefix form. This means the expression $3x+4=7$ would have the equals symbol at its root, with the plus to the left and the terms $3x$ and 4 below. Thus the operators are the first objects encountered when reading an expression. The browsing moves are mapped to keystrokes based on those used in the Unix editor Vi.

Such an approach relies on the reader's ability to work with a tree form of an expression and use, reliably and swiftly, the abstract style of movement around the tree. Such an interaction may be suitable for more advanced student but Mathtalk is aimed at students less advanced in their educational careers and so takes a simpler, concrete approach to basic presentation and browsing.

ASTER represents an expression in a fundmentally different manner to that seen on paper by most school-children. In contrast, Mathtalk's layout is essentially based on that of the paper representation and would be familiar to those children using either a visual or tactile presentation. A tree representation is likely to be unfamiliar and difficult for most school-children and without evaluation it would be dangerous to assume ASTER's approach is usable in such a context. It is also important to note, that when visually disabled children are integrated into mainstream education, notations need to be shared for co-operative work to take place.

The AFL can be used to describe how an expression is to be spoken. A promising technique described in ASTER is that of variable substitution. An AFL rule can be made that substitutes one part of a complex expression with a label, that can be rendered at a later point. For example, the expression

$$I = \int_0^\infty e^{-x^2}\,dx \tag{2}$$

is rendered in full as

> 'I equals the integral from zero to infinity of e to the negative x squared with respect to x'.

but by substituting the integrand with z ASTER can render the expression as:

> 'I equals the integral from zero to infinity of z with respect to x, where z is . . . '

This enables the listener to gather an overview of the expression and then obtain the detail of the integrand after this overview. It remains a moot point whether such a rendering gives too much information, negating the effects of the substitution. As described below, Mathtalk uses a method of hiding complex information so that information flow can be controled and an overview of an expression given. This approach is simpler and less flexible than ASTER's, but has been proven to be useful.

Both ASTER and Mathtalk use prosodic cues in the synthetic speech to denote structure within an expression. ASTER uses the prosodic cues described by O'Malley and colleagues (O'Malley et al., 1973) and Streeter (1978). Mathtalk uses thesame rules, but extends and rationalizes them with a further study of spoken mathematics.

The Mathtalk project also investigated whether such prosodic cues worked with a synthetic speech presentation and had any effect on the usability of spoken mathematics.

ASTER and Mathtalk have tackled two ends of the same problem and have had different goals in mind during their development. ASTER is a complete system for reading technical documents. Its principle achievements have been to develop a recogniser that can transform Latex input to a data-representation suitable for audio rendering. The audio formatting language was developed to describe how such a representation can be rendered.

Mathtalk was developed to explore methods of presenting complex information, of which algebra notation is a particularly fine exemplar. The object was not to build a complete system, suitable for everyday use. Instead, Mathtalk was used to explore and evaluate designs that enable a passive listener to become an active reader. Ideally, the design principles obtained from the Mathtalk program could be used in the AFL to deliver a method of presentation and control that is usable by all levels of mathematics students. As will be seen below, the emphasis in the Mathtalk program has been the development of principles to guide the design of such systems through evaluation.

An alternative approach to the non-visual representation of mathematics is being developed by Gardner in the form of *DotsPlus* (Barry and Lundquist, 1994; Gardner and Barry, 1993). This is a printed tactile representation. It combines conventional (but tactile) symbols (operators, fraction lines etc.) with braille (for letters and numbers). For instance, in Equation 1 the symbols *a, b, c, x, 2* and*4* would all be represented as braille, whereas the symbols =, – ± and the fraction line would all resemble their printed form – but enlarged and printed in raised ink which can be read with the fingers. At present DotsPlus can only be printed statically; the technology does not exist to generate it interactively and therefore there is no possibility of incorporating it in an interactive workstation.

3 FOUNDATIONS OF THE DESIGN

The aim of the design of the Mathtalk program was to transform a passive listener to an active reader of complex information. The basis of the design arises from a view of the fundamental differences between visual reading and listening to complex information. For our purposes, the essential features of visual reading are the printed page and the selection afforded by the visual system.

The printed page acts as an external memory (Schönpflug, 1986) and thus relieves the reader of the burden of retaining a large amount of complex information. The visual system, in combination with the external memory, allows fast and accurate control over what is selected to be read (Rayner and Pollatsek, 1989). It is this control that makes visual reading active.

The external memory and fast, accurate control over information allowed by the visual system mean that this modality has a high bandwidth. The lack of these features and the corresponding lower bandwidth mean that control becomes even more significant in listening. The Mathtalk project has concentrated on maximizing the use of the available bandwidth while at the same time giving the user a high degree of control.

The form of the information on the printed page helps this process of fast and accurate selection. In the case of algebra notation the order of precedence is instantiated in typographic rules (Kirshner, 1989): Least precedence operators are surrounded by a large amount of white space; multiplication and division are represented by horizontal and vertical juxtaposition; exponentiation by diagonal juxtaposition and other visually obtrusive markers divide information into groups. Such spatial arrangements are found to help many people to correctly parse an algebraic expression (Kirshner, 1989).

Many of these features are in direct contrast to the situation of the listening reader for whom there is no usable external memory. What is spoken has to either be retained in the listener's internal memory or it is lost. A listener can retain the gist of an utterance, the surface structure being lost (Ellis and Beattie, 1986). This is usually acceptable for everyday conversation and listening to plain text in synthetic speech. However, as algebra notation is concise, lacking redundancy, loss of any of this information can be catastrophic.

The listener does not control which part of the information is to be heard. When there is control, for instance with a tape recorder, the control is so slow and inaccurate that it is almost useless. This lack of control makes the listener passive and this passivity often leads to lapses of concentration (Aldrich and Parkin, 1988), which leads to a greater need for control over the flow of information. Short-term memory is easily overloaded leading to loss of information, increased mental workload and a lack of cognitive resources to be focused on the comprehension task itself.

These problems are exacerbated by the type of information in algebra notation. It is dense, concise and the loss of any information transforms the meaning of that information. The notation is often complex, due to the larger variety of symbols, and structural complexity that can rise to an arbitrary degree.

From this description of the problem two basic themes can be drawn out for the design process:

- Compensation for the lack of external memory and
- the provision of fast and accurate control over information flow.

If these two themes are successfully addressed the passive listener should be transformed into an active reader.

The development of Mathtalk proceeded in four phases, reflected in the sections below:

- The first question was how to speak algebra notation in order to compensate for the lack of external memory. A traditional approach of adding lexical cues to delimit structure was compared to a method of inserting prosodic cues to accomplish the same task.
- The second question was how to control the information flow. This was achieved with a structure-based browsing language mediated by a command language derived from spoken commands.
- The third question was how to gain an overview of the information in order to plan the reading process. A novel non-speech audio glance, developed from the prosodic component of speech achieved this goal.
- Each of these components of the Mathtalk program were evaluated separately and amended as appropriate. The final phase of the development was an evaluation of the integrated Mathtalk program.

From this process of design and evaluation a set of principles for the design of audio tools to enable active reading of complex information were derived.

Mathtalk is implemented as a program running on PC-compatible computers under MS-Dos. It uses the API Multivoice speech synthesizer and SoundBlaster or Crystal River Beachtron music synthesizer.

4 HOW TO SPEAK ALGEBRA?

The first question that must be asked in the development of a user interface to read algebra notation is *what* information must be presented. Only then can the designer address the problem of how to speak that information. Algebra notation is used for the communication and manipulation of mathematical concepts, for a single user and between individuals. The user interface therefore must afford the same purpose. Within the wider area of presenting mathematical ideas what information is present in the notation itself and what information is brought to the presentation by the reader influences the design of the user interface.

If the missing functionality of external memory and control of information flow can be replaced it is probable that blind people are as capable as their sighted peers of bringing the same resources to learning and doing mathematical tasks.

Standard algebra notation does not present the mathematical semantics of an expression. The manner in which $ax^2 + bx + c = 0$ is displayed does not explicitly inform the reader that it is a quadratic equation. The presentation may, however, help the reader to decide that it is a quadratic expression. It is part of the reading process that the reader brings his or her mathematical knowledge to bear upon the information presented to decide that it is a quadratic equation.

The symbols $2x^2$ may either be correctly parsed as $2(x^2)$ or incorrectly as $(2x)^2$. The presentation displays the grouping of the symbols unambiguously, but does not indicate the meaning of that positioning. This principle should also apply to the speech presentation. It should enable parsing, but not explicitly indicate the semantics of the grouping. So, the display should present the grouping and association of objects in the expression, but not indicate the meaning of that positioning and not indicate any deeper mathematical meaning of that presentation. A consequence of this design decision was that the Mathtalk program would not ‘read to’ a blind person, but that the blind person would do the reading, that is, would become an active reader.

To avoid indicating deeper mathematical meaning in speech is easy. To only display grouping of symbols, the manner of grouping, without indicating some of the syntactic meaning of that grouping presents some problems. The symbols $2x^2$ may be spoken in a variety of ways: For instance, as 'two x squared' or 'two x superscript two'. These renderings span a range of added meaning. As will be seen later a global principle of minimal semantic interpretation is applied. However, the English language often lacks neutral words for constructs and use of unfamiliar words will have a concomitant effect on usability. Thus compromise is sometimes needed so that the best form of presentation is used.

4.1 Presenting Structural Information with Lexical and Prosodic Cues

Now that there is a principle to guide *what* information to include in a spoken display, the design of *how* to present that information can proceed. During the design of Mathtalk two options for presenting structural information were evaluated:

1. A traditional method of inserting lexical cues to delimit ambiguous groupings in an expression.
2. The use of prosodic cues to delimit the same structures.

Chang (1983) developed a set of rules for inserting lexical cues into a spoken expression. For example the utterance 'one plus two over three plus four' has four different possible parsings. Inserting the lexical cues 'begin fraction' and 'end fraction' can make explicit which parsing is intended: 'one plus the fraction two over three plus four end fraction'.

The large variety of options presented by Chang cover a spread of interpretations of mathematical intention of an expression. These rules were refined and adapted, to comply with minimal interpretation and avoid lingering ambiguity, for implementation in the Mathtalk program. With these rules, Expression 1 would be spoken in the following manner:

> 'x equals the fraction minus b plus or minus the root of b super two minus four a c denominator two a'.

Two principles emerge from these rules:

- The notion of *simple* and *complex* structure can be used to guide the insertion of lexical and other cues. When the structure is simple, the symbols can be spoken unadorned with lexical cues. Complexity arises when more than one term is grouped together, by either spatial cues or parsing markers. In such cases grouping ambiguity arises and lexical cues are inserted to avoid confusion.
- This has the advantage of reducing the number of lexical cues, giving a principle of minimum speech and maximum information.

As can be seen from the example above, the lexical presentation can make the structure of an expression explicit. It does, however, have the side-effect of vastly increasing the amount of spoken material. Such a presentation may overwhelm already stretched short-term memory resources. In addition, the lexical cues may obscure the contents of the expression. The *suffix effect* (Baddeley, 1992) implies that words appended to a spoken list will overwrite the most recently heard items in that list. This would suggest that lexical cues terminating a structure would overwrite some of the contents of that structure in a listener's memory.

A further problem with using the method above lies with the synthetic speech presentation. All speakers know that they can lend an utterance meaning above that simply contained in the words. The utterance, 'the last time we met Robert was horrible', can be given two different meanings simply by moving a pause before or after the word 'Robert'. Similarly changing the pitch profile or emphasis within an utterance can drastically change the effect of that utterance.

These effects of pausing, rhythm, pitch and stress are collectively known as *prosody* and can be thought of as the non-lexical information content of speech. Two important roles of prosody are of interest in the design of the Mathtalk program. The first is prosody's ability to indicate syntactic structure. Prosodic cues are highly correlated with clause boundaries in speech (Crystal, 1975; Beech, 1991; Streeter, 1978) and listeners can use such cues to recover syntactic information from speech. The second role for prosody is to increase the memorability of speech (Baddeley, 1992; Garnham, 1989). The chunking of information into significant sub-units and the rhythmic component of speech are thought to aid retention of speech in short-term memory (Baddeley, 1992).

To a great extent, these features are absent from synthetic speech. Most commercially available speech synthesizers lack any rhythmic component and prosodic effects for structure can only be added where punctuation marks allow. In many cases, neither the semantic nor the structural content of a sentence is available that would allow the insertion of such cues. The lack of these cues could exacerbate any problems with the lexical cue presentation outlined above: A verbose stream of relentless speech is more likely to overwhelm a listener when it lacks any prosodic cues.

However, if the structural information is present and a set of rules exist for the prosodic presentation of those boundaries, there is the prospect of allowing dynamic insertion of such cues into a synthetic speech presentation of algebra. As described above, the role of algebra notation is to present that grouping information unambiguously. By making the spoken form easier to remember and improving the recovery of structure, some of the qualities of an external memory can be given to the audio presentation.

Rules for Algebraic Prosody

Some significant work on the use of prosodic cues in algebra already exists. O'Malley et al. (1973) derived a set of prosodic rules for simple algebraic expressions. They found that pauses were highly correlated with syntactic boundaries at operators, parentheses, fractions and superscripts. Listeners were reliable in recovering such information using these cues alone and the rules could reliably predict where cues should be inserted in a spoken expression. A further study was carried out by Streeter (1978). Using a set of short expressions she found that listeners reliably used the cues of pitch, duration and amplitude to recover structural information from spoken algebra.

These studies indicate that rules for algebraic prosody exist and that listeners can use these cues in human speech to recover structural information. Again, these cues were highly correlated with the structure of an expression, suggesting that as the structure is explicit, such cues could be dynamically inserted into a machine generated algebraic utterance. The question remained as to whether such information, when generated by a speech synthesizer, can be used in the same way by listeners.

The rules described by O'Malley and colleagues and Streeter were not sufficient for the purposes of the Mathtalk program; they did not cover a wide enough range of algebra and were not varied enough in length and complexity to give a sufficiently rich set of rules.

To confirm and extend the rules for inserting prosodic cues into spoken algebra a short study was carried out as part of the development of the Mathtalk program[1]. Two experienced speakers of mathematics (a school-teacher and a university mathematician) were given a set of 24 expressions to speak. These expressions were in contrasting pairs. For example:

$$3x + 4 = 7 \tag{3}$$

$$3(x + 4) = 7 \tag{4}$$

$$y = x^n + 1 \tag{5}$$

$$y = x^{n+1} \tag{6}$$

were two contrasting pairs. By keeping the lexical content similar, but changing the structure the effects on prosodic cues would be more apparent. The speakers were recorded in two separate sessions, the expressions being shuffled for each recording. The recordings were then analysed for pausing, pitch contour and amplitude patterns.

Details of the rules derived and the analysis can be found in Stevens (1996). Figures 1, 2 and 3 show the effect the insertion of these cues on the spoken form of three expressions presented by the Mathtalk program. The division into terms by pauses, the pitch contour within terms and the marking of terms by amplitude can be seen.

[Figure 1 about here.]

[Figure 2 about here.]

[Figure 3 about here.]

[1]These measurements were carried out with the invaluable help of Professor John Local and students of the Department of Language and Linguistic Science at the University of York, UK.

4.2 Comparing the Two Presentations

An experiment was performed to test the efficacy of each presentation style. It was hypothesized that prosodic cues could be used to recover as least as much structural information as could the lexical cues. Also, by avoiding the suffix effect and by chunking the information, the prosodic cues would enable a higher degree of an expression's content to be retained. Finally, it was hypothesized that the prosodic cues would reduce the mental workload associated with the task.

EXPERIMENTAL METHOD

Two groups of twelve sighted, normally hearing participants were used. Sighted participants were used because of availability and experimental practicality. During the component evaluations a reasonable assumption of equivalent cognitive profiles of sighted and visually disabled listeners was made. For the evaluation of the final Mathtalk program such an assumption would not be valid.

One group heard expressions with lexical cues then prosodic cues (LP group). The second heard the same group of expressions with lexical cues, then the set of expressions used for the prosodic condition with neither lexical nor prosodic cues (LN group). A direct recall task was used. An expression was heard once and when the speech had finished the participant was asked to write what he or she remembered of the expression. The participant was asked to use an ellipsis or question mark to indicate forgotten items.

A NASA Task Load Index (TLX) workload assessment (NASA, 1987) was used to provide a subjective rating for the task workload. Reduced mental workload is an important facet of the usability measures of efficiency in terms of human resources and the user's satisfaction with the system. Participants had to give quantitative ratings for five load factors, such as mental demand, perceived performance level and effort expended. After the second condition in each group, the participant was asked to quantify these measures relative to the first assessment. Finally, the participant's overall preference between the conditions was recorded.

Two matched sets of 12 expressions were presented. All expressions contained one or more fraction, parenthesized sub-expression or superscript. In the lexical first set the scopes of these complex items were delimited using the lexical cues, as described earlier. The second set had these boundaries indicated by using only the prosodic cues. A second version of this set was prepared, with neither lexical nor prosodic cues (the no-cues condition). A Berkeley Systems Best Speech synthesizer was used to speak the expressions through external loudspeakers.

Results and discussion

The answers were marked separately for recall of structure and content. For an expression to be correct overall, it had to have both the gross syntactic structure and greater than 75% of the content correct. Figure 4 shows the mean percentage of correct recollections for these factors and the overall scores for each condition in each group. T-tests were used to test for a significant difference between the means and for difference between the workload factors. Percentage differences were also calculated for the TLX factors.

[Figure 4 about here.]

As can be seen from Figure 4, participants were able to recover more structure and content from expressions heard with prosodic cues than lexical cues, and thus performed better overall. These differences were significant when analysed by t-test at the 95% confidence limit.

[Figure 5 about here.]

All the factors in the TLX were significantly different (as measured by t-test) in favour of the prosodic condition, indicating that algebra expressions spoken with prosody are easier to listen to than those spoken using lexical cues alone (see Figure 5). Frustration was 22% higher in the lexical condition corresponding to participants' frequent comments that the lexical cues were intrusive and overwhelming. Mental demand was higher in the lexical condition (12%), but effort expended was not so different (8%), but still significatly different perhaps reflecting that even with prosody the task is hard work.

Participants showed a distinct preference for the prosodic condition. This was measured using a scale presented after both conditions: Zero indicating preference for condition one, 20 for condition two and ten no preference. The mean bias towards the condition with the prosodic presentation was six and three for the lexical condition. This bias was significantly in favour of the prosodic condition.

This experiment demonstrated that the addition of prosody to spoken complex information could make that presentation more usable. Structure can be indicated without increasing the lexical load; the presentation improves the retention of the lexical content and decreases the mental workload associated with the task. This gives the spoken presentation some of the qualities of an external memory. Thus, the use of prosody to increase the usability of spoken complex information forms one of the design principles in the Mathtalk program.

5 CONTROLLING THE INFORMATION FLOW

Simply improving the spoken presentation of algebra notation is not enough to allow visually disabled people to read by listening. The display has been much improved, but prosody did not solve all the problems of reading by listening. No matter how good the presentation, the listening is still passive and error prone.

Control of focus of attention and granularity of view could further facilitate apprehension of structure, allowing a large expression to be broken down into manageable units. Most importantly it would allow certain parts of the expression to be reviewed with speed and accuracy. Giving readers control should make them active and such access should relieve them of the burden of remembering all the material. Instead, the listener could use the display as a memory, thus freeing cognitive resources.

The aspects of understanding in reading and decisions on how to gain that understanding are deemed to be best left to the reader. The Mathtalk program only attempts to offer the information for reading in the best manner possible in the auditory mode. Readers obviously vary in how they extract information and use of strategies to achieve mathematical goals. So the Mathtalk program should not prescribe how the reader should tackle a task. The design of the browsing was to offer a series of moves that the user could develop into higher-level tactics, stratagems and strategies (Bates, 1989).

A structure-based browsing was chosen to enable active reading within the Mathtalk program. The main problem in designing such a language is to make it rich enough to provide the user with the pinpoint control required while not making it so complex that it is unusable. Browsing structure gives a suitable task-based mechanism for reading.

5.1 Hiding Complex Objects

A major feature of Mathtalk's presentation style was the hiding of complex objects. Any object that groups more than one term together, by either parsing mark or spatial location is folded-up and referred to only by its name during browsing. For example, Expression 1 would be represented as three objects: 'x', 'equals' and 'a fraction'. Such a mechanism allows greater control over information flow than that given by browsing alone. In addition it should facilitate disambiguation of grouping. For example, as a reader moves along the base-level, he or she will find that everything except the x and $=$ are within the fraction. Whilst prosody can disambiguate an utterance, such a presentation confirms the structure in a quick and easy manner.

The added control comes from the amount of speech that is given at any one time. Rather than speaking the whole of a fraction on moving to that object the reader is only informed of the nature of that object. He or she can then choose to hear all or part of that object, without moving into the object. The user may also easily skip over that object in a single move, rather than having to move through all of its contents.

As objects are hidden, the reader must be able to move into and out of those objects in order to browse. Speaking the contents of a complex object would utter the simple contents in full, but still hide complex objects at a lower level.

As with the basic speech presentation shown in Section 4, the question of what information should be spoken during browsing must be answered. The principle of minimal interpretation was invoked. This makes the full utterance and the finer grained browsing consistent.

5.2 Elements of the Control

The aim was to allow the user to visit all parts of the expression with speed and accuracy. The actions of moving to the **next** and **previous** objects are intuitive parts of any structure-based browsing language. The actions of moving **into** and **out-of** hidden objects is a natural consequence of that design.

For an auditory system an extra facility has to be added that is not needed in a visual system. This is the notion of **current**. In a visual display the current selection is indicated by some means in a persistent fashion (i.e. by highlighting in reverse video). The auditory display is transient, so the current selection or focus of attention also disappears. So the action **current** is needed to give this focus.

The moves are often small scale and local. Control will also involve larger scale shifts in the focus of attention. These can be adequately captured by the moves to the **beginning** and **end** of an object. These common moves can form the basis of the browsing through the algebra expression.

5.3 The Command Language

This control needs to be mediated externally, rather than by eye movement and focus, as occurs in a visual interface. In this case the control, given by browsing functions, will be mediated by a command language expressed on the computer's keyboard. An algebra expression might have a rich structure so any command language to manipulate the necessarily large number of browsing functions will itself be large and complex. The design questions are how to:

1. Cover the wide range of potential structures;
2. provide this coverage in a reasonably learnable, predictable manner;
3. give the speed of control to complement the inherent accuracy of structure-based browsing;
4. make the reading active, without disrupting that process;
5. provide feedback about reading moves made, progress of the reading, errors made and general orientation information;

6. maintain reading as the primary task;

7. enable moves to be combined into tactics and strategies.

The moves and the objects on which they act were formed into a simple command language that would cover the necessarily complex nature of browsing around an algebra expression. The browsing language was developed from the names of the moves and objects themselves. Combining a move or *action* with an object or *target* forms a command that falls naturally into a spoken form. For example ‘beginning of expression’, ‘next term’ and ‘previous character’ emerge easily from the set of actions and targets as intuitive commands.

Figure 6 shows the action and target words used in the browsing language. An action word was combined with a target word and mnenomically mapped to the keyboard. Thus, **nt** invoked the move **next term**. The actions were grouped together semantically: **current**, **next** and **previous** fall together, **into/out-of** and **beginning/end** were intuitively paired. This grouping should make the actions easier to learn.

[Figure 6 about here.]

There is a need to be able to speak the contents of a complex object without moving into that object. The action **current** cannot do this task. For instance, **current item** when on the hidden object ‘a fraction’ would only utter that object’s name. This was part of the functionality of the hidden objects described earlier. **Current fraction** could be used within a fraction to speak the contents of that fraction. The same action cannot be used for both tasks. It is possible that ambiguity could arise if **current** was used for both (if there were nested fractions). So another action, **speak** was used to utter the contents of a complex object while the focus was ‘on’ that object rather than within that object.

From a small number of action and target words a very large number of commands can be generated. Very few combinations did not generate valid commands. Some are present for the sake of completeness and to reduce the number of errors. It is important to note that not all the commands need to be learnt in order to successfully browse an expression. It was hoped that simply by learning the relatively small number of command words suitable commands would be generated by the users themselves.

This design gave consistent generation of a large set of commands. All browsing commands were two-letter sequences. Thus the command language already has one level of consistency. A further level of consistency was gained from the style of browsing itself, which was consistent with directing a human reader and fell naturally into a spoken form. The relatively small number of words and potential familiarity of style could make the language learnable.

The command **current level** emerges as interesting. This command utters all simple objects in full, but hides any complex objects. This means, that by asking for the **current level** the reader may get an 'overview' of a complex object. For example, Expression 1 would be rendered as 'x equals a fraction' by this command. By using this command the reader exercises control over the amount of information he or she has to process and in addition has the structure confirmed.

As well as the unconstrained browsing, a default browsing style was designed. A single command could be given to reveal the expression a chunk at a time. This would give the reader the opportunity to move through an expression, from left to right building up a representation of the expression in a controled manner without having to think of or issue any other commands.

An expression is unfolded term-by-term. The term was chosen as it formed the basic unit of spoken algebra and is a commonly manipulated unit within algebra. When a complex object is encountered within a term, the unfolding stops at that point with an indefinite rendering of that object. So a fraction is simply referred to as 'a fraction'. On the next command, that complex object is unfolded term-by-term. Expression 1 is unfolded, on each cue from the user, as:

1. x;
2. equals a fraction;
3. numerator minus b;
4. plus or minus the root of a quantity;
5. the quantity b super 2;

6. minus four a c;

7. denominator two a.

5.4 Feedback

Three types of error are possible when using the command language:

1. First keystroke error. A mnemonic for an action not appearing in the language was used.

2. Second keystroke errors. A non-existent target or target not usable with the accepted action was issued.

3. Inappropriate command. A well formed command was issued, but not one that could work in the current context. An example of this would be to issue the command **into fraction** when the focus of attention was not a fraction object.

A simple system of non-speech audio messages, using the PC speaker, was used to indicate these errors. One beep for a first-keystroke error; two beeps for a second keystroke error and three for an inappropriate command error.

A system of non-speech messages was used to indicate the beginning and end of levels or complex objects within the expression. These terminus sounds were given as descending and ascending C-major chords. A start was indicated by a rising sound and the end by a falling sound, to be consistent with the use of falling pitch through the algebraic utterance.

As the reader reached the end of the level a tone would be heard. Attempts to move past the boundary would cause repetition of the terminus sound and the final term. Tones were played after the term spoken at the terminus of the level. The speech and non-speech was presented serially to avoid any masking of the information in either audio source (Crispien et al., 1994). This was meant to place the level within 'sound brackets'. A non-speech message, rather than a spoken message, was designed to avoid any potential suffix effect as observed with the lexical cues in the speech presentation investigations in Section 4.

5.5 Evaluating the Control

The co-operative evaluation method (Monk et al., 1993) was used to assess the usability of the browsing functions and command language. A set of tasks were designed for the system being evaluated and the user asked to perform these tasks. During this process the user was asked to ‘think aloud’, to say what he or she was doing and why. The participant was also encouraged to interact with the experimenter. This gave a rich source of information about the usability of the system, that cannot be captured by simple quantitative measures alone.

Monk and colleagues have described how relatively few participants can be used to capture the majority of the usability problems in an interface and as many cycles are used, together with re-design, a usable interface can be developed rapidly. Only the final cycle of evaluation of Mathtalk’s browsing is reported below. The command language, in particular, had changed over the course of evaluations. As a result a wider set of browsing functions was available and the command language had changed from a simple cursor-key based paradigm to the one described above. In the present instance of the command language, the cycles of evaluation had improved the command words used, improved feedback and increased functionality. Importantly, different modes used for different styles of browsing had been removed.

The purpose of this evaluation was not to demonstrate that enabling greater control was the right design decision, but whether this form of control was usable and performed the task for which it was designed. That is, could the command language deliver control over what is to be spoken by the Mathtalk program?

Six fully sighted participants were used in the evaluation. All were computer literate graduate students with no significant experience with speech synthesizers.

Ten expressions were prepared for the experiment. These ranged from syntactically simple to more complex expressions. The range of complexity was similar to that seen in mathematics curricula, at the 16 to 18 years age group, for example see (Bostock and Chandler, 1981). Some of the expressions and associated tasks can be seen in the examples of reader-Mathtalk dialogue given below.

Results and Discussion

This evaluation demonstrated the general usability of the browsing. The participants were able to use the command language and the browsing functions to accomplish the tasks and demonstrated a high degree of control over the information flow. The evaluation was also able to demonstrate some flaws in the design and point towards solutions.

All participants successfully learnt the commands, often uttering the two letter command derived from the experimenter's speech. As the commands matched the tasks, simply extracting the command, taking the mnemonic mapping would complete the task successfully.

The default browsing strategy was widely used. Figure 7 shows that one-fourth of the total commands issued by all participants were for the default browsing style. The range varies from 16% up to 30%.

[Figure 7 about here.]

Three commands were prominent in cases of strategy development:

- **current level**;
- **beginning expression** and
- the default strategy.

Users took advantage of the **current level** command to speak the base-level amount of speech and giving an overview of the expression. This command acted like a glance at the overall structure of the expression. This glance was an emergent property of the hidden objects. This view was used instead of speaking the whole expression (see Figure 7).

That some signs were seen of tactic development suggests that the control component fulfils this part of its role. The method of hiding complex items was seen to be useful, as demonstrated by the use of **current level** instead of **speak expression**. Figure 7 shows the proportions of **current level** and **speak expression** commands used during the evaluations. The hiding of complex objects reduces the amount of speech generated and emphasizes the structure of complex expressions. This probably accounts for the disparity in usage of the two commands.

The hiding of complex objects facilitated the development of the major glance and read strategy. Thus the hidden objects formed a major part of the users' ability to control the flow of information. The following dialogue gives an example of the utility of the hiding of complexity to find the difference between two lexically identical expressions. In the following descriptions each item is marked with **F*n*** for participant's speech, otherwise commands are emboldened at the start of each item. Typewriter typeface is used for synthetic speech.: 'What are the differences between expressions five and six?'

ne `expression five.`

cl `x super n plus one equals y.`

ne `expression six.`

cl `x with a superscript equals y.`

'The second has a complex superscript.'

No users explicitly requested the need for a mute function to terminate spoken output. The lack of a need for a mute may be a consequence of the fine control that users had over the amount of speech being used and justifies the design for control with speed and accuracy. The reduction of speech to a minimum, and not automatically speaking any of the expression, seemed to reduce irritation and frustration in the participants. Very few errors were made by participants when issuing commands. The overall error rate was 3.3%. Errors were taken as commands with either an erroneous first or second keystroke, together with those commands given in an inappropriate context (for example, **next fraction** where no fraction exists). Those commands that might be judged to not further the achievement of goals were not included.

The terminus sounds (described in Section 5.4) were appreciated by all the users, each of whom commented on their usefulness. Both beginning and end sounds were used to confirm location.

Participants made several useful comments on how the terminus sounds could be improved. Participant C4 noted that the sounds were useful, but that the sounds were overloaded, that is the same sound was used to terminate all complex objects including the whole expression. On some occasions participants would assume the end sound of a nested complex object was the end of the whole expression when there was more to read. Participant C5 adopted the tactic of trying to move on from the object that gave the end sound, so that if there were more to read attention would move to that object.

5.6 Improvements to the Control

The following improvements to the functionality and the command language were made:

1. When moving between expressions the pointer was placed at the start of the new expression, rather than resuming any previous position held in that expression.
2. The command language was made consistent within the expression. This meant changing the functionality of **current expression** to speak the whole expression. Accordingly an extra command **which expression** was introduced to give the number of the expression being read.
3. The name of the **speak** action was changed to **show** to avoid confusion with phrases encompassing commands. The intra-expression consistency meant that **show** only caused the contents of complex objects to be spoken and no longer uttered the whole expression. This further increased consistency.
4. The terminus sounds were made consistent with the timbres used in the audio glance (see Section 6). The terminus sound at the end of the whole expression was repeated to add this information to the general end sound.
5. The start sounds were played before the spoken object to 'bracket' the objects with terminus sounds.
6. An attempt was made to ensure the default reading style no longer repeated the current object when another command had been used.
7. The keystroke error sound was made more meaningful by replacing the PC speaker beep with a typewriter sound, linking it to the keystroke.
8. Users could recover from a keystroke error by using the `backspace` key.
9. A mute function was added that not only terminated all speech, but cleared any remaining keystrokes in the buffer.

10. The action **current** was redesigned to have a widest possible scope with complex objects, to enable higher levels to be spoken from within nested objects.

11. In a similar manner to **current**, the **out-of** action could now act upon any complex target at the present or higher scope, to avoid the necessity of ‘climbing out’ of nested complex objects.

12. When speaking a term the operator to the left was spoken, to avoid confusion when moving backwards.

This and earlier cycles of evaluation has demonstrated that the browsing functions and command language give the control over information flow for which they were designed. Accuracy is inherent in the structure based browsing and the language allows all parts of the expression to be visited. The co-operative style of evaluation meant that many design flaws could be found and amended rapidly.

6 PLANNING THE CONTROL OF INFORMATION: ALGEBRA EARCONS

A scan or glance is proposed as the first stage in the reading of an expression (Ernest, 1987). A glance gives information about overall structure, complexity and can give the reader expectations about the expression. In addition the reader may review the expression for any unknown or difficult symbols. This idea is supported by Larkin (Larkin, 1989) who suggests the form of the expression on the page prompts the reader to decide on the type of the expression and potential solution strategies.

Such a glance is usually not available to a blind reader. With a spoken presentation it is not possible to take an abstract or high-level view and reading is usually reduced to a bottom-up process of integrating a series of symbols that have been heard in a temporal, ‘left-to-right’ manner. The only way to ascertain the nature of the expression is to repeatedly read the expression in full, retain and then integrate the information.

The following definition was used as the basis for designing a glance: A glance is a rapid, high-level view or abstraction that contains the salient or relevant information in the environment, pertinent to the current task.

For the reader, the task to be accomplished with a glance at an algebra expression is to assess the nature of that expression in order to plan the reading. The glance has to enable the reader to judge the structural complexity of that expression: Ranging from a simple idea of length to a full frame work that only lacks lexical detail. To do this the glance should contain information about the presence of certain types of object, their relative location and the size of those objects. This information must be presented in a manner that allows the reader to rapidly extract information at the levels described.

To achieve this, the glance designed within Mathtalk is one at the *structure* of an expression. This is consistent with the notion that the main purpose of the display is to present the structure or grouping within an expression and that the browsing is movement around that expression.

6.1 Choice of Medium

The audio glance could use either synthetic speech or non-speech audio. The chosen medium had to fulfil the following criteria:

- Rapidity;
- presence of type, but not instance, of object;
- location of objects and
- relative size of object.

A full spoken utterance could be used and the listener left to extract the information salient for the glance. Such an approach was seen to be difficult in Section 4.

A structural description could be given. For example, ‘a single operand equals a fraction with a large numerator and short denominator’ would describe Expression 1. Such a description is itself long and complex and may involve too much decoding. The utterance ‘the equation has three terms’, says nothing about the size or nature of the terms or the balance of the expression. A richer description, such as ‘three terms, the first has two operands and a superscript . . . ’ contains the right information for a glance but is too long. Another method would be to use a mathematical description as a glance, for example, ‘a quadratic’. The description ‘quadratic’ accurately describes many expressions.

Compressed synthetic speech could also serve as a glance. *SpeechSkimmer*, developed by Arons (1993), uses speeded up recordings of natural speech that retain many of the prosodic cues that indicate document structure. Speeded up speech, which retains structural cues such as division into terms and the grouping of objects into complex items would fulfil some of the criteria for a glance. The only information lacking would be on the type of the object being represented. Speeded up speech, especially in the system developed by Arons, could allow a gradually increasing range of detail to be exposed in a glance. It remains to be seen whether speeding up synthetic speech has the same effects as speeding up natural speech and whether prosodic cues remain usable with relatively short utterances.

The alternative to synthetic speech is non-speech audio. The association of non-speech sounds with an expression can capture many of the properties of the glance defined above. It would be difficult to describe the detail of an expression in sound, for example, the instance of a particular letter. Instead different musical timbres could be associated with different classes of object in an expression to give the type of an object. In this way sound can give the abstract function of a glance. The criteria of indicating type of object and location of object can now be fulfiled. The delivery of non-speech audio can be rapid. It should be possible to play a short sound in order that the listener can recognise the associated object type, where the spoken form may be much longer.

6.2 Constructing Algebra Earcons

One method used to add non-speech audio sounds to the computer-user interface is the *earcon* (Blattner et al., 1989; Brewster et al., 1994). Earcons are abstract structured sequences of non-speech audio used to give messages in the computer interface. The audio glance developed for Mathtalk takes advantage of the structured nature of earcons to develop a new type of prosody based earcon called an *algebra earcon* whose structure reflects that of an expression.

Prosody can indicate the structure of an utterance, but the speech signal also carries the lexical detail of the expression. The requirements for the audio glance would be fulfiled by presenting the listener with prosody without the lexical detail, that is, prosody without the speech. Earcons and prosody share the same parameters of rhythm, pitch, amplitude and timbre. Earcons deliver their message through the structure defined by these parameters. This affords the opportunity to present the prosodic structure of an utterance as an earcon and thus avoiding the lexical detail of speech. The main intention of the audio glance is to replace the lexical detail while retaining the structure presenting properties of prosody [2].

Different objects within an algebra expression are replaced with sounds with different musical timbres, enabling a listener to discriminate elements within the expression without knowing the instance. The sounds used are shown in Figure 8. The timing, pitch and amplitude of these sounds are then manipulated according to the prosody based rules below.

[Figure 8 about here.]

A priority is to establish a rhythm by which a listener could group items together, discriminate elements of structure and aid retention. Overall rhythm is important. In spoken algebra the term forms the *foot* or basic unit of rhythm in the utterances. The foot is the equivalent of a bar in music (Halliday, 1970). The bar length for an earcon is based on the length of the longest term in the expression. For simple terms, each operand contributes one beat to the bar length, except the last operand in a term wich contributes two beats. This lengthening mimicks the final syllable lengthening in speech. An extra rest is added for a printed binary operator. A rest is used because, for a glance, the division into terms is the important feature, not the nature of the operator. In contrast, relational operators are important cues to the structure of an expression. In length calculations, a relational operator is included in the following term, being counted as one beat, plus a separator of one rest.

All complex objects (including superscripts) are represented by a continuous tone with a constant pitch, as non-terminal parenthesized sub-expressions are in speech.

[2]Paper is an inappropriate medium for conveying descriptions of non-speech sounds. Examples of Algebra Earcons can be heard by visiting the World-Wide Web pages at `http://www.cs.york.ac.uk/maths/`.

This indicates that such an item is present, but reveals nothing of its contents beyond its length and location. This is consistent with the idea that an algebra earcon is a glance. The lengths of complex objects are calculated as above, except that binary operators do not make a contribution, reflecting the faster, pauselesss uttering of these objects in speech.

After the maximum term length has been calculated, each term in the expression is fitted into this bar length for the algebra earcon. Shorter terms are padded at the right with rests to preserve the rhythm of the algebra earcon.

Algebra earcons are played in the C major scale. The pitch of each new term starts at middle C (C_3). Subsequent objects are played at one note below the previous. The last term's pitch starts at A_4. This mimicks the sharp pitch fall at the end of an algebraic utterance, that indicates the impending end of the expression to the listener. If the relational operator precedes the final term, the note representing the first operand is played at F_4, as the relational operators are also played at A_4. Superscripts are played at a pitch one tenth above. This represented a dissociation between the earconic form for a superscript and the spoken form. In the spoken form, superscripts follow the pitch fall of the term to which they are attached. In the earcon, the higher pitch used to represent superscripts was chosen as a correlate of the higher position in print. The pitch change is introduced to add redundancy to the indication of a superscript. Simply using the musical timbre to find the superscripts may not be sufficient if the pitch trend simply follows that of the rest of the term.

Sub-expressions are played one eighteenth below the preceding object or initial pitch for a term if the quantity is the first item. This form of presenting sub-expressions mimicks that of those spoken in the middle of the utterance. The linear pitch falls seen for complex objects and the termini of utterances are not used in order to make the form of the earcon as simple as possible. By the same reasoning the declination effect is not used in earcons. A sharp pitch fall is used at the end of the earcon to signal the termination of the earcon.

Simple and complex fractions are both represented by pan pipes, but with a different pitch profile. Simple fractions have the same pitch fall throughout as simple terms, but there is a one octave drop at the start of the denominator. The last note in a simple fraction is lengthened as if it is the last note in a term. Again, this representation is very similar to that of such fractions in the spoken form.

Complex fractions are represented by two long notes of constant pitch, separated by two silent beats. This change in representations for complex fractions mimics the similarity between complex fractions and parenthesized sub-expressions, which is also observed in the spoken form. The silence between the two terms of the fraction represents the fraction line or 'over'. The second note, the denominator, is played two notes lower than the first. This attempts to indicate that the denominator was 'lower' and separate from the numerator.

For all complex objects any objects appearing as a prefix or suffix are separated from the complex object by a silent rest. This mimicks the separation seen in the spoken forms. Such separations are thought to aid discrimination between objects in the earcon.

Amplitude is increased in the same pattern as the spoken form. Amplitude is raised for the first operand of each term, unless that operand is complex. Only simple objects have amplitude increased. Superscripts and the relational operators are also increased in amplitude.

[Figure 9 about here.]

An example of a complete earcon for the expression $3x+4=7$ is illustrated in Figure 9(a) This has three terms, making a three-bar algebra earcon. The bar representing the first term, '$3x$', has a length of four beats: A note of one beat for the '3', two beats for the 'x' and one rest for the '$+$' from the following term. The second bar, for the term '+4', has a length of three beats. Two for the '4', as the only operand is the final operand, thus given a length of two beats. One beat is added for the minimal separation of one silent rest from the next term or motive. The final term, '$=7$', has a length of four beats. One for the equals symbol, one rest separating this from the '7' and two for the '7' itself. A separator rest is not added to this term as it is the final term of the expression. Therefore, the bar length of this earcon is four beats. The first and third terms already fit into this bar length. The second '$+4$' has an extra silent beat appended to make it fit this length. Having developed a bar length for the earcon, the loose rhythm of the spoken form can be fitted into a more formal, stronger musical rhythm.

The next stage in the construction of the earcon is the assignment of pitches and timbres. A piano note at C_3 is used for the '3' and one at B_4 for the 'x'. For the start of the new term, the note representing '4' is again played at C_3. The marimba timbre used for '$=$' is played at A_4. To emphasize the pitch fall at the end of the expression, the piano note for '7' is played two notes below this, at F_4. The notes for '3', '4', '$=$' and '7' were all increased in amplitude. This completes the generation of the earcon for the expression $3x+4=7$.

The example $3(x+4)=7$ (Figure 9(b)) has the same lexical content as the previous expression, but a different syntax and therefore a different earcon. There are two terms,'$3(x+4)$' and '$=7$'. The sub-expression '$(x+4)$' has a length representing the two internal terms but with no separation for the '$+$', giving a length of four beats, two each for the final operands of two terms. The coefficient, '3', adds a further beat and a rest is added to separate it from the quantity. As before, the '$=7$' has a bar length of four beats. No adjustment for bar length was needed as there are only two terms.

The piano timbre used for '3' is played at C_3. The sub-expression is played as a single note at A_6 with a cello timbre. Finally the '$=7$' is played as before. This time only the '3', '$=$' and '7' are increased in amplitude. This example shows how the earcon can distinctly show the difference between two lexically similar expressions.

6.3 Evaluation of Algebra Earcons

This evaluation sought to demonstrate whether the algebra earcons could work as a glance, that is, convey the presence, location and size of structural objects in a rapid manner to a listener. The experiments did not seek to find if listening readers could or would use algebra earcons as a glance in the manner proposed. First, could listeners recover enough information about the objects within an expression such that they could determine its type? Second, do algebra earcons present all types of object within an expression to equal effect?

To this end, a multiple-choice paradigm was used to fulfil the aims of the experiment. An advantage of this design is that the distractors presented along side the stimulus can be designed such that all aspects of the rules for constructing algebra earcons can be probed.

A two-condition, within-participants design was used. A significant bias towards the correct choice in a question would indicate the earcons were able to present the structure of an expression. Looking across questions for those with a low score would reveal which aspects of expression structure caused problems. The options in the multiple choice were designed such that only one aspect of the distractors differed from the correct answer. If participants were lured to one of these choices then flaws in the construction of algebra earcons could be determined.

Twelve fully sighted, normally hearing participants were used in this evaluation. The same rationale for using sighted participants in previous evaluations were deemed to stand for the current experiment. The participants were a mixed group of graduate and undergraduate students from a range of disciplines. All participants were familiar with the form of algebra expressions and could name parts of expressions.

A total of 30 expressions were made, equally divided between syntactically simple and complex. The simple expressions had no complex objects, but could have many simple ones. The complex expressions always had at least one complex item, but could also include simple objects. Within each set a range of expression lengths were used to see if participants could be overwhelmed in the same way as listeners to spoken expressions.

Results and Discussion

In this experiment a high proportion (approximately 73%) of questions were answered correctly. Participants performed much better than chance in both simple and complex conditions. In both conditions the means were approximately 11 correct in 15 responses. (See Figure 10 for individual and combined scores for each condition). A binomial test for 11 correct in 15 responses, with a probability of success being 0.25, gave a probability of this result happening by chance of 0.0001.

[Figure 10 about here.]

An examination of the results across questions revealed which presented the most problems. In all but one case the most common answer was the correct one. Incorrect answers were usually concentrated on one or two of the distractors, making the determination of faults in earcon design easier. In the simple condition 46 errors were made and these were distributed amongst 21 of the 45 distractors. In the complex condition 54 errors were distributed amongst 26 of the 45 distractors, making a total of 100 errors. The distractors were originally designed based on experience from errors encountered during the recall reports in Section 4. The distractors on which these errors were made were collated into the following categories, that are explained in the list below. When a category, such as omissions, was too broad and a large number of errors were related, a sub-group was formed.

1. There were 30 **timing errors** where a sum was chosen in preference to a product, or *vice versa*. eighteen of these were in turning sums to products. This may suggest that the gap between objects was too short.

   Five of these errors were due to shortening of complex objects. The relative length of complex objects is given by the length of the note.

2. There were 28 **omission errors** where objects other than superscripts were omitted. Ten of these were the omission of terminal objects. Only two occurred in the complex condition. The complex expression earcons often contain fewer objects and this may account for the reduced number of omission errors.

3. Fourteen **superscript errors** were made. Twelve of these were the omission of a superscript from the end of an earcon.

4. **Timbre errors** cause distractors with different types of object from the original to be picked. These could either be caused by participants making an incorrect mapping between musical sound and object type, or by not being able to discriminate between timbres. There were twelve timbre errors which involved transformation to or from fractions, indicating that the fraction (pan-pipe) sound was difficult to discriminate.

5. **Scope errors** are distractors in which the complex item is dilated to subsume other objects or contracted to add further objects to the expression. Ten errors were made by picking distractors with altered scope. Seven of the ten scope errors can be accounted for by problems with the perception of length in complex superscripts.

6. There were a total of six **relational operator errors**. One was a simple omission. Four were with the translocation of the relational operator with

another printed operator. One was in the transformation of a term preceded by an equals sign into a superscript.

The timing or length of pauses between objects in the earcon caused problems. Errors due to the representation of fractions account for most of these errors. The fact that no distractors with timing errors were *unpicked* supports the finding that timing was a problem. That so many errors involved transforming sums (pauses) into products (no pauses) indicates that the pauses between objects to indicate separation into terms may be too short for some listeners to use easily. To this end, the minimum rest between terms was increased from one to two beats. The rest separating coefficients from complex objects was removed. This change in design attempted to make the timing structure more prominent, whilst retaining the rapidity of the glance as far as possible.

Omission errors formed the largest category of errors. There are memory limits to how many objects or groups of objects listeners can maintain after hearing them. Algebra earcons with more sounds or groups of sounds were remembered less well. Distractors with omission errors were not chosen for expressions with few objects in the earcon. This would support the suggestion that the majority of omission errors occur when the number of sounds are large.

Inherent memory limitations make it difficult to resolve such errors. The algebra earcon could be made slower, giving the listener more time to process the information. In addition, the glance need not be fully correct. For instance, that one operand is missed from the start of an expression would not seriously impair the use of the representation held by the user as a glance.

Two types of object, fractions and superscripts, caused a large number of problems. Some participants complained that the representation of the fractions was too fast, making it difficult to discriminate the content. Others mentioned that the pan-pipe sound was faint, relative to other sounds. The loudness of the pan-pipe timbre was increased and a rest introduced to represent the ‘over’ thus slowing down the simple fraction.

When a superscript appears on the final object in an expression the violin sound used in the algebra earcon has its pitch lowered. This may have made it more difficult to discriminate from other sounds. The design was changed so that all superscript sounds were played at the same pitch and the sound was made louder. The change in the pitch of the terminal violin sound may not aid recall by enhancing detection, because the problem may be simply one of memory limitation.

Errors due to mistakes with the relational operator were rare and a large number of the distractors with altered relational operators remained unpicked. However, complaints about the timbre used led to the more distinctive 'rim-shot' percussion sound being used instead.

Follow up recall tests suggested that participants were able to recall information about the general structure of expressions rendered as earcons. These recall reports displayed a range of detail. For short expressions or simple earcons (as well as some longer ones) a fully detailed description was given of an expression that would be represented by the glance. One participant on hearing the algebra earcon for a quartic expression (including all the powers) simply said 'It's a quartic'. For the expression $a(b+c)-d(e-f)^g = h$ one participant recalled: 'a times expression, plus b times expression to the power of something equals something else.' For longer expressions or more complex earcons progressively more information was lost. The weakest recall reports would simply state the size and balance of an expression (how much was on either side of a relational operator) and the presence, but not location of certain objects.

6.4 Conclusions

Listeners were able to recover enough syntactic information from the algebra earcon to choose an appropriate expression from a list of similar alternatives. The fact that many of the stimuli were long and complex, and the distractors sometimes very similar, indicates the ability of algebra earcons to convey structural information to a listener at a glance. The errors made fell into distinct categories that enabled some of the problems with the algebra earcons to be highlighted and design amendments made.

The recall reports indicated that this success was not only due to the recognition component of the multiple-choice experiment. Many participants recalled a detailed framework of an expression's structure. Large complex expressions were sometimes only recalled as an impression of size, balance and presence of certain objects. Any information from this spectrum of glances could provide suitable information about structure conforming to the definition of a glance and needs of a glance at algebra.

7 THE INTEGRATED MATHTALK PROGRAM

The object of this final stage in the development of the Mathtalk program was to test if the integrated system did in fact transform the passive listener to the active reader. The Mathtalk program was compared to the use of expressions, presented in Latex format (Lamport, 1985) in a word-processor accessed using a screen reader and synthetic speech. The survey of secondary level mathematics undertaken as part of the EU Tide project Maths (Boormans and Cahill, 1994), revealed that blind mathematics pupils did not use tape recorded speech, but did use some linear form of algebra accessed via a word-processor. In addition, it has been reported that many users of mathematics made use of Latex notation for performing mathematical tasks (Stöger, 1992; Burger et al., 1996). Thus the comparison between Mathtalk and this method has ecological validity.

The word-processor condition (the combination of Latex notation and the word-processor) contains all the grouping information necessary for an unambiguous reading of an expression. However the presentation in speech does not add any of those features found to aid parsing and retention of memory available in prosody. Importantly, the word-processor presentation contains equivalents of the lexical cues found to be so disruptive of the retention of content in Section 4. The IBM Screen reader (Thatcher, 1994) speaks the expression

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{7}$$

rendered in Latex as:

> ‘x equals backslash frac open brace hyphen b backslash pm backslash s q r t open brace b circumflex two hyphen four a c close brace close brace open brace two a close brace.’

Displaying this notation within a word-processor also allows the listening reader to control the information flow. The reader can only move character to character or word to word within an expression. Whilst this allows the reader to visit all parts of an expression it will be more difficult to visit specific portions of an expression and have larger objects spoken in isolation, for example, fractions and sub-expressions. This impoverished control and display compared to the Mathtalk program should highlight the differences between access and usability.

7.1 Design

A co-operative style of evaluation was used (see Section 5). Blind participants were given a mixture of navigation and mathematical tasks to perform on a set of algebraic expressions. Participants were asked to 'think aloud'. Performance on the tasks, recordings of commands issued and user protocols gave evidence of style of interaction and an objective measure of performance. A NASA-TLX and a post-experiment questionnaire were used to assess the participant's mental workload. Preferences (as measured in Section 4) and comments on the two systems were also used.

Two counterbalanced conditions were used in a within participants design: The word-processor condition and the Mathtalk condition. In the evaluation of the browsing component, the balance of tasks was towards the navigation and orientation within and without expressions. This time the tasks more closely resembled real mathematical tasks. The user was asked to substitute values into the variables within expressions and calculate the arithmetic value.

Some qualitative and quantitative measures were used to assess usability:

- The time taken to accomplish each task;
- the number of commands used and number of errors made during the tasks;
- the type of moves made during the tasks;
- the mental workload associated with the tasks;
- the users' satisfaction with the two methods of presentation.

The following changes had been made to the Mathtalk program from that used in the evaluation of the browsing language described in Section 5:

- The action **glance** had been added to the list of actions.

- The command changes detailed in Section 5 had been completed. The most significant of these was to change **speak** to **show** and to make **current expression** consistent with the other **current** commands within complex objects. This meant introducing the **which expression** command to speak the expression number.
- The algebra earcons were re-implemented using the Proteus music synthesizer. This synthesizer had much stronger timbres that should have been easier to discriminate. Piano was used for base level operands; silence for printed, non-relational operators; drum for relational operators; trombone for fractions; violin for sub-expressions and an electronic 'beep' for superscripts.
- The terminus sounds were mapped onto these timbres and the other changes recommended in Section 5 were implemented.

One set of ten training expressions and two sets of twelve matched expressions and questions were set for each condition. No visual display was available. The IBM Screen Reader was used to access the WordPerfect word-processor used to access the Latex form of the expressions. This enabled the participants to use the Multivoice speech synthesizer in both conditions. None of the participants were familiar with this synthesizer, but the quality was such that no training was needed. None of the participants were familiar with either the Mathtalk program or the IBM Screen Reader, but all were familiar with WordPerfect.

Four blind participants were used in this evaluation. The participants were not only visually disabled, but also computer users at a reasonably advanced level of mathematics education. The four participants were 17 to 32 years old, and either had already taken or were in the process of taking mathematics exams usually taken at age 18.

A general explanation of the purpose and style of the experiment was given to the participants. It was stressed that it was the software the participants were evaluating; their mathematical ability was not being tested. The nature of each condition was described to the participant. The speech and non-speech audio were presented to the participants using external loudspeakers.

7.2 Results and Discussion

Each participant used many more commands in the word-processor condition than in the Mathtalk condition (see Figure 11). Despite using fewer commands, the Mathtalk presentation provided a greater variety of appropriate views of the expressions. The main strategy in the word-processor condition was a character-by-character reading and rereading of an expression. In contrast, in the Mathtalk condition, terms were read rather than single items; complex objects were moved to and spoken as a whole and glancing and speaking of whole expressions was used.

Two typical dialogues between reader and Mathtalk then reader and word-processor are reproduced below. In the first the reader is substituting the value $x = 2$ into the expression $y = (x+3)(x-2)$:

Current level `y equals a quantity times a quantity.`

Next quantity `a quantity.`

Show quantity `x plus three.`

Reader 'is eight.'

Next item `a quantity.` <end sound>

Show quantity `x minus two.`

Reader 'times two is sixteen.'

For the word-processor condition (the commas separating items indicate repetition of the command) the expression $y = 2^{x+1} - 5$ was represented as `y = 2^{x +1} -5`:

up/down `six period y equals two circumflex left brace x plus one right brace hyphen five.`

Reader 'I'll skip through it, because its too long. I know it's got a circumflex in it.'

Experimentor '*What does that mean?*'

Reader 'Squared, something to the power of.'

Right cursor `six, period, _, y,, _, equals, two, circumflex, left brace, x, plus, one, right brace, hyphen, five, _, space, six period ....`

Reader ‘y equals two with a power x plus one, minus five.’

[Figure 11 about here.]

For the word-processor condition the range of strategies and commands used were very narrow, in spite of the range of browsing commands available in the word-processor. Figure 12 shows the percentage of the total keystrokes contributed by each command. All of the commands used by each participant are included in this figure.

[Figure 12 about here.]

Using only the cursor keys complex objects such as parenthesized groups and fractions could not be treated as single units – a technique that appeared to facilitate the evaluation and substitution tasks in the Mathtalk condition. The overall structure seems to have been lost in a welter of symbol names and little moves.

The Latex notation itself was probably the reason full utterances were not used. The braces, parentheses and special words preceded by a backslash made the utterances very long. The expressions were also spoken without any pauses other than inter-word pauses. This made the utterance ‘relentless’. This presentation style was an equivalent of the lexical condition of the experiment performed in Section 4 in which little structure or content was reliably recovered. The hierarchy of views available in Mathtalk, together with the prosodic presentation, meant that a full utterance was a useful component of the display.

In the word-processor condition there was an inability to reliably notice the end of an expression when browsing. As the participant moved character-by-character through the expression, a single move could take the focus of attention onto a new line and cause that line to be spoken in full. The user then had to either move up a line or several characters backwards to regain the current expression. Such wanderings required reorientation and rereading.

In the Mathtalk condition, the style of usage of the browsing commands varied between the participants, though some common features were present. The range of commands used can be seen in Figure 13. The Contrast in the number of different commands used can be seen in Figure 14. All used the facility to multiply the **next** and **previous** actions to move around the expression list, rather than visiting each expression. (For instance, the command **3ne** issues the **next expression** command three times). All used the glance in the navigation condition and a mixture of **current expression** and **current level** to gain views of the whole expression. Use was made of hidden objects when participants moved straight to that object and revealed its contents with **show**. Another general feature was the reliance on term-by-term reading, with heavy use of the default strategy.

[Figure 13 about here.]

A far larger range of strategies and tactics were available in Mathtalk and the participants took advantage of this opportunity. This contrast may be seen in Figures 12 and 13 where the percentage contributed by each command is shown. For the word-processor condition all keystrokes are accounted for by only a few commands. In contrast, though some moves are popular, a larger range are used in Mathtalk to give different views of an expression and move accurately to a particular position. For example, moving straight to a fraction or quantity and showing that object as one item.

[Figure 14 about here.]

Muting of full utterances was frequent[3] in the word-processor condition, but was requested only once in the Mathtalk condition. The example shown below indicates that the participant felt he had enough control over the information flow in Mathtalk to not need a mute very often. This, and similar comments from other participants, indicate the success of designing for control of information flow.

> 'I didn't think there was a need for it. On this one it just reads the whole line, where on the other you have to make out to get it to read the

[3] For example, each time a reader moved to a new line a full-utterance was produced. Muting was effected by issuing a new key-press. Exact numbers were not recorded as they were in the order of hundreds.

line. . . . you have more control in the last one.'

[Figure 15 about here.]

Although navigation times for Mathtalk averaged only 76% of the time required when using a word processor, neither navigation nor evaluation times (see Figure 15) were found to be significantly faster for Mathtalk than for word processing. In the Mathtalk condition fewer commands were used in the same time span as the word-processor condition. However, as described above, the participant using Mathtalk usually gained several views of an expression during this time and built up a fuller description of the expression in easy stages. In the word-processor condition, the participant usually took a single, character-by-character view of the expression and gave a poorer description of it.

The pattern of use of the algebra earcons was clear cut. They were heavily used in the navigation tasks, with almost every participant using them as the initial view of the expression to be explored. During the expression-evaluation tasks the audio glance was only used on four occasions, by one individual. During the navigation tasks all participants were able to give suitable descriptions of expressions. Using the terminal sounds to associate musical timbre with type of object meant that participants could name the object being terminated. The repetition of terminus sound at the end of the expression meant that participants were sure when they had reached the end of the expression. The inability to move past the end of an expression or internal object and become mixed with the next improved the apprehension of structure and avoided confusion.

Participants showed a distinct preference for the Mathtalk condition. This was measured using a scale presented after both conditions: Zero indicating preference for condition one, 20 for condition two and ten no preference. Taking condition one to be the word-processor and condition two to be the Mathtalk program, the mean rating was 16 suggesting a favoring of Mathtalk. The small number of participants meant no statistical test was undertaken. The NASA-TLX subjective mental workload scales (see Section 4) suggested that participants found the Mathtalk condition less mentally demanding and frustrating with a lower overall mental workload (5.5 for Mathtalk vs. 10.2 for the word-processor). Again, no statistical test was carried out due to the small number of participants.

This evaluation demonstrated that, in general, the Mathtalk program enabled a more usable reading interaction with algebra notation. This result supports the general principle of designing for external memory and control to give active reading. Support for this came from the participants' comments.

Mathtalk allows a wider range of views of an algebra expression and the protocols revealed these were exploited by the participants to give a more effective interaction with fewer commands. With the word-processor, participants essentially only used a character-by-character reading strategy. In contrast, when using Mathtalk, moves more appropriate to the structure of an expression were used.

8 CONCLUSIONS

The design and evaluation work within the Mathtalk project has led to the following design principles being recommended to form the basis of designing auditory displays of complex information:

- By basing the design on compensation for lack of external memory and provision of control over information flow a passive listener can be transformed to an active reader.
- When the structure of complex information is known, prosodic cues can be inserted into the synthetic speech output to facilitate recovery of structure, retention of content and reduce mental workload.
- Provision of fast and accurate control over information with structure based browsing, incorporating hiding of complexity, can make reading active.
- A glance or overview of the structure of information can be provided by combining the prosodic features of the spoken output, the hiding of complexity and the use of earcons to convey information.

The difficulty of the problem of making mathematics accessible suggests that a multi-modal approach is not only attractive but necessary. Some of the potential of such an approach has been investigated within Mathtalk and that work is being extended within the Maths Project. Maths is funded by the European Tide initiative, with partners from several European countries[4]. Maths extends the concept of a multi-modal system to use soft braille displays, braille input and enlarged character display, speech input, as well as the audio component to read, write and manipulate algebra notation in a commercial graphically based text and mathematics editor (Edwards and Stevens, 1995).

This paper really presents work in progress in the project and as such it is not possible to draw many hard guidelines as yet. So far the multi-modal approach appears to be successful and it can be said that eventually the Maths project will provide a lot of information as to what approaches to these kinds of problems are appropriate. The Maths project will provide information on a larger number of visually disabled participants as well as longitudinal studies to evaluate the user interface. Note that the approach is truly multi-modal, in that it relies on a variety of input *and* output modes.

The particular field chosen here, mathematics, may seem to be a minority interest, but this is not true since it is the concern of *every* school student. At the same time the implications of the work will be much broader. There are many applications in which people need access to large amounts of complex information. Sighted people are usually accommodated by way of visual interfaces and this project will provide good information as to how that information should be transformed into non-visual forms accessible to blind people.

9 NOTES

Background This paper is based on worked carried out as part of a Ph.D. thesis and as part of the European funded Maths project.

Acknowledgements We would like to thank Dr. Peter Wright of the University of York and Dr. Stephen Brewster of the University of Glasgow for their contributions to this work.

[4]F. H. Papenmeier, Germany; University of York, England; Grif, S. A., France; Katholieke University Leuven, Belgium and University College Cork, Ireland.

Support Mathtalk was developed partly on the basis of a studentship from the UK Science and Engineering Research Council (number 91308897). The Maths Project is funded by the European Commission through its Tide Initiative, project number 1033.

Postal Addresses Robert Stevens

Department of Computer Science

The University of York

Heslington

York UK

YO1 5DD

Email: `robert@minster.york.ac.uk`

Alistair Edwards

Department of Computer Science

The University of York

Heslington

York UK

YO1 5DD

Email: `alistair@minster.york.ac.uk`

Philip Harling

Department of Computer Science

The University of York

Heslington

York UK

YO1 5DD

Email: `philiph@minster.york.ac.uk`

References

Aldrich, F. and Parkin, A. (1988). Improving the retention of aurally presented information. In Gruneberg, M., Morris, P., and Sykes, R., editors, *Practical Aspects of Memory 2: Current Research and Issues*. Chichester, England: Wiley.

Arons, B. (1993). Speechskimmer: Interactively skimming recorded speech. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 187–196.

Baddeley, A. D. (1992). *Your Memory: A User's Guide*. Penguin Books.

Barry, W. A. Gardner, J. A. and Lundquist, R. (1994). Books for blind scientists: The technological requirements of accessibility. *Information Technology and Disability*, 1(4). Article 8. (On-line journal available on Internet, ISSN 1073-5127 `http://www.rit.edu/~easi/itd.html`).

Bates, M. J. (1989). The design of browsing and berry picking techniques for the online search interface. *Online Review*, 13(5):407–424.

Beech, C. M. (1991). Interpretation of prosodic patterns at points of syntactic structure ambiguity. *Journal of Memory and Language*, 30:643–663.

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

Boormans, G. and Cahill, H. (1994). Problem analysis: A formative evaluation of the mathematical and computer access problems experienced by visually impaired students. Deliverable D1, The EU Tide Maths Project (TP1033).

Bostock, L. and Chandler, S. (1981). *Mathematics: The Core Course for A-Level*. Stanley Thornes (Publishers) Ltd. Cheltenham.

Brewster, S. A., Wright, P., and Edwards, A. (1994). A detailed investigation into the effectiveness of earcons. In Kramer, G., editor, *Auditory Display: The Proceedings of the First International Conference on Auditory Display.*, pages 471–498. Reading, Massachusetts: Addison-Wesley.

Burger, F., Knasmüller, G., Miesenberger, K., and Stöger, B. (1996). Access to mathematics for the blind. In Berger, D., editor, *New Technologies in the Education of the Visually Handicapped*, volume 237, pages 263–269. John Libbey Eurotext Ltd.

Crispien, K., Wuerz, W., and Weber, G. (1994). Using spatial audio for the enhanced presentation of synthesized speech within screen-readers for blind computer users. In Zagler, W. L., Busby, G., and Wagner, R. L., editors, *Computers for Handicapped Persons: Fourth International Conference, ICCHP'94*, pages 144–153. Berlin, Springer-Verlag.

Crystal, D. (1975). *The English Tone of Voice*. Oxford University Press.

Edwards, A. D. N. (1989). Soundtrack: an auditory interface for blind users. *Human-Computer Interaction*, 4(1):45–66.

Edwards, A. D. N. (1991). *Speech Synthesis: Technology for Disabled People*. Paul Chapman, London.

Edwards, A. D. N. and Stevens, R. D. (1995). Une interface multimodale pour l'accèss aux formules mathématiques par des élèves ou étudiants aveugles. In *Comme les Autres: Interfaces multimodales pour handicapés visuels, Special number 1*, pages 97–104, Université Pierre et Marie Curie, B23, 9 Quai Saint-Bernard, 75252, Paris Cedex 05 and ANPEA (ISSN 0010-2520). INSERM.

Ellis, A. and Beattie, J. (1986). *The Psychology of Language and Communication*. Weidenfeld and Nicolson.

Ernest, P. (1987). A model of the cognitive meaning of mathematical expressions. *British Journal of Educational Psychology.*, 57:343–370.

Gardner, J. A. and Barry, W. A. (1993). Report on dotsplus. unpublished report, Department of Physics, Oregon State University.

Garnham, A. (1989). *Psycholinguistics: Central Topics*. Routledge, London.

Halliday, M. K. (1970). *A Course in Spoken English: Intonation*. Oxford University Press.

Kirshner, D. (1989). The visual syntax of algebra. *Journal for Research into Mathematics Education*, 20(3):274–287.

Knuth, D. E. (1984). *The TEX Book*. Addison Wesley.

Lamport, L. (1985). *Latex – A Document Preparation System – Users Guide and reference manual*. Addison Wesley, Reading.

Larkin, J. H. (1989). Display-based problem solving. In Klahr, D. and Kotovsky, K., editors, *Complex Information Processing*. Lawrence Erlbaum: Hillsdale New Jersey.

Monk, A., Wright, P., Haber, J., and Davenport, L. (1993). *Improving Your Human Computer Interface: A Practical Technique*. BCS Practitioner Series. Prentice Hall.

Mynatt, E. D. and Weber, G. (1994). Nonvisual presentation of graphical user interfaces: Contrasting two approaches. In Adelson, B., Dumais, S., and Olson, J., editors, *Celebrating Interdependence: Proceedings of Chi '94*, pages 166–172. New York: ACM Press.

NASA (1987). *Task Load Index (NASA-TLX)*. NASA Human Performance Research Group, NASA Ames Research Centre.

O'Malley, M. H., Kloker, D. R., and Dara-Abrams, B. (1973). Recovering parentheses from spoken algebraic expressions. *IEEE Transactions on Audio and Electroacoustics*, AU-21:217–220.

Raman, T. V. (1994). *Audio Systems for Technical Reading*. PhD thesis, Department of Computer Science, Cornell University, NY, USA.

Rapp, D. W. and Rapp, A. J. (1992). A survey of the current status of visually impaired students in secondary mathematics. *Journal of Visual Impairment and Blindness*, 26(Feb):115–117.

Rayner, K. and Pollatsek, A. (1989). *The Psychology of Reading*. Prentice Hall.

Schönpflug, W. (1986). The trade-off between internal and external information storage. *Journal of Memory and Language*, 25:657–675.

Stevens, R. D. (1996). *Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People*. PhD thesis, Department of Computer Science, The University of York, Heslington, York, UK. YO1 5DD.

Stöger, B. (1992). Blind and visually impaired people studying computer science and mathematics. *Journal of Microcomputer Applications*, 15:65–72.

Streeter, L. A. (1978). Acoustic determinants of phrase boundary representation. *Journal of the Acoustical Society of America*, 64:1582–1592.

Thatcher, J. (1994). Screen reader/2 – programmed access to the GUI. In Zagler, W. L., Busby, G., and Wagner, R. L., editors, *Computers for Handicapped Persons: Proceedings of ICCHP '94, Lecture Notes in Computer Science 860*, pages 76–88. Berlin: Springer-Verlag.

Weber, G. (1994). Braille displays. *Information Technology and Disability*, 1(4). Article 8. (On-line journal available on Internet, ISSN 1073-5127).

Weber, G. (1995). Reading and pointing - new interaction methods for braille displays. In Edwards, A. D. N., editor, *Extra-ordinary Human-Computer Interaction: Interfaces for Users with Disabilities*, pages 183–200. New York: Cambridge University Press.

List of Figures

Figure 1 Representation of the timing, pitch change and amplitude in $ax^2 + bx + c = 0$. The arrows show the trend of pitch change; periods indicate pauses; *italic* typeface indicates increased speed and **boldface** indicates increased amplitude.

Figure 2 Representation of the timing, pitch change and amplitude in $3(x+4) = 7$. See Figure 1 for key.

Figure 3 Representation of the timing, pitch change and amplitude in $(a+b)(a-b)$. See Figure 1 for key.

Figure 4 Mean percentage recall for structure, content and overall for each condition in each group.

Figure 5 Percentage changes in TLX factors from lexical to prosodic condition in the LP group.

Figure 6 The set of action and target words, with keystrokes, used to generate commands for the final evaluation of the command language and browsing functions.

Figure 7 Command usage as a percentage of the totals: **CL** is **current level**; **SE** is **speak expression** and **Default** refers to the default browsing style. The final row gives the overall total of commands and the average percentage for each command.

Figure 8 Figure of musical timbres used in algebra earcons.

Figure 9 The algebra earcons for $3x+4 = 7$ and $3(x+4) = 7$ in music notation. Length of notes, rests and pitches of expression objects are shown. Instruments have been omitted.

Figure 10 Number of correct responses out of 15 presentations for both simple and complex conditions and combined totals.

Figure 11 Total number of commands for each participant and means for each condition.

Figure 12 Percentage of total contributed by each command for each participant in the Latex condition. A zero percentage indicates a contribution of less than one percent. **K** = Keystroke and **P** = Proportion.

Figure 13 Percentage of total commands issued for the top ten most frequently used commands for each participant in the Mathtalk condition. **C** = Command and **P** = proportion. Command abbreviations: be= Beginning Expression; ce= Current Expression; cl= Current Level; ct= Current Term; ge= Glance Expression; ne= Next Expression; ni= Next Item; nt= Next Term; pe= Previous Expression; sf= Show Fraction; sq= Show Quantity; we= Which Expression.

Figure 14 Total number of different commands used by each participant in each condition.

Figure 15 Mean times in seconds taken to complete tasks in the two conditions.

a x super **two** . plus **b** x . plus **c** . **equals zero**

Figure 1:

Three. *x plus four*. **equals seven** .

Figure 2:

a plus b . $\uparrow$ **times**. *a minus b*

Figure 3:

	Group			
	LP		LN	
	Lexical	**Prosody**	**Lexical**	**Nothing**
Structure	67	88	62	37
Content	52	79	45	67
Overall	49	76	40	35

Figure 4:

Factor	Lexical group	Prosodic group	T(11)	P	Percentage Change
mental demand	15.92	13.50	3.294	<0.01	12
time pressure	13.92	10.50	4.492	<0.01	17
effort expended	12.33	10.67	2.209	<0.05	8
performance level	7.42	11.83	-5.54	<0.01	-23
frustration	13.17	8.83	3.17;	<0.01	22

Figure 5:

Action	**Key**	**Target**	**Key**
Speak	s	Expression	e
Current	c	Term	t
Next	n	Item	i
Previous	p	Quantity	q
Beginning	b	Super	s
End	e	Fraction	f
Into	i	Numerator	n
Out-of	o	Denominator	d
		Level	l

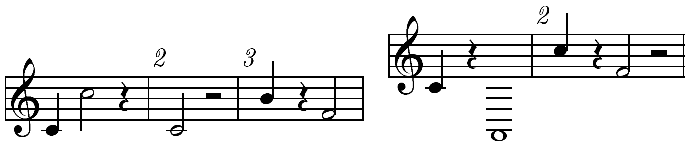
Figure 6:

Participant	**Total**	**CL**	**SE**	**Default**
E7	179	20	3	17
E8	168	12	2	25
C3	192	16	5	24
E3	202	14	2	29
E5	248	10	7	26
Overall	989	14	4	24

Figure 7:

Object	**Timbre**
Base-level operands	Acoustic Piano
Binary Operators	Silence
Relational operators	Marimba
Superscripts	Violin
Fractions	Pan pipes
Sub-expressions	Cello

Figure 8:



(a) $3x+4=7$

(b) $3(x+4)=7$

Figure 9:

Participants	**Simple**	**Complex**	**Total**
E1	10	7	17
E11	11	7	18
E5	11	8	19
E6	8	11	19
E12	11	9	20
E10	9	12	21
E3	11	11	22
E7	12	12	24
E9	12	12	24
E4	13	12	25
E2	13	13	26
E8	13	13	26
Across Participant Mean	11.17	10.58	
Across Question Mean	8.93	8.4	

Figure 10:

	Participants				
Condition	F1	F2	F3	F4	mean
Mathtalk	257	239	322	341	285
Word-processor	640	662	142	487	483

Figure 11:

F1		F2		F3		F4	
K	**P**	**K**	**P**	**K**	**P**	**K**	**P**
Next-word	15	Previous-char	13	Next-line	7	Next-line	16
Next-char	35	Next-char	70	Next-char	63	Next-char	132
Previous-line	13	Next-line	7	Previous-line	7	Previous-line	16
Next-line	11	Previous-line	4	Line-start	6	Previous-char	13
Previous-char	13	Line-start	3	Previous-line	4	Line-start	8
Previous-word	7	Document-top	0	Previous-word	6	Previous-word	
Line-end	1	Document-end	0	Next-word	3	Line-end	
Document-top	0	Next-word	0	Document-start	0	Document-top	
Line-end	0	Document-end	0	Line-end		Line-end	
Total %	100		100		99		100
Total commands	640		662		142		487

Figure 12:

F1		F2		F3		F4	
C	**P**	**C**	**P**	**C**	**P**	**C**	**P**
Default	19	Default	47	ne	15	Default	17
ge	14	ne	13	ge	12	cl	15
ce	9	ce	12	nt	8	ge	13
ni	9	ge	7	ce	7	ne	11
ne	8	Multiple	6	cl	6	Multiple	5
Multiple	7	we	5	be	5	sq	4
cl	6	be	3	ni	5	sf	4
Errors	6	cl	3	Multiple	4	we	4
pe	4	pe	2	we	3	ce	4
be	4	ct	1	Errors	3	Errors	4
Total %	86		99		68		81
Total Commands	257		239		322		341

Figure 13:

Condition	Participant			
	F1	F2	F3	F4
Mathtalk	23	12	34	24
Word-processor	9	9	8	5

Figure 14:

Tasks	Condition	
	Mathtalk	Word-processor
Navigation	69	91
evaluation	92	94

Figure 15: