Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People

Robert David Stevens

Submitted for the degree of Doctor of Philosophy

The University of York

The Human Computer Interaction Group,

The Department of Computer Science.

January 1996

Abstract

This thesis proposes a set of principles to aid the design of user interfaces that enable blind users to read complex information by listening. Prior to this work speech based interfaces tended to ‘read at’, rather than being read by the listener. By addressing the themes of control of information flow and the lack of external memory, a set of guidelines have been produced that transform the passive listener to an active reader.

Prosody was used to add information to a spoken presentation of algebra in order to enhance its role as an external memory. A set of rules were developed that inserted prosodic cues for algebra into synthetic speech. An experiment found that these cues enhanced the recovery of syntactic structure; the recovery of content and reduced mental workload.

A structure vbased browsing method and associated command language were used to add control over the information flow.An iterative cycle of design and evaluation allowed the development of a style of browsing that would allow the fast and accurate control needed for active reading.

The final component of the system was an audio glance at the structure of an algebra expression. This was a combination of the prosodic rules that enabled presentation of structure and audio messages called earcons. Experiments were conducted that showed these *algebra earcons* were able to rapidly convey a suitable representation of an expression from which structural complexity and type could be judged, thus facilitating the planning of what browsing moves to use.

The three components of the system wer drawn together and evaluated. A comparison was made with a conventional style of presentation. The new design was found to be more effective, efficient and satisfying by the users of the system. The design guidelines set forth in this thesis offer a method to make the access to complex information by blind people more usable.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank a group of people without whom this thesis would have been different and more difficult to acheive. Peter Wright of the Department of Computer Science at York has acted as both a psychology and experimental design guru. Proffessor John Local, of the Department of Linguistic Science at York, has given me much sound advice, help and encouragement on my investivations into prosody. I would also like to thank Alan Dix, of the Department of Computer Science at The University of Huddersfield, for knowing much more about mathematics than myself. Finally, but certainly not least, I would like to thank Alistair Edwards who supervised me in this project and did all that was necessary and more.

There are a host of people without whom the duration of my studies would have been much more grey. Principal amonst these are Stephen Brewster and Philip Harling. Thanks should also go to Ian Pitt for his musical and electronics expertise and to Andy Dearden for his ability to find his way through a LaTeX manual.

The mountain of papers and paper would not have been manageable without the legion of readers and editors of scanned text. There are many, but amonst the longer standing are Andrea Gunther, Sandra Foubister and Deta Dickenson. Connected to this mountain of paper are the staff of the general office, particularly Truda Counsell, whom I would like to thank for all their help.

This work would not have been possible without the funding provided by an EPSRC studentship.

Mary Rose, my love, did all the right things during this work. Importantly she did not ask when it was going to finish nor did she seem to resent, too much, the invaded weekends. I would like to thank her very much.

Dedication

I would like to dedicate this thesis to my parents for my education.

Declaration

I declare that the work presented in this thesis is my own.

The following papers have described the general approach taken in the design of the Mathtalk program and have charted its development: Stevens and Edwards (1993), Edwards, Pitt, Brewster, and Stevens (1995), Edwards and Stevens (1993), Stevens and Edwards (1994a), Edwards and Stevens (1994), Edwards, Stevens, and Pitt (1995).

The evaluation of the effects of adding prosody to synthetically spoken algebra appeared as Stevens, Wright, and Edwards (1994) at HCI'94. A description of the browsing component of Mathtalk appeared as Stevens and Edwards (1994b). Part of Chapter 5, which describes the audio glance, appeared as Stevens, Brewster, Wright, and Edwards (1994). Part of Chapter 6, on the final evaluation of Mathtalk, appeared in Stevens and Edwards (1996), that will appear in April 1996.

In all these cases I have exploited only those parts of the work that are directly attributable to me.

Chapter 1

Introduction

1.1 Introduction

To attempt to design an auditory interface that enables a blind person, listening to speech output, to read complex information such as algebra, is a natural progression in the design of user interfaces for visually disabled people. Synthetic speech is widely used in adapted interfaces for visually disabled computer users (Edwards 1991; Griffith 1990). However, speech presentation is almost exclusively used for linear text representations of natural language.

The introduction of computer technology into the work-place and education has had a great impact on opportunities available to visually disabled people in both education and employment (Griffith 1990). The main means of interaction between human and computer is by keyboard for input and a visual screen for output of information. With appropriate tactile marking of the keyboard, input of information presents few problems for visually disabled people. Naturally, the visual display of information needs special adaptation. Software products called screenreaders provide an alternative to visual display by rendering textual information into either braille, synthetic speech or enlarged visual displays (Edwards 1991). These programs follow the focus of attention around the visual display, rendering what is typed and what is displayed by the computer for purposes of dialogue.

Such adaptations make mainstream applications such as word-processors, spreadsheets, data-bases and compilers accessible to visually disabled users. This means that visually disabled children and adults can input, review and manipulate many kinds of information making educational targets and employment easier to attain.

Advancements in user interface design, such as the graphical user interface, initially presented problems of accessibility for visually disabled computer users (Boyd, Boyd, and Vanderheiden

1990). However, technical advancements in the design of screenreaders now mean that visually disabled people have access to products such as Microsoft Windows (Crispien and Petrie 1993) and thus the usefulness of computers in the lives of visually disabled people will continue. Yet, not all types of information are fully accessible with the current range of screenreaders.

As mentioned above, for the most part, screenreaders render only textual representations of natural language. For this type of information, being able to move backwards and forwards through the text, is enough to give adequate access to that information. Unfortunately, access to simple linear text is not enough to fulfill the educational and employment needs of the majority of visually disabled people.

Speech output has not been used for the presentation and access of complex, but essentially still text-based, information with any great success. This thesis provides a set of principles for the design of auditory computer user interfaces that will allow designers to make tools that enable blind computer users to access, using synthetic speech, complex information as part of their everyday work in education and employment.

1.1.1 Simple and Complex Types of Information

To put this work in context, an immediate question is what is complex information? It is difficult to provide a hard definition for why something is complex and another type of information simple. It is not even true that all of one type of information is complex and all of another simple. It is more that some types of information have an inherent potential to be complex. This thesis mainly deals with the design of a user interface that facilitates the reading of algebra notation. In an attempt to define simple and complex notation algebra will be compared to printed natural language. In this thesis, algebra will be the exemplar of complex information and printed natural language will be taken as an exemplar of a simple information source.

Plain text can be regarded as simple because it is linear, structurally simple and generally redundant in its information. In all this work, a distinction should be drawn between structure and meaning. This thesis concentrates on the presentation of information that has an inherent potential to have a complex structure, rather than complex meaning. Complex meaning tends to be in the eye of the beholder. The phrase ‘I think therefore I am’ has a simple structure, but profoundly complex meaning. The work in this thesis starts from the viewpoint that the reader does the understanding and the medium, either paper or speech, has to present the information in a usable and understandable manner that allows the listener to derive meaning.

Text is essentially linear and is read from left-to-right. Text is broken into chapters, paragraphs, sentences, words and characters. Adequate access to such information can be gained by simply moving through a document line-by-line, word-by-word and character-by-character. Most word-processors and editors even allow movement at higher order structures through keyboard commands or document outliners and style-sheets.

Products such as the CAPs Workstation (Bauwens, Engelen, and Evenepoel 1994) allow sophisticated browsing through document structure. However, no matter how large the document or complex its meaning, all objects within that document can be accessed in a linear, left-to-right manner. This essentially simple, linear structure can be used to define plain text as a simple information source.

An important aspect of written or spoken text is that the information is generally redundant. When listening to speech, the listener does not usually remember the surface structure of the utterance for long (Ellis and Beattie 1986). However, the gist of the information can be retained for a longer time. The important feature is that the gist is usually good enough for comprehension of the text. The fact that it is not essential to remember every item of an utterance to achieve comprehension is another factor that means text can be regarded as simple.

Algebra notation is, however, not so simple in its structure. Braselton. and Decker (1984, p276) describe, in the context of teaching reading skills, why mathematics is more complex than ordinary text:

> ‘Mathematics is the most difficult content area material to read because there are more concepts per word, per sentence, and per paragraph than in any other subject …Reading mathematics is complex because of the mixture of words, numerals, letters, symbols, and graphics that require the reader to shift from one type of vocabulary to another. To complicate matters further, examination of mathematics textbooks reveals that the math concepts presented may be appropriate to the grade level to which the books are designed; however, the reading level of the text is often one, two, or even three years above the level of the population for which the text is intended …’

This reasoning can also apply to algebra notation alone. Within a particular expression, algebra notation can use both dimensions of a paper. Text, whilst it forms a two-dimensional array on the page, is simply formed by one character horizontally juxtaposed with the next. Algebra can use sub- and superscripts before and after an item: For example, ${}^{a}_{b}c^{d}_{e}$. Fractions use vertical juxtaposition: $\frac{x+1}{x-1}$. Even when symbols are written in a horizontal line, different spacings are used within an expression: $ab + cd$. One expression can be nested within another: $a(b + c(d + e)) = f$. Finally, the range of symbols possible in algebra is enormous: Letters, numbers, Greek letters, and a vast array of special symbols.

This use of extra dimensions, spacing rules and explicit parsing systems means that an algebra expression has a richer and potentially more complex structure than plain text. An algebra expression can be simple, but these structures can be combined to an arbitrary complexity. The denseness that arises with such a rich notation adds to the complexity of algebra notation.

A profound difference between information such as algebra notation and plain text is that the latter falls naturally into a spoken form. Indeed, text is essentially written speech. This cannot be said for mathematics, program source code, tabular information or more diagrammatic structures such as trees.

The use of short-term memory for a spoken algebra expression is not reliable. Every single item within the expression must be remembered exactly. Loss of a single item can completely change the meaning of an expression or the outcome of a manipulation task. The rendering of parentheses in spoken algebra is notoriously difficult. They are either omitted, mis-placed or inserted in such a ponderous manner as to make the utterance unusable.

Thus, the presentation of algebra notation in speech has many more problems than the presentation of plain text. The difficulty principally arises from the structural complexity of the information.

1.1.2 Active Reading and External Memory

Reading complex information relies heavily on pencil and paper. Whenever two mathematicians meet, they may talk mathematics, but they will almost certainly start using pencil and paper to support communication. This is also true of the lone mathematician, who will invariably use paper to externalise many of his or her manipulations. Providing an auditory equivalent for this reliance on an external source for information and working with the information forms the core of this thesis.

The difficulty in speaking and retaining spoken algebra, or any such complex information, means that paper is an essential part of the reading process. Paper acts as an external memory; the permanence of the image on the page means that the reader is relieved of the burden of retaining the information (Schönpflug 1986). This can mean mental resources can be devoted to the comprehension of the information, rather than its retention.

The manner in which the information is presented can also help in the process of reading and understanding (Kirshner 1989). The lay-out of the information on the page can also help by prompting the reader to use procedures in the accomplishment of the task (Larkin 1989). Finally, the control and the presentation style can be combined in the visual modality to give different levels of information. One such, high-level, view can be a glance. This ability to obtain different views allows planning and flexibility in the reading process.

Visual reading is an active process, whereas listening to spoken material tends to be passive (Aldrich and Parkin 1988). The external memory can only be effective when the information it contains can be accessed with speed and accuracy. The external information source, the paper, combined with the speed and accuracy of control in selection afforded by the visual system allows the control over information flow that makes reading active.

Such control is not possible with the auditory system: A listener cannot move back and forth over the contents of an utterance to check its content. This inability tends to make the listener the passive partner in the process. Any aid for reading must reverse this situation.

1.2 The Mathtalk Program and the Maths Project

Apart from a general need to develop usable access to complex information, there is a special need in the case of information related to mathematics. Mathematics forms a vital core of school education. Along with tuition in the use of language, learning in mathematics is seen as a basic requirement in most educational systems. Indeed, it is now a mandatory part of all European national curricula in State education systems (Howson 1991). In addition mathematics, and especially the symbolic manipulation exemplified by algebra, forms a vital part of many other disciplines.

Despite its importance, many visually disabled children underachieve in mathematics at school (Rapp and Rapp 1992; Kim and Servais 1985; Stöger 1992). This is not to say that visually disabled children lack any mental ability to perform mathematical tasks or understand mathematical concepts, but more that they lack the simple mechanical means to perform those tasks. An incident from ‘Through the Looking Glass’ by Lewis Caroll (1982, p216), when Alice is quizzed by the Red and White Queens illustrates this point:

> ‘Manners are not taught in lessons,’ said Alice. ‘Lessons teach you to do sums, and things of that sort.’
>
> ‘And you do Addition?’ the White Queen asked. ‘What’s one and one and one and one and one and one and one and one and one and one?’
>
> ‘I don’t know,’ said Alice. ‘I lost count.’
>
> ‘She can’t do Addition,’ the Red Queen interrupted.

In this scene Alice finds herself in the same situation as many visually disabled children. She knows that she can perform the simple arithmetic task, but the presentation of the task prevents Alice from accomplishing it successfully. This situation leads the Red Queen to suppose that Alice simply cannot do arithmetic.

The spoken presentation of the sum ‘one and one and one and one and one and one and one and one and one and one’ is simple, but Alice lacks the means to review the information to count the number of additions. The transient speech signal means unless the whole sum is retained, its exact form is lost to her. Simply remembering the gist, that it was a sum, is not enough, as is the case with many natural language utterances. If the sum were presented on a piece of paper, an external memory, then Alice could undoubtedly do the sum.

A basic tenet of this thesis is that visually disabled children have the cognitive facilities to do mathematics to the same extent as their sighted peers. The difference is simply a mechanical one: Not having the external memory provided by the piece of paper and the control of information flow afforded by the visual system in combination with the paper, means a visually disabled child cannot adequately deal with the algebra notation.

As mathematics plays such a vital role in many disciplines, the inability to use its associated notations (of which algebra forms the core) is a disability in itself. In the wider context, the development of means by which many sources of complex information used effectively could enhance the educational and employment prospects for many people.

The Mathtalk program was written to promote active reading of algebra notation and to explore the design of the user interface that allowed such an interaction. The Mathtalk program was developed to evaluate and demonstrate the design principles derived from this work.

As the Mathtalk program was written to test user interface design issues, its presentation of algebra notation is not complete. Enough of the notation is translated into a machine representation so that the core of algebra notation can be presented, to an arbitrary complexity.

The Mathtalk program was developed in three stages. First, a general presentation style was developed, that is, the spoken output. This was used to explore the first design question of how to present the information. The second component was to add browsing. This was used to explore how best to add control to the reading process to make it active. Finally, an audio component was added to the Mathtalk program that added an audio glance, designed to allow planning of the reading process.

The Mathtalk program only allows reading of algebra notation. Its restricted domain was designed to allow development of a user interface that enabled an active, usable reading interaction to take place. To allow only reading, and that only in the auditory mode, is not enough. To facilitate the use of mathematics in education and employment by the widest possible range of visually disabled people, both the reading and manipulation of algebra must be allowed in a variety of interaction modalities.

The Mathematics Access for Technology and Science (Maths) project was set up to further this goal. The success of the work presented in this thesis led to the setting up of this project. The Maths project is a European Union funded project under the Technology Initiative for Disabled and Elderly People (Tide) and seeks to develop a multi-modal algebra workstation for visually disabled school-children.

1.3 The Wider Field

Research into the design of computer user interfaces for visually disabled people can be divided into two main subject areas. The first is the design of adaptations to mainstream applications to make them available to visually disabled users. Complementary to this is the design of specialist adaptations specifically for the visually disabled community.

The primary example of the first are screen readers. These are pieces of software that make the information present in a visual interface available in a different modality: Either braille or speech. Screenreaders attempt to follow the flow of control around the display and also allow the user to explore the display.

These adaptations are general because they attempt to allow access to any information presented by programs within a particular operating system. This allows visually disabled people to use the same software products, for example word-processors, data-bases and spreadsheets, as their sighted colleagues.

Specialist software is produced to fulfill needs, perceived or real, not catered for by either general adaptations or mainstream software. Specialist versions of mainstream software such as word-processors can be written, that may cater more exactly for the needs of visually disabled users. An example of this would be the Vincent Workstation (Vincent 1982), a dedicated hardware and software combination for basic computer applications.

Specialist software can also be written when a need in the visually disabled community does not exist within the sighted community. Examples of this are Soundgraph (Edwards and Stevens 1993) a product for the writing and reading of simple line graphs in sound or the use of infra-red spectra in sound (Lunney and Morrison 1981).

The Mathtalk program is an example of this type of specialist software. There is a need for the reading and manipulation of mathematics in an equivalent manner to that seen with pen and paper. In the narrowest sense, such software is not needed by a sighted school-child, as pen and paper already exist. Thus, special software needs to be written to fill a gap left in mainstream software.

Where software does exist for presenting and manipulating algebra it generally performs automatic symbolic manipulation. It is, therefore, unsuitable for the typical teaching situation. Pre-existing software is also inaccessible because of its presentation mode. The special layout required by algebra and the diverse symbols used, mean that screenreaders are unable to cope with such a complex information source. The design principles presented in this thesis could be included in screenreaders, given the internal format of the algebra is in some standard form, to bring rendering of mathematics into general adaptive software. This approach has been adopted by the Maths project, which will use the Standardised General Mark-up Language (SGML) as its standard internal format. Specialist modules of the screenreader will be able to present algebra and allow manipulation and input of algebra notation when it exists in a suitable format.

General and specialist software typically use either braille, speech or both to provide output. Many factors govern the choice between the two. The most important of these is the preference of the end-user. The choice of synthetic speech output as the medium to explore the reading of complex information was not based on a view that speech is better than braille. In some senses, the choice of speech was prompted by the worse provision for mathematics in audio form, combined with increased flexibility of speech. The following factors influenced the choice:

1. Braille codes exist for the presentation of algebra notation (BAUK 1987; Nemeth 1972). In this sense a braille reader is in a better position than a speech user. Whilst not attempting to trivialise the problem, once algebra is contained within the computer in a form that captures all the relevant information, the rules already exist for its presentation in braille.

2. In braille, the information exists permanently on the display and the reader is in active control of the information from that display. In many ways a visually disabled person can already read algebra notation or other type of complex information given that it can be presented in braille.

3. Many visually disabled people use speech output in order to use computers (Edwards 1991; Griffith 1990). If not because of preference, this may be because that user cannot read braille. This means there will always be a need for a usable access to complex information via synthetic speech.

4. A usable method for reading complex information, especially algebra, does not exist for speech as it does for braille. The lack of a permanent display, the resultant load on memory, and the passive nature of the interaction mean that reading complex information by listening is not currently possible. This absence of presentation methods and inherent problems with reading by listening, coupled with the wide use of speech as a presentation medium means that the research into how to accomplish reading by listening is needed.

5. Speech synthesis is a fast, flexible and relatively inexpensive form of output.

6. There are some positive aspects of speech that may be taken advantage of by a designer. Speech can contain more information than that present in the words alone. All speakers know that they can alter the meaning of what is said by how they say it. This feature, known as prosody, is explored in this thesis as a mechanism to improve the presentation of complex information.

7. Languages are rich in symbols and and so offer more ways of expressing information than is possible with a finite set of tactile symbols.

The view taken in the Maths project is that a truely multi-modal interface offers the best solution. A user can then use the modality best suited to a particular task or process and one mode may support or complement another.

Other types of complex information have been investigated. A brief description of these will put the work on algebra notation into context. Much of the effort into the design of software for visually disabled people has concentrated upon the adaptation of GUI, especially Microsoft Windows, for use by visually disabled people.

This is not only a technical problem, but also one of design for a complex user interface (Mynatt and Weber 1994). The spatial display of windows, icons and menus in multitudinous configurations leads to great complexity in visual presentation. Consequently it is difficult to render such a display in a usable manner in the auditory modality.

This complexity has a different nature to the complex textual information that is the subject of this thesis. However, the investigation of rendering complex textual information in speech can inform the design of such complex, general user interfaces.

One of the aims of the work in this thesis is to increase the information content of synthetic speech without increasing the quantity of speech. For all sorts of complex displays there is a danger that increased complexity simply means more speech. Packing more information into the speech and providing control over the flow, together with overviews of that information, should make such complex displays easier to use.

Some interest has also been shown in the display of algebra notation in speech. Raman (Raman 1994a; Raman 1991; Raman 1992) wrote the ASTER program to provide audio renderings of technical documents that include mathematics. The ASTER program can take documents written in the typesetting language LaTeX (Lamport 1985) and allow browsing of the document structure rendered in speech. Raman's work has concentrated on the extraction of information from the typeset document and the provision of tools to facilitate audio formatting. The Mathtalk project has concentrated on the other end of the problem: How best to use speech, non-speech audio and browsing to allow active and usable reading of such complex information. Both approaches are essential for the facilitation of usable interfaces for reading. Without the tools and internal representation of the information it is not possible to generate a user interface that allows that information to be read properly.

1.4 A Definition of Terms

1.4.1 Visual Disability

The terminology used for the target population of end-users of this work deserves some definition. The term *visually disabled* is used as a generic phrase for all those people who have either little or no vision or restricted vision not corrected by human artifice. The work in this thesis does not address those visually disabled people who have useful sight and would therefore make use of some kind of enhanced visual display. These people would be described as *partially sighted*. The target end-users of this work are visually disabled people described as *blind*. Blind people are taken to be those with little or no useful vision, who would be either registered blind in the UK or be legally blind in the USA. The age of onset of visual disability is of no concern in this work, so the definition of terms relating to etiology will not be discussed.

1.4.2 Algebraic Definitions

Many of the labels used in this thesis for algebraic objects are colloquial in nature. These are used in preference to more exact terminology as they are the names used in the classroom by both teacher and pupil. The label *term* is the prime example: A term is colloquially defined as a set of operands between printed operators; a strict definition would be the terminal nodes of a tree representation of the same expression. The latter would have little meaning to most in a classroom, where the former would be in common use. The list below contains only those labels that are given such colloquial definitions.

Term A term is a group of operands contained between printed operators (usually of least precedence). For example, $ax^2 + bx + c = 0$ has four terms, ax^2, bx etc.

Expression The complete expression of a mathematical idea. The widest grouping of the set of algebraic symbols combined together according to the rules of algebraic precedence. An equation is a special case of expression that contains an equality operator.

Sub-expression A sub-expression is a group of terms contained within parentheses. For example, $3(x+4) = 7$ has a sub-expression $(x+4)$.

Fraction Two sub-expressions vertically juxtaposed separated by a fraction line. For example, $\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ is a fraction with the expression $-b \pm \sqrt{b^2 - 4ac}$ forming the *numerator* and the expression $2a$ forming the *denominator*.

1.4.3 Usability

Usability is a term used frequently in this thesis. The objective of this research is to increase the usability of reading by listening. The definition of usability is taken from the draft usability standard ISO-9241 (ISO-9241 1993). It takes three measures to define usability:

Effectiveness refers to the accuracy and completeness with which intended goals are achieved. It also encompasses the flexibility of the product to the user's needs.

Efficiency is a measure of the amount of human, economic and temporal resources that are expended in attaining the required level of product effectiveness.

Satisfaction is the immediate (ease of learning) and long term (ease of use) comfort and acceptability of the overall system.

A useful view on usability was given by Stig Becker when he said 'access is not usability' (Edwards 1993). Simply making algebra notation accessible is not enough to promote greater achievement in mathematics education by visually disabled people. Access needs to be provided in a way that is effective, efficient and satisfying in a context of use that is appropriate to school mathematics education.

The work in this thesis does not follow the draft usability standard ISO-9241, but does so in spirit. Work at the beginning of Chapter 3 defines a context of use by examining the role of algebra notation in reading and the nature of the reading process. Great effort was put into making the speech presentation more effective in presenting the information in the notation; the attention to mental workload addresses efficiency and satisfaction; and the focus on giving the reader control over information flow to make him or her active also address the notion of efficiency and effectiveness.

1.4.4 Musical Notation Used in the Thesis

In Chapters 4 and 5 some musical notation is used to describe the pitches of notes. There are eight octaves of seven notes in the western diatonic system (Scholl 1993). There are many different systems for notating pitch. The one used in this thesis is described in Scholes. In this commonly used system a note, for example ’C’, is followed by a subscript octave number, for example: Middle C (216 Hz) is C_3 and A above middle C (440 Hz) would be A_3.

1.5 Thesis Aims

The design principles produced in this thesis aim to provide designers with the means to enable active reading of complex information by blind people using an auditory presentation. The aim is to provide the listening reader with the necessary control over information flow to make him or her the active partner in the reading process. In addition, they aim to give the reading interaction some of the qualities of the external memory used in visually reading algebra. Generally, the design principles present a exemplar for the effective use of speech and non-speech audio in the computer-user interface.

This thesis concentrates on how people will actually use information held in machine readable form, rather than simply describing how it can be held in that form and thinking the task is complete. This means the thesis concentrates on the design issues at the user interface. Four design questions can be formed to drive the design process in a speech based auditory interface for reading algebra notation:

1. What information to present? An important first question is what information is contained in the display being read and what information or knowledge the reader brings to that interaction. The temptation to use a spoken presentation to ‘read’ to a blind listener should be avoided. Reading to a blind person leaves that listener passive and not a true reader.

2. How to present that information? Having decided what information to present, the next stage is to render that information in such a way that it captures some of the qualities of an external memory.

3. How to control that information? To become active, the listening reader needs to be able to select information with speed and accuracy from the page.

4. How to plan that control? To be effective and efficient in the reading process the reader needs some foreknowledge of the information to be read in order to plan his or her reading. This is accomplished with an overview.

The techniques used to answer each of these questions give rise to the design principles for the listening reading of complex information.

The proposed set of design principles should increase access in a usable fashion. To help ensure this outcome, each of the techniques used was evaluated experimentally. This gives the design

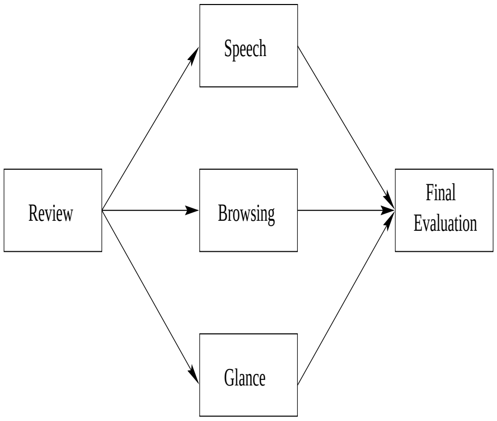


Figure 1.1: The structure of the thesis in summary form.

principles a solid foundation by demonstrating that they have the required effect.

1.6 Contents of the Thesis

Figure 1.1 shows the overall structure of the thesis. Each chapter is presented in summary below. In the review chapter, an investigation of the reading and listening process are undertaken to provide a notion of what the design principles have to achieve. Out of this review the potential of prosody to improve the display; the potential of browsing and the need for a glance emerged.

The next part of the thesis deals, chapter by chapter, with each of the components of the Mathtalk program designed around these ideas: The speech and prosodic component; the browsing component and the audio glance component. These three components are drawn together in the final work chapter for an evaluation of the integrated Mathtalk program. In this chapter, the general applicability of the design principles to enable active reading were tested with a paper design for another complex information source. In the last chapter, a summary of the thesis and its contributions to the field are discussed.

Chapter 2 forms the background to this thesis and reviews potential solutions to the problems of reading by listening. The chapter starts with a description of the process of visual reading, with special reference to reading algebra notation and the form of the print on the page. The visual reading process is then contrasted with the process of listening to speech. In the design of tools to assist the reading process, it is important to understand the essential characteristics of the processes of reading and listening.

What is known about the experience of visually disabled children with mathematics is described. Other solutions to presenting algebra in speech are described to set the design of the Mathtalk program in context.

Two topics are then proposed as potential solutions to the problem of reading algebra by listening: The prosodic component of speech and the use of browsing to control selection in computer displays. The review reveals that prosody is able to indicate the structure of an utterance and improve the memorability for speech. The current knowledge of algebraic prosody are also reviewed. The nature and components of the process known as browsing are reviewed for what they can bring to the design process.

The notions of using prosody and browsing form the core of the first two work chapters. Chapter 5 returns to the improvement of the presentation of complex information in speech with the development of an audio glance.

Chapter 3 investigates the questions of what information in algebra to present and then how to present algebra in speech. Separating what information is present on the page and what knowledge the reader brings to the reading interaction forms the core of the design process. The presentation in Mathtalk is based upon the principle of non-interpretation of what is printed on the page. If Mathtalk is to emulate visual reading, then it is the user who must do the reading, not the computer.

First, a current method of disambiguating the structure of an algebraic utterance is presented. This involves the insertion of lexical cues that name and delimit constructs within an algebra expression (Chang 1983). General rules are presented for this method, together with some potential criticisms of the method.

The chapter continues with an investigation of prosody, which may avoid the problems of the previous method. First an investigation into extension of the rules for algebraic prosody is described. This chapter concludes with an evaluation of the effects of adding prosody to synthetically spoken algebra and a comparison to the lexical cue method. Prosodic cues were found to improve recovery of structure; enhance retention of content of an expression and to reduce the mental workload associated with the listening process, in comparison to the standard method.

Chapter 4 describes the design of a fast and accurate means of control to give active reading of algebra notation. That is, answering the question of how to control information flow. The first stage of the design process was to draw out the nature of the browsing needed. Once a structure based browsing had been chosen, the moves and objects within the browsing design were discussed. The rest of the chapter describes a series of iterations of design and evaluation of browsing functions and the browsing language that mediates the control. The chapter concentrates on the design of the functionality of low-level moves and a mediating language that can combine these moves into higher-level tactics and strategies with appropriate feedback that keeps the reading as the top priority.

Chapter 5 returns to the topic of improving the presentation of complex information in audio. This chapter describes the design of an audio glance at the structure of complex information. This was a solution to the problem of how to enable planning of the reading process.

As little information exists about glances, the chapter first discusses the nature of a glance and forms a definition. Then what is needed in a glance at algebraic structure is discussed. A design for *algebra earcons* is presented. These combine the use of prosody to indicate structure with the design of non-speech audio messages called earcons to provide a glance.

After presenting detailed rules for the construction of algebra earcons, their utility in presenting structure at a glance was experimentally evaluated.

Chapter 6 presents the evaluation of the integrated components of the Mathtalk program. All of the work in this thesis relied on evaluation to validate each of the components for usability. The efficacy of the whole design to promote active reading was also validated empirically.

The co-operative evaluation method (Monk, Wright, Haber, and Davenport 1993) was used to collect both quantitative and qualitative measures on the performance of blind participants undertaking algebraic tasks. The evaluation was used to gauge how well the participants read the algebra expressions, rather than their performance at the tasks. The Mathtalk program does not teach mathematics, but was designed to make the listening reader more effective, efficient at reading algebra and to find the interaction more satisfying. Improvement of the user's algebraic skills, will, hopefully, be an indirect result of this increased usability.

This final work chapter concludes with a paper design, using the design principles derived from the Mathtalk program, for an interface to another source of complex information. This was the Treetalk program. Treetalk should enable a similar active reading of syntax trees for phrase structured grammar analyses of English sentences. This paper design was used to demonstrate the general applicability of the design principles derived from this work.

Chapter 7 summarises the contributions of the thesis, discusses its limitations and suggests some areas for further work.

This thesis has moved the design of auditory interfaces for blind computer users into new areas. Instead of only being able to use computers to access plain text or have reading machines 'speak' at a passive listener, the listening reader can now become the active reader of complex types of information. The extensive use of prosody in the interface offers a fruitful path to designers of user interfaces including synthetic speech for both visually disabled users and the wider community.

Given the appropriate structural information, the designer can make a synthetic speech presentation much more usable by the insertion of prosodic cues. The use of prosody also led to the development of, for the first time, of an audio glance, that is able to offer different views of complex information to a visually disabled reader. The emphasis on active reading, via browsing, based on the themes of external memory and control of information flow propose a novel attitude to the design of computer based tools to enhance the abilities of visually disabled children and adults in education and work.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the literature pertinent to the development of a user interface that facilitates the reading of standard algebra notation by listening. In Section 2.2 certain aspects of the reading process are discussed. The review concentrates on the mechanical rather than the cognitive aspects of reading; that is, the input of information from the external world, rather than the understanding of that information. Then Section 2.3 describes certain aspects of reading by listening and how this relates to the process of visual reading. Part of this section deals with the process of listening to synthetic speech, which is the chosen medium of output for the reading tool. From this part of the review the twin design foundations of this thesis emerge: That is, *external memory* and *control* over information flow.

A small amount of literature exists on the visual reading of algebra notation. This is reviewed for what it tells the designer about the tasks involved in the reading process and the mechanisms involved in that process.

In the second part of the review, the current solutions for reading algebra notation with speech are described. The problems and benefits of these solutions are described.

The abstraction of the problem into external memory and control suggests two avenues that could provide solutions. The prosodic component of speech has the potential for adding some of the features of a printed algebra notation into the impoverished synthetic speech presentation.

A simple solution to the poor control of information flow is to allow the listening reader to browse the display. The literature describes the browsing process and it can be seen to be related to the selection of what is to be read during reading. This description also helps the design by describing what must be included to make browsing effective. From this material some high-level goals and needs of the browsing component are extracted.

2.2 Reading

It is important to realise the differences between visual reading and the listening reading so often used by blind people, obvious though they may seem. What will emerge is the role of paper and the control of information flow afforded by the visual system in the reading process. Having determined the characteristics of the visual reading process, these characteristics can be emulated in the design of an auditory interface for listening reading.

Rayner and Pollatsek (1989) describe reading as: ‘reading is the ability to extract visual information from the page and comprehend the meaning of the text’ (Rayner and Pollatsek 1989, p23). The reading process can be divided into three broad domains:

1. The input of information from a physical, external source, into the reader’s memory via the visual system;
2. the recognition of words and their integration into higher level structures such as sentences;
3. the process of understanding what has been read.

A large body of literature has been written on the deeper, psychological aspects of reading: word identification; the speed of that identification; the building of syntactic and semantic representations on the way to a comprehension of the text.

In this discussion the greater part of the information has been obtained from Rayner and Pollatsek (1989). The information on eye-movement is uncontroversial and similar figures and descriptions may be obtained from other texts (Ellis and Beattie 1986; Hulme 1984). Rayner and Pollatsek take a particular stance about the role of inner speech and sub-vocalisation that may not be shared by other authors. However, as will be seen below, their outlook initiates a particular design attitude in the development of user interfaces for listening reading. The wider acceptance of their views in the psychological community and the data on which they are based are not strictly relevant to this discussion.

It seems that there is a point of convergence for the processes of visual reading and listening. The processes before this point of convergence are the real differences between reading and listening and will indicate what the design solutions should tackle.

Readers are often aware of a voice inside the head articulating what is being read (Rayner and Pollatsek 1989). This inner speech is thought to have two components: Sub-vocalisation and a phonological code. It is the latter which is of interest in this discussion. The phonological code is the auditory image kept in working memory during reading. Written material is thought to be converted to this speech based representation. The phonological code contains all the features of articulated speech, including suprasegmental features of rhythm, pitch, duration and stress. The role of inner speech is somewhat speculative. Rayner and Pollatsek state that 'Some proponents of inner speech have argued that reading is little more than speech made visible' (Rayner and Pollatsek 1989, p190). A phonological representation of what has been read may be a point of convergence of the processes of listening to speech and reading written language.

It is thought the iconic input from the visual system is converted to an auditory representation after word identification. This is, presumably, an equivalent representation gained from the spoken version of the same text. From this point onwards the processes of comprehension for both listening and visual reading are unlikely to differ significantly. If this is true, then it is only in how the information is gathered from the external world, and processed to this point of convergence, by which visual and listening reading differ.

So, it can be seen that listening and reading differ in source and input mechanism, rather than at any deeper level. These differences may be termed the mechanical, rather than cognitive/comprehension aspects of the reading process. What are the consequences of this? What is it about eyes and the printed page that makes it such a powerful combination for reading? A passing, but telling phrase from Rayner and Pollatsek is: 'In vision, of course, the eyes are a major device for selection. You point your eyes at those stimuli you want to process and ignore others' (Rayner and Pollatsek 1989, p1).

2.2.1 External Memory

It is the combination of eyes and the printed page, the seemingly easy selection of information that is the key feature that is missing from listening reading. In this thesis the process of selection will be called *control over information flow*. The substrate over which the gaze moves will be called *external memory*.

The internal representation formed during reading has a limited capacity and lifetime (Baddeley 1990). If the text has not been comprehended and stored in a more permanent fashion, the information must be refreshed. The reader must often resort to reviewing the external memory, if it exists, to recover the information lost from short-term or internal memory. Through this seemingly effortless selection process the visual system is able to review information that would be lost to the auditory system.

This control is only possible because of the printed page. The paper forms an external memory (Schönpflug 1986). Internal memory is knowledge in the head (Norman 1988). It is the reader's short and long term memory. If the knowledge in question is in short term memory and is lost, then that loss is permanent unless it can be refreshed from another source, for example, long term memory or an external source. External memory is knowledge in the world. It is present as a stimulus to be processed or ignored by the reader. The salient feature as far as reading is concerned is the permanence of the external memory. The print is on the page to be read and re-read as the reader chooses.

Zhang and Norman (1994) describe the features of an external representation of memory as:

1. External representations can form memory aids. The problem states exist in front of the reader in the form of diagrams or physical objects, so do not need to be memorised.
2. External representations can provide information that can be directly perceived and viewed without being interpreted and formulated explicitly.
3. External representations can anchor and structure cognitive behaviour. The physical structures in external representations can constrain the range of possible cognitive behaviours, in the sense that some are allowed and others prohibited.
4. External representations change the nature of a task. As some information is held externally, internal resources can be devoted to other tasks.
5. External representations are an indispensable part of the representational system of any distributed cognitive task.

This availability of an external source is not true, in general, for the listener. The speech signal is transient and presented serially to the listener. The consequences of these concepts are discussed in Section 2.3. It is, in reality, not the serial nature of the presentation that is the problem, it is the tempo. Visual reading is serial, it is just that the visual system is able to move rapidly over the external memory: It is in control of the information flow. For the listener, presentation is generally slower and the recipient has no active control over the flow of the speech.

Reading is an active process. It is the reader who chooses what to read, when to read and at what pace the reading proceeds. For active reading the roles of external memory and control are inseparable. The eyes cannot give active reading without substrate over which to move the gaze and the external memory is of little use if it cannot be accessed with speed and accuracy.

Topic	Fixation Duration (ms)	Saccade Length (mm)	Regressions	WPM
Light fiction	202	9.2	3	365
Mathematics	254	7.3	18	243

Table 2.1: Comparison of eye movements during the reading of light fiction and mathematical text. Saccade length is measured in character spaces. Regressions are measured as the percentage of fixations that were regressions.

2.2.2 Characteristics of Visual Control

A description of how the eye moves around text is illustrative of the fine control it has over the information flow. A skilled reader typically reads at about 250–300 words-per-minute (Rayner and Pollatsek 1989; Hulme 1984).

The eye does not move with a continuous flow over the page. Reading progresses in a series of saccades (jumps) and then fixations. The saccades move the point of fixation in accordance with how much information has been or can be apprehended. Forty nine percent of reading time is taken up with fixations. The rest of the time is taken up with the selection of which portion of the text to next fixate and the move to that location.

As well as fixations and saccades there are movements known as regressions and skips. During the reading of a line the eye sometimes makes a regression, a movement backwards, to refixate some material. Return saccades occur when the eye moves from the end of one line to the beginning of the next. Short, common words on the page can be skipped during reading, that is they are not fixed, though they are comprehended.

More regressions typically take place during the reading of complex information. During the reading of complex texts the rate slows down. The duration of each fixation is longer and the number of regressions increases. Rayner and Pollatsek compared eye movements during reading texts on different topics. They found that the informational density of the text determines how fast the eye moves. Table 2.1 shows a comparison of the different types of eye movement in reading light fiction and mathematical text.

Reading rate is much slower for the more complex ‘difficult’ mathematical text; fixation duration is longer; saccade length shorter and the number of regressions greater. This study is illustrative of the fine control that is needed when reading complex, informationally dense material such as mathematics and how the visual system is capable of such a task.

During reading the eye does not usually select the wrong portion of text to read. Visual cues within the print, word, sentence and paragraph boundaries, allow reading to proceed without missing portions of text and having to go back and sort out the problems. Readers can skip short, common words and avoid redundant letters in long words.

This control is an inherent part of active visual reading. The eyes afford the sighted reader control over the information flow from the external memory. This control has two components: speed and accuracy. Normal visual reading is faster than listening (see Section 2.3). Some skilled listeners, particularly visually disabled users of speech synthesisers, can listen at up to 400 words per minute (Vanderhieden 1989). However, this must be compared, not to normal visual reading rates, but to speed readers whose rates can be measured at thousands of words per minute (Rayner and Pollatsek 1989).

The listening reader will need such control over the flow of information and this has to be part of the design goals. Such a task as that given above shows the utility of an external memory. Rather than having to retain the whole of even a simple equation in the reader's internal memory, the page holds the information. The control afforded by the visual system allows easy access to all the information, freeing limited mental resources for the mathematical task. How the external memory influences the reading of algebra will be revisited in Section 2.4.

In this section two features of the active reading process have emerged. These are the concepts of external memory and control of information flow, and the intimate link between the two. It is the control that makes visual reading active and that control is made possible by the external memory.

2.3 Listening

As with the process of visual reading, what is of interest in the design of an interface for reading is how information is retrieved and processed in the early stages of listening, rather than the deeper stages of comprehension. It was argued above that the processes of listening and reading meet at the stage of storage of the incoming signal as an auditory representation in short-term memory. The contrast between listening and visual reading is the reliance on this internal form. As the listener does not have an external representation to act as a memory aid, this review will concentrate on how speech is stored in the internal memory.

Some fundamental differences between the two systems have already been mentioned. Listening is typically a slower process, at approximately 180 words per minute, than the equivalent visual process. More importantly the sound medium cannot act as an external memory in the same way as the printed form. Speech is a temporal, serial medium. Speech is spread out in time rather than in space. Once a word or phrase has been heard it is lost, unless it is remembered by the listener. The ears can only hear what is currently audible, not what has been audible or what will be audible.

The auditory system has the ability to select what to listen to in the current sonic environment via the attentional system. However, the inability to scan the ‘spoken text’ to review the information leads to the major difference between the two systems. The access to print with the eyes is also serial, but the control is rapid and accurate giving the impression of being able ‘to see more than one thing at once’. Without an external memory and fast and accurate control, the auditory system cannot select information in an equivalent way to the visual system.

This means that where visual reading is an active process, that of listening to speech tends to be passive. For a given information source, the auditory system cannot control the flow of information. The listener is passive where the visual reader is in active control.

This passivity has several consequences. Aldrich and Parkin (1988) describe the differences between the use of oral and written presentations of text. As well as the speed differences described above, the listener is passive in contrast to the active reader. In consequence the listener finds it difficult to maintain attention, loses concentration and thus his or her place in the text. As a consequence there is a need to review and relocate within the spoken text. With a recording this is difficult and tedious and with direct speech it is usually impossible. These differences highlight the central role of an external memory in making the reader active and therefore efficient and effective in accomplishing reading tasks.

The transient nature of the speech signal needs to be examined more closely. For a listening reader using either recorded speech or an adapted computer display, the speech signal can be said to be permanently present. On a tape, the speech is permanently there, just as ink is permanently on the page. What really differs is the control over the information flow and the richness of the information in the permanent record.

A tape recorder enables the listener to review information, but the control is so crude (it lacks speed and accuracy) that any review of information becomes so slow and tedious that a large burden is placed on the listener’s short-term memory. Schönpflug describes the trade-off between the use of internal and external memory. For a sighted reader less effort is involved in using an external rather than an internal memory. In contrast, much more effort has to be used with a tape recorder, so the listening reader resorts to trying to retain the information internally. As described above, the visual reader also uses short-term memory during the reading process. However, when this internal representation fails, the control afforded by the visual system can take advantage of the external memory to refresh that image.

2.3.1 Short-term Memory

Short-term memory has a limited capacity. In a classic experiment, Miller (1956) showed that short-term memory could only hold 7 ± 2 items of information. These items can be single items or chunks of information related by some means. The view of short-term memory has changed since Miller's experiments. Baddeley proposes that short-term memory is divided into a visual and auditory store, controlled by a central executive. The auditory store is thought to be limited by time, a temporal store, rather than the amount of information (though the two are related). The auditory store can contain up to 1.5 seconds of information. This store is fragile and can be disrupted by incoming information.

The auditory representation can be refreshed by means of the articulatory loop (Baddeley 1990). The stored acoustic signal can be rehearsed to refresh the trace, so maintaining the auditory image. If not refreshed or committed to longer term storage, this information will be lost.

How information is received by the auditory store can affect how long it is maintained. In the speech signal, the prosodic component, can have a major effect on the memorability of the signal. The influence of prosody on short-term memory will be discussed in Section 2.6.

Two other relevant phenomena are the primacy/recency effect and the suffix effect (Baddeley 1992). In the primacy effect it is seen that the first items in a set of information are preferentially retained. In contrast the recency effect exhibits a preferential retention of the most recently presented information. This is thought to be a balance between initial processing and storage and rehearsal by the articulatory loop.

The auditory suffix effect counteracts the recency effect. It is the effect whereby recall of a list of spoken items (such as a sequence of digits) is impaired if a further speech sound is added to the end of the list. Thus the auditory suffix effect operates on the most recently stored items in a list and has little effect on the retention of items appearing earlier in the list. It can therefore be viewed as an effect which interferes with the operation of the recency effect.

As control over the information flow in the visual system is so good, any suffix effect can be largely avoided. When previous portions of an text are forgotten, the eyes can quickly be moved to review that information.

The ability to review and refresh is not the case in audition. The listener has to rely on his or her memory for spoken material and incoming material can dislodge recent material unless it has been processed. With difficult material this processing may well take longer, making the suffix effect more important.

Listeners are good at retaining the gist of an utterance, but lose the surface structure rapidly (Ellis and Beattie 1986). The gist is good enough for most natural language, but not for algebra, where the loss or rearrangement of a single item can drastically change meaning. This reasoning, taken with the fragility of short-term memory, high-lights the need for enhancing memorability for text and giving control over information flow.

2.3.2 Listening to Synthetic Speech

In this section the literature on the perception and comprehension of synthetic speech is reviewed. As synthetic speech is of poor quality compared to natural speech, knowing the limitations of its use will aid the design of a spoken presentation of algebra notation.

Synthetic voice production is modelled on relatively few of the many parameters of natural language (Luce and Feustel 1983). The resulting voice, and its comprehension, is similar to listening to natural speech degraded by noise. Whilst listeners can comprehend speech in such an environment, it is more difficult (Handel 1989). Whilst synthetic speech is more difficult to comprehend than natural speech, many people learn to listen and comprehend synthetic speech with accuracy and sometimes at great speeds (Schwab, Nusbaum, and Pisoni 1985; Vanderhieden 1989).

Much of the investigation of the intelligibility of synthetic speech has been done with lists of single words (Waterworth 1987). Comprehension depends on the quality of the speech system and varies over a wide range, approximately 99.5% for natural speech to 75% for a poor quality synthetic system (Ralsten, Pisoni, Lively, Green, and Moulinix 1991).

These measurements were made with pauses between the words. However, when the lists are read with shorter pauses between them, retention is degraded far below that of natural speech. Listeners seem to exhibit either a primacy or recency effect (Waterworth 1987). The proposed reason for this is that the limited capacity of working memory is taken up with processing the acoustic input into a correct phonological representation, or in rehearsing and maintaining material already present, but not both. The presence of either a primacy or recency effect is due to listener's strategy choice (Waterworth 1987).

Smither (1993) conducted an experiment to investigate the demands synthetic speech puts on short term memory. He tested natural speech against synthetic speech on young and old adults. His results showed that synthetic speech put a heavier load on short term memory than natural speech. Older participants performed worse than younger ones but both groups performed worse with the synthetic speech. So synthetic speech increases the already large demands on the short-term memory of the listening reader.

Ralsten et al. found that comprehension rates for single words presented in synthetic speech were greatly reduced when pauses between presentations were reduced. This probably reduced available processing time and therefore increased mental load.

It is more interesting to look at the comprehension of words in connected speech, especially the effects of prosody on understanding. '…a listener has serious problems in understanding longer messages, particularly if the materials are novel and/or syntactically complex' (Pisoni, Nusbaum, and Greene 1985) in (Ralsten et al. 1991, p472). This obviously has a significant implication for the presentation of complex information such as algebra notation.

The inclusion of pitch contour in the production of syntactically simple sentences was not found to be significant (Slowiaczek and Nusbaum 1985). However, a correct pitch contour was found to be significant when the spoken sentence was syntactically complex. The role of prosody in apprehending syntactic information is reviewed in Section 2.6.

Elovitz, Johnson, McHugh, and Shaw (1976) found that the inclusion of prosodic cues in synthetic voice output increased comprehension and listener satisfaction. Indeed when prosody features were assigned at random within the utterance a similar effect was seen. This was thought to be due to relieving the fatiguing effects of listening to the monotonous synthetic speech.

Another prosodic effect, speed, tends to decrease intelligibility (Slowiaczek and Nusbaum 1985). Increasing the speed of speech lowers intelligibility by increasing the cognitive load on a listener. One hundred and fifty words per minute seems to be optimal for a good comprehension of synthetic speech (Slowiaczek and Nusbaum 1985). In most cases it is the segmental features that play the greatest part in intelligibility, except when syntactically complex material is presented.

Another strategy for increasing intelligibility of synthetic speech is training. People who have some training with a synthetic voice develop new processing strategies for dealing with the poor quality of synthetic speech (Schwab, Nusbaum, and Pisoni 1985). With training, people are able to overcome the poor segmental quality of synthetic speech, lack of prosodic features and a high speech rate. Training can overcome the need for slower speech for adequate comprehension described above. Such effects are dramatically exhibited by visually disabled users, who can listen and comprehend synthetic speech at up to 400 words per minute (Vanderhieden 1989). It is likely that the learning strategy will remain important for most users of synthetic speech. However, inclusion of other features, such as prosody, could make this task easier, particularly when the information is complex.

The high mental workload associated with synthetic speech means that all that can be done to improve speech quality should be attempted. A facility for the control of information flow is required so that only the amount of speech that can be adequately comprehended, is spoken at any one time.

2.4 Reading Algebra

Little research has been carried out on reading algebra notation, however, one general conclusion can be drawn. This is that the form of the print on the page and an overview of the expression is important. The absolute reliance on external memory is an extreme case of that seen in general

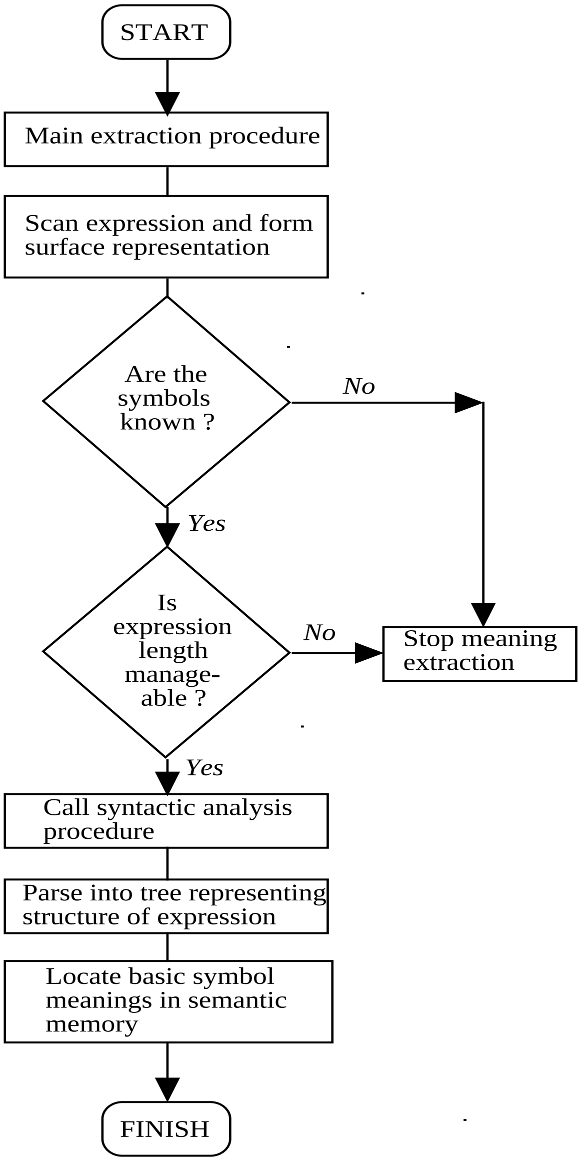

START
Main extraction procedure
Scan expression and form surface representation
Are the symbols known ?
No
Yes
Is expression length manage-able ?
No
Stop meaning extraction
Yes
Call syntactic analysis procedure
Parse into tree representing structure of expression
Locate basic symbol meanings in semantic memory
FINISH

Figure 2.1: Flow diagram for proposed model for comprehension of an algebra expression, taken from Ernest (1987). See the text for details.

reading.

Ernest (1987) has proposed a model for the understanding of an algebra expression. The initial part of this model is useful in putting the reading of algebra notation in context. Ernest's model for the reading of an algebra expression is shown in Figure 2.1.

Ernest proposes that the model works in the following way:

> 'A mathematical expression is visually scanned by the reader, whose gaze may rest upon the expression for a while. A representation of the surface structure of the

> expression is formed. This representation is checked for understanding, which involves checking that all symbols are known and checking that the complexity or length of the expression is manageable. If either of these two tests are failed the procedure is aborted and decision referred to a decision making executive function. Otherwise the syntactic analysis procedure is called and executed' (Ernest 1987, p345).

In the syntactic analysis procedure the main operator of the expression is located. Procedures are called to form a parse tree. The foundation of this tree are the rules of precedence of the algebra domain (Ernest 1987) or the reader's understanding of them. This representation of an algebraic expression would be determined by the reader's mental model of the algebra domain. Ernest suggests that it is this representation of the understanding of an expression which is used to guide the mathematical transformations a person wishes to execute.

It is the initial part of this process that poses difficulties for a blind reader. The ability to scan, judge complexity and fixate certain portions of an expression is difficult in speech given its transient form. The restricted capacity of working memory would mean that a large number of expressions would be too complex to manage. So this procedure would be aborted, restarted and repeated until the structure of an expression is apprehended.

How the form of the expression on the paper influences process is of paramount interest to the designer. How print algebra notation represents grouping and instantiates the order of precedence will influence the control of information flow by the sighted reader. If such aids exist, they also need to be available to the listening reader.

Kirshner (1989) investigated what he calls *the visual syntax of algebra*. The spatial properties of algebra notation were found to facilitate the parsing of expressions for many people. Kirshner describes two visual sub-systems (A and B) working within algebra notation. These sub-systems interact to facilitate the parsing of an expression.

Sub-system A consists of visually obtrusive markers and physical groupings of characters. For example in the expression

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

the parentheses provide visually obtrusive parsing markers.

Sub-system B is implicit as Sub-system A is explicit. For Sub-system B, Kirshner correlates the spacing rules of algebra notation with the order of precedence in algebra. These correlates are summarised in Table 2.2. Such cues would enable a reader to easily find the major (least precedence) operator which Ernest (1987) suggests forms the root of a parse tree. Removing these visual cues significantly reduced many people's ability to correctly parse an expression (Kirshner 1989).

Level	Operators	Visual Characteristic	example
0	$=$	Wide spacing	$a = b$
1	$+, -$	Spaced	$a - b$
2	$\times, \div$	Juxtaposition	$ab, \frac{a}{b}$
3	Exponentiation	Diagonal juxtaposition	e^x

Table 2.2: The correlation of operator precedence and visual characteristic. Higher level operators are performed first,that is, take precedence. Adapted from Kirshner (1989).

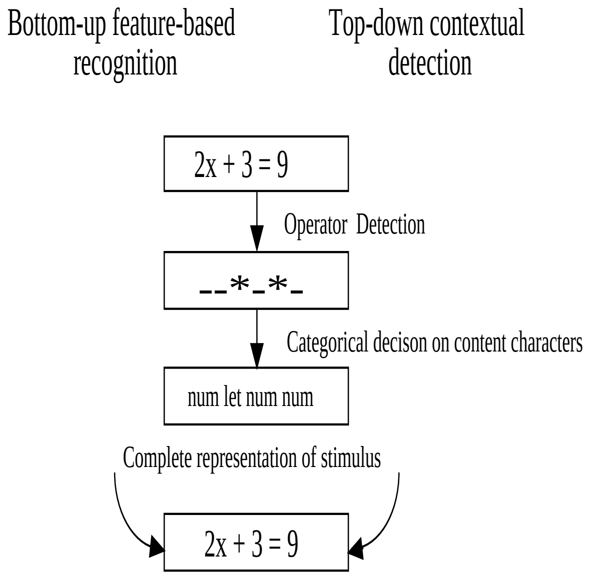

Bottom-up feature-based recognition
Top-down contextual detection
2x + 3 = 9
Operator Detection
__*_*_
Categorical decison on content characters
num let num num
Complete representation of stimulus
2x + 3 = 9

Figure 2.2: A model for the initial perception of an algebra expression, adapted from Ranney (1987). A top-down and a bottom-up process interact to form a representation of the expression. See the text for details.

So as well as acting simply as a memory, the printed expression can also aid the parsing process, fulfiling more of the roles of an external memory described by Zhang and Norman (1994). This facility should also be present in the audio rendering of an expression.

The research of Ranney (1987) ties in well with these ideas. He proposes a model for the initial perception of algebra notation, combining a top-down and a bottom-up process (see Figure 2.2). The top-down process starts with an operator detection level. The expression is scanned for operators that divide the expression into terms (cf Ernest above), presumably via the visual cues described by Kirshner. The reader's knowledge of algebra syntax and conventions enables him or her to set up categorial expectancies for characters, that is, either letters, operators or numbers etc. This process then interacts with the bottom-up feature-bound recognition process to give values within this template.

The notion of external memory and the form of presentation can be seen to come together in the description of display based reasoning given by Larkin (1989). Problem solving is quite commonly done in the context of an external display. Larkin describes a model that explicates the role of these displays, among other things, for school maths and science. Her model describes a general hypothesis about how humans use displays in solving problems. In many of the tasks she analysed, skillful use of the display seems to be the dominant problem solving process.

Larkin outlines some of the features of display based solution of a linear equation, for example,

$$-3-4(2x-9)=7+5x$$

1. It is largely error free.
2. The task is not badly disturbed by interruption, especially if one has completed writing one step before responding to the interruption. Even if interrupted within a step (see e.g. 2.1) most of us could probably recover:

   $$-3-4(2x-9)=7+5x-4(2x-9)=10 \tag{2.1}$$

   In the next step of solving equation 2.1 the -3 at the left of equation 2.1 will disappear and the 7 on the right will become a 10. All numbers are accounted for except the $5x$, which must have been left unwritten at the interruption. There is a constraint that all parts from one step must be accounted for in the next step. In this case the external visual display affords recovery from the interruption. The visual cue of the $5x$ just above the newly written line acts as a reminder, reducing the possibility of error.
3. Equation solving is commonly done in many orders. In the preceding example some of us might start by adding 3 to both sides and others by clearing the parentheses.
4. When done by skilled solvers the equation solving process is easily modified and extended.
5. The smooth easy performance of experts requires learning. When a solver looks at a display, various visible objects suggest or cue information about where they ought to be placed in order to solve the problem, for example, in solving linear equations one knows that ultimately the numbers must be on the right and a single instance of the variable on the left. The display (even a simple one such as paper) cues such knowledge. Internal strategic knowledge is cued by seeing the external display.

This description of reading algebra emphasises the need for an ability to scan and gain an overview of an expression. This ability falls easily into the description of reading as a process of control over information flow facilitated by external memory.

2.5 Visually Disabled People and Mathematics

As with the literature on reading algebra notation by sighted readers, the information base on the experience of visually disabled people and mathematics is sparse. However, two points are clear from the literature: First, for whatever reason, visually disabled people fare badly in mathematics education; secondly, the range of usable access methods for mathematics are few and generally inadequate.

In general, the experience of visually disabled children at mathematics is one of poor achievement. It is thought that both the teaching of mathematics to visually disabled children and the learning of the concepts by those children is difficult (Kim and Servais 1985). A survey in the USA (Rapp and Rapp 1992) found that 89% of visually disabled children using print took mathematics courses at grades 9–12. This figure fell to only 48% of children using braille taking similar courses. The simple conclusion is, that as soon as access to print is removed, the ability to do mathematics is severely reduced.

Rapp and Rapp place much of the blame for this predicament on the unavailability of mathematics text books in an accessible form. Mathematics in braille still has to be transcribed by hand (Wallace and Wesley 1992) and speech based solutions, other than Mathtalk, for accessing technical text have only recently been developed (Raman 1994a).

The literature has little to say about the experience of blind children and the use of algebra notation. The available literature concentrates on early mathematics (Monahan 1985). Many of the solutions presented use tactile objects and diagrams to replace materials used in mainstream education. Several reasons may exist for the under-representation of algebra. The low numbers of children reaching the higher end of mathematics education, even examinations at age 16, means the demand for algebra in mathematics may be low. The other point may be that algebra notation is a slightly more tractable problem than other aspects of mathematical education. There are braille notations available for mathematics (BAUK 1987; Nemeth 1972). Many blind school children will be taught these notations and some progress with reading, writing and manipulation of algebra can be made using braille typing machines. Many practitioners may see that provision of diagrammatic or pictorial information as therefore being more problematic and the provision of lower school mathematics of higher priority.

The thesis developed above, that external memory and control of information flow are vital for active reading, leads to a conjecture that can form the foundations of the design of the Mathtalk program. That is, it is the mechanical, not the cognitive aspects of mathematics that are problematic for visually disabled students. The principle problem that a blind student has is with accessing mathematical information in a usable manner. This does not mean that blind people are either cognitively or intellectually incapable of learning or understanding mathematical concepts or performing mathematical tasks. As the reading and manipulation of algebra depends so much on the external memory, the form of the expression on the page, and the fast and accurate control over access to the information, the removal of these supports to learning has a major effect on mathematics education.

That visually disabled children achieve as well as their sighted peers in other subject areas less dependent on complex, informationally dense information forms, would suggest that innate ability to understand and use mathematics also follows the norms of their sighted peers. Any deficit in mathematical ability is more likely to be a consequence of lack of external memory and control of information flow than any ‘non-mathematicality’ of visually disabled children.

2.5.1 Current Speech Based Solutions

One obvious method for communicating written material containing mathematical notations is to read that information onto tape. This approach is principally used for text-books, rather than exercises used in a classroom. The general problems observed with listening to taped books are likely to be exacerbated in highly technical material, such as mathematical texts. The degree of control exhibited by the sighted reader when reading mathematics (see Section 2.2) will also be needed by the listener to taped mathematics. A tape player is not capable of such fine control. In addition, unlike braille, taped mathematics offers no facilities for manipulating an expression.

The main problem with spoken algebra notation is seen to be ambiguity in the delimiting of constructs within an expression. For English two sets of guidelines are known to exist that attempt to alleviate this problem. These are provided by the Confederation of Taped Information Suppliers (COTIS), the other was written by Larry Chang (1983).

The other method for producing algebra in the speech medium is by computer generated synthetic speech. Screenreaders cannot access algebra notation displayed in a standard form. The only ways for most blind people to access algebra on a computer, with either speech or braille, is to use a linear programming language notation to represent the mathematics (Edwards 1993; Stöger 1992). Such notations can be displayed in a word-processor and accessed by screenreading software.

As well as the research presented in this thesis, there has been one other attempt to produce a computer based presentation of algebra notation in speech. This is the ASTeR program developed by T. V. Raman (1994a). The thrust of Raman’s work has been on the conversion of a machine readable representation of technical information into a form that can be displayed and browsed using synthetic speech and non-speech audio. The ASTeR program, and its relationship to this work, is described in detail in Section 2.5.1.

Algebra Spoken on Tape

Chang (1983) offered a comprehensive methodology for the presentation of mathematics in speech. These guidelines were devised, not only for recordings of human readers, but for any future applications using synthetic speech to render algebra or speech recognition for writing algebra in a computer.

Chang described the need for a set of rules in the following way:

> ‘Mathematical material is primarily presented visually and when this material is presented orally it can be ambiguous. While the parsing of a written expression is clear and well defined, when it is spoken this clarity may disappear. For example,“one plus two over three plus four” can represent the following four numbers, depending on the parsing of the expression $\frac{3}{7}$, $1+\frac{2}{7}$, 5 or $5+\frac{2}{3}$. However, when the written form is seen there is little doubt which of the four numbers it represents. When reading mathematics orally such problems are frequently encountered. Of course, the written expression may always be read symbol by symbol, but if the expression is long, or there are a cluster of expressions, it can be very tedious and hard to understand’ (Chang 1983, p1).

Chang’s method involves addressing two main problems in the speaking of algebra. The first is consistency and familiarity of symbol names. The second part of Chang’s work concerns the disambiguation of structure within an expression. To avoid such ambiguity he proposes that lexical cues be inserted to describe the explicit and implicit printed cues that delimit the structures within an algebraic expression.

Chang offers a series of choices on how to delimit structures such as parentheses, fractions, superscripts, trigonometry as well as more complex structures such as matrices and constructs found in calculus.

Chang first offers a set of common mathematical symbols and states the need for consistent naming of symbols. The meaning of many mathematical symbols is context dependent, so the reader needs some mathematical knowledge to ensure a correct rendering. Chang covers this by dividing the mathematical domain into a series of topics and varying the rendering of certain symbols within those topics. Any comprehensive application for rendering algebra notation must be able to accommodate these variations in as transparent a manner as possible for the user.

The main thrust of Chang’s rules are to present the structure of an algebraic expression in as unambiguous and usable way as possible. Though not explicitly stated as usability, the rules aim to give a rendering of an expression that flows in a way that that is both easy to listen to and speak.

The basic approach is to name each symbol in turn. The expression $3(x+4)=7$ would be spoken as '3 open parenthesis x plus four close parenthesis equals seven'. Such a rendering is unambiguous, but as Chang himself noted it is overly long due to the clumsy words such as 'parenthesis'. A rendering such as '3 times the quantity x plus four end quantity equals seven' is shorter and flows more easily. The third level offers more interpretation with the rendering 'three times the sum x plus four end sum equals seven.' A fraction is delimited in a similar manner, with the lexical cues 'the fraction', 'numerator', 'denominator', and 'end fraction'.

The increasing interpretation used in some presentations becomes more pronounced when comparing the renderings of

$$\frac{dy^2}{dx^2}=2x$$

which could be rendered as either 'the fraction numerator d y super two denominator d x super two end fraction equals two x' or the second derivative of x with respect to y equals two x.' The second approach flows more easily and may be what listeners are used to hearing from their teachers, though methods of speaking such an expression can vary widely. The drawback is that the speaker (either human or machine) needs to recognise an expression as being calculus in order to achieve the second reading.

Chang offers a similar range of possibilities for another major construct, namely the superscripts. These vary from the cumbersome 'exponent', through 'to the', which both indicate that the superscript causes exponentiation and finally a simple descriptive use of 'superscript'.

Some of Chang's rules can cause ambiguity in the rendering. For example, Chang suggests speaking the following expression $\frac{a}{b}+\frac{c}{d}$ as 'the fraction a over b plus the fraction c over d'. This could be misinterpreted as $\frac{a}{b+\frac{c}{d}}$ In another example, Chang does not close sub-expressions unambiguously: $(a+b)^2+(c-d)^2=r^2$ 'the quantity a plus b squared plus the quantity…' Such a rendering may be interpreted as a nested sub-expression containing exponents, rather than a product of two sub-expressions, each with an exponent two.

Chang offers an intuitively simple method for rendering algebraic structure unambiguously. His 'rules' are suggestions for how mathematics should be rendered in speech. The approach varies from simple description of structure to a full interpretation of the mathematical intention of the sentence. For a computer presentation of algebra the latter approach is fraught with difficulties. Semantic tags indicating the intention of the expression would have to be included in the machine's representation to allow such a rendering. Chang's guidelines are also somewhat flawed, as they can lead to structural ambiguity. To be useful, a more rigorous set of rules will be set up to delimit algebraic constructs.

Chang implicitly tackles the general usability of insertion of lexical cues by attempting to use cues that are short and simple. He also tries to reduce the number of cues wherever possible. This leads to severe ambiguities in some of the renderings presented in the method. In Chapter 3, a subset of Chang's rules are presented for the range of algebra presented by the Mathtalk program. A rigorous set of rules are presented in an attempt to avoid such ambiguities.

The ASTeR program

The work of Raman on the ASTeR program (Raman 1992; Raman 1994b; Raman 1994a) provides a useful complement to the work on the Mathtalk program. ASTeR has concentrated on the retrieval of technical information from a machine readable form into one that can be rendered sonically.

A tool, the audio formatting language, has also been developed so that this internal form can be rendered to the listener in any manner possible. The ASTeR program also allows movement around the information source.

Work on the Mathtalk program, however, has concentrated on the form of presentation and how the browsing should take place. Raman suggests that the form of the rendering is entirely subjective. The premise of this thesis is that such a statement is not true. It is important to find the best ways of presenting complex information sonically, if a usable reading interaction is to be achieved. This is a direct analogy of how the presentation of printed material will affect how easily it is read (Hulme 1984; Morrison and Inhoff 1981; Hartley 1980).

Raman's work has provided the basic structure for the representation of algebra notation and the tools for manipulating that expression's rendering. However the work on ASTeR provides no guidelines for the best ways of presenting complex information in the auditory modalities to the listening reader. An important facet of this bias was that no evaluation was made of any of the user interface components with potential end users of ASTeR. The research on Mathtalk attempts to address these two issues in a manner that can be generalised to many forms of complex information that need to be presented in the auditory mode.

ASTeR's Internal Representation

Information within ASTeR is represented as an attributed tree. Each node of the tree represents a level of the hierarchy in the document structure. One of the objects in ASTeR's representation is a math-object, which is used to capture the structure of a mathematical expression. ASTeR uses a quasi-prefix notation to describe mathematical structure. The prefix level comes from the style of representing operators ad operands within an expression. Instead of representing $a + b$ conventionally as infix notation (the operator appears between the operands) ASTeR's attributed

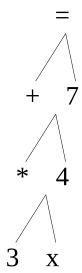


Figure 2.3: A tree structure as produced by the ASTeR program for the expression $3x+4=7$. Each of the nodes can themselves have branches to other trees that contain structures such as superscripts.

tree uses prefix notation $+ab$, where the operator appears first and applies to the following operands. This form falls naturally into a tree representation as shown in Figure 2.3. Each node within the expression tree can have one or more of a series of attributes: Superscripts, subscripts, presuperscript, pre-subscripts and accents above and below the node object. These attributes themselves can contain quasi-prefix trees.

The Audio Formatting Language

As well as the internal representation used by ASTeR, the main tool that Raman provides for facilitating audio renderings of mathematical expressions is the audio formatting language (AFL). Raman states the purpose of AFL to be: 'AFL provides for audio renderings the same power that TEX provides for visual renderings.' That is, AFL is a language that allows information to be marked up for audio display just as typesetting languages such as TEX (Knuth 1984) describe how printed material is to appear on the page.

The AFL is used to define rules for the display of information in speech, the pronunciation of that speech or non-speech audio. Thus AFL can be used to give a multi-modal display of mathematics. The AFL can also be used to give fundamentally different renderings of the underlying internal structure of an expression by manipulating the order or the detail in which the objects are rendered.

The audio formatting language can be used to describe how objects in a document are to be rendered. These rules can be used to define methods for using both speech (segmental and suprasegmental) and non-speech audio sounds. Raman describes two ways of using these sounds; as either persistent or fleeting sounds. Persistent sounds have a duration defined by the duration of some other object being rendered. For example, a voice pitch can be defined that persists throughout the rendering of a particular object. A fleeting cue lasts for as long as is defined internally for that cue. For example, a short tone can be used to indicate the initiation of a new item in a bulleted list.

Collections of rules can be gathered together into lists, which act as style sheets for the rendering of documents. Different styles can be tailored to different types of document content. Different rules also afford the user different views of a document's content; each of which is invoked using a new AFL rule.

Raman's approach is that rendering styles in the user interface are 'entirely subjective' (Raman 1994a, p59). To achieve this the user has to define his or her own rendering rules in the LISP language used to write the ASTeR program. This is in contrast to the approach taken in work on Mathtalk; that methods of presentation and interaction should be explored, developed and evaluated to ensure usability of the system.

Raman uses some of the same techniques developed in the Mathtalk program, principally as a method for demonstrating the usefulness of the AFL for developing rendering styles. ASTeR uses prosodic cues to help present the ambiguous grouping within an expression, as used for the Mathtalk program (Stevens 1991). The two other methods of presentation proposed by Raman are the use of AFL to define how different objects are described verbally and a method of variable substitution to avoid presentation of too much information to the listener. These three methods are described below.

The simplest use of the AFL is to describe how mathematical objects are to be rendered in speech. The basic style of rendering may not suit all instances of an object. As the visual cues in algebra notation are overloaded with meaning, some constructs need to be rendered differently in some contexts. Again, Raman's principle that rendering style is subjective and therefore should be configurable by the user, means that much work may be left to the user.

The second use for the AFL demonstrated by Raman was to include prosodic cues into the spoken presentation of an expression. He describes how objects such as parenthesised sub-expressions and fractions can be grouped together by pauses in the speech signal (a fleeting cue) and changes in voice pitch (a persistent cue). He also describes raising pitch for superscripts and lowering pitch for subscripts. Each of the rendering rules for prosodic presentation was defined by an AFL rule and collected into a style. Detail is given how the audio space can be divided into a series of steps to make deeply nested expressions unambiguous. However, few details are given for the rules included in ASTeR's default rendering style.

No evaluation was reported on the effectiveness of the prosodic presentation. This is important when claims are made that the audio space used by ASTeR allows up to six levels of nesting to be presented unambiguously. The rules Raman used were based on those of Streeter (1978) and O'Malley, Kloker, and Dara-Abrams (1973), which are described in detail in Section 2.6, found in human speech. That such cues can be effectively transferred to synthetic speech and are useful for the listener needs to be investigated and this was one of the important aims of research described in this thesis.

An interesting technique used in ASTeR is that of variable substitution. Formatting rules can be defined that substitute objects within an expression with simple labels. This substitution reduces the amount of material to be spoken in the first pass through the expression; the detail of the substituted object are then rendered after the end of the expression. For example, the expression

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

would be spoken, in full, as

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

However, with variable substitution the expression would be rendered as:

> ‘Capital i equals integral with respect to x from zero to infinity of f dx, where f is …’

This rendering is intended to allow the overall structure of the expression to be rendered before rendering the detail of the integrand. Like the other aspects of the user interface to ASTeR there has been no evaluation of the effectiveness of variable substitution. The approach of reducing the amount of information to be understood has a good basis, but the automatic rendering of the substitution after the overview may well negate its effects. In addition, the need for the user to define such rules in the AFL would preclude its use by all but the most expert user.

Browsing with ASTeR

The browser provides basic tree-traversal commands that allow the user to focus attention on any part of the expression or document. These can be described by the following atomic actions:

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.

Raman proposes that this small set of moves: ‘Using the above atomic actions we can define all the moves the eyes are capable of performing.’ (Raman 1994a, p81). Raman offers the example of the expression

$$Z = \frac{e^{-x^2} + e^x}{3sinx^2 + cos^2x}\,dx$$

the reader can quickly move to the denominator from the numerator by taking advantage of the layout of the expression. ASTeR’s atomic moves to achieve this task would be:

1. Mark current node.
2. Move to previous sibling;
3. Read new node;
4. Return to marked node.

The user achieves such goals by executing a series of atomic browsing moves. Whilst all possible structural moves are possible, and therefore all structure based tasks, the style of the interface will probably present some problems for many users. The example above will, in many cases be much more complex. If the user is at some point within the numerator, the user will have to make many atomic moves to reach the numerator node in order to reach the denominator. All the moves made by the eyes are possible in ASTeR, but not with the same degree of speed and accuracy in selection.

A more fundamental problem may present itself to some users. ASTeR explicitly presents the algebra expression as a tree and makes the prefix form prominent. Though the target user group of ASTeR is not made explicit, by implication it is aimed at more advanced mathematicians than the school-children that are the target of the Mathtalk program. The tree presentation of the expression will be unfamiliar to most potential users of a program like Mathtalk, especially school children, and will not be understood by many.

For example, when browsing the expression shown in Figure 2.3, the whole expression would usually be spoken as ‘a plus b equals c’. However when browsing, the first node encountered is the ‘equals’, then on the left-hand side the ‘plus’, before either of the operands. Without any evaluation, it seems to be a dubious claim that ASTeR provides all the listener needs for reading algebra notation.

Raman uses a keyboard mapping based on the cursor movements of the Unix editor VI to enable the user to access ASTeR’s tree browsing commands. These are:

j Move to child in the tree;

k move to parent in the tree;

l move to right hand sibling in the tree;

h move to left hand sibling in the tree.

As well as traversing the tree itself, the user has to be able to access the attributes present on a node. ASTeR uses the ˆ and _ keys to access superscripts and subscripts respectively. These keys were used as they are the commands to invoke these constructs in TEX. As a consequence, keys adjacent to these were used to access the other attributes.

As well as these basic browsing moves the AFL can be used, via its LISP interface, to define how some of these moves behave. Rules can be defined so that only certain objects are rendered. For example, only the expressions within a certain chapter could be spoken. This sort of ability reinforces the view that ASTeR is flexible and sophisticated, but says little about its usability or appropriateness for the basic mathematical reading tasks that need to be tackled by school-children.

2.6 The Prosodic Component of Speech

Having explored the problems encountered by the listening reader of algebra notation and some of the solutions, the rest of this chapter explores two potential solutions to the problems of poor external memory and lack of control over information flow. The prosodic component of speech offers a method of increasing the information content of spoken algebra and introducing some of the qualities of an external memory. The activity known as browsing is an obvious technique for offering control over what is spoken. Browsing will be explored in Section 2.7.

Spoken language has an abundance of information over and above the sounds that make up individual words (Slowiaczek and Clifton 1980). These features can be referred to as the non-verbal information content of speech. Every speaker of a language knows that his or her ‘tone of voice’ can carry a large amount of information over and above that in the words themselves. For example, the sentence:

> ‘Robert does research on drugs.’

Does this mean Robert is a biochemist developing new drugs or involved in more nefarious activities? By emphasising either ‘research’ or ‘drugs’ the meaning of the sentence can be altered. The same cues can be used to indicate the grouping within an utterance. It is this ability that will be exploited within the Mathtalk program to present the structure of an expression to a listener.

The phonetics of a language are the sounds that appear in that language. The phonology of a language is the set of rules that govern the use of the sounds within a language (Handel 1989).

There are sounds that are associated with the lexical content of speech, the segmental sounds, governed by segmental phonology (Lehiste 1970). There are also sounds not strictly associated with the segmental features of speech. These suprasegmental features make the phenomena known as prosody and paralinguistics.

Prosody is defined as:

> ‘The basic psychoacoustic properties of sound are the source of the main linguistic effects: pitch and loudness. These effects, along with those arising from the distinctive use of speed and rhythm, are collectively known as the prosodic features of language’(Crystal 1987, p171).

Paralinguistics literally means ‘alongside language’. It is the global effects of how something is said: whispering (conspiratorial), husky (sexy) etc. Paralinguistics give the emotional content of speech (Edwards 1991).

2.6.1 Prosody in Spoken English

Nespor and Vogel (1986) describe a hierarchy of prosodic phonology, with an utterance being broken into a series of tone units, phonological words, feet and syllables. In the following sections the basic features of prosody, and their purpose, are described.

Rhythm

The rhythm of spoken English is based on a unit known as the foot. The foot is like a bar in music. Each foot holds one or more syllables (Halliday 1970). The first syllable in a foot is always salient and carries the beat. The salient syllable is the stressed syllable. Syllabic loudness is usually referred to as stress. A foot can be one salient syllable, but may contain other weak syllables.

Halliday points out that the same sentence may have several distinct rhythms. For example (a ‘/’ denotes a boundary between feet):

- Peter spends his /weekends at the /sports club.
- Peter /spends his /weekends /at the /sports /club.

The implication is that each foot takes an equal time to speak. This is approximately true if the tempo of an utterance does not change (Halliday 1970). So the more syllables in a foot, the more quickly that foot is spoken to maintain the same tempo. Rhythm is related to timing, where timing can refer to the duration of a foot, but it is also extremely important as applied to the duration of silent-pauses within the utterance (Edwards 1991).

Tonic Prominence

The unit of intonation in English is the tone group or tone unit. The pitch contour within a tone unit or succession of tone units gives the melody of speech. A tone unit consists of a number of feet, varying in number up to seven or eight (Halliday 1970). The tone unit structure of an utterance also reflects the information structure of speech. The tone unit is one unit of information the speaker is trying to convey.

Within each tone unit there is always some part which is especially prominent. This is the part the speaker wishes to show to be important; the focus of information. This prominent part is called the tonic.

The tonic always starts on a salient syllable, that is, at the start of a foot. This is the tonic syllable. This syllable is often longer and louder than other syllables. The tonic syllable carries the majority of the pitch change within the utterance, and this makes it prominent (Halliday 1970). For example (a '//' denotes a tone unit boundary, $\underline{underlined}$ syllables denote prominence):

> '//Peter spends his /weekends at the / $\underline{sports}$ club//'

The final syllable in the tone unit can also be lengthened to aid discrimination of boundaries.

The focus of attention is *new* information. *Given* information is that which is already available to the listener. New information is that which the listener could not have supplied for him- or herself. This distinction is dependent on context. This process relates what is being said to what has been said before. New information tends to follow given information, but can be anywhere within a tone unit. If the context of an utterance is ignored the tonic usually falls at the end of a tone unit. The placing of a tonic here, spoken with a falling tone is known as a 'neutral' tone (Halliday 1970) and denotes 'default' or usual meaning.

The language is couched in a succession of melodies carried by the tone unit. The melody is made up from continuous variations in pitch or a pitch contour. These are stretches of falling, rising or level pitch. There is practically no limit to the number of pitches possible within speech and the human ear can discriminate very finely between them. This is the major difference between music and speech: Pitch within music is strictly defined and consistent; that in speech varies within and between speakers (Crystal 1987).

Intonation and Meaning

If the intonation of a sentence is changed, the meaning can also change. The possible intonation patterns are part of the speaker's grammar of a language (Halliday 1970). Just as different tenses determine meaning, so can intonation patterns. For example:

Falling tone Where are you /$\underline{go}$ing.

Rising tone Where are you /$\underline{go}$ing.

This displays a difference in attitude, the first a normal question, the second deferential (Halliday 1970).

In general tone expresses speech function, while tonic prominence expresses the structure of information. The choice of tone relates to mood and type of statement etc. Placing of tonic prominence and division into tone units relates to how a message is divided into units of information (Halliday 1970). This information structure indicates to the listener where new information lies and how it relates to that already given in the discourse. This is the message's structure.

An utterance is divided into a series of one or more tone units. The tone units are separated by pauses or perceived pauses. Silent pauses rarely exist, for instance, between words. It is more usual for syllables at the end of words and tone units to be lengthened and these are perceived as gaps by listeners (Garnham 1989). However, for the purpose of discussion in this text these boundaries will be referred to as pauses. These pauses and tonic structure indicate the information structure of an utterance. A speaker also inserts pauses during the 'planning' of an utterance, or to precede a section of complex information (Lehiste 1970; Garnham 1989).

In the context of presenting algebra notation, tone is relatively unimportant. A neutral tone would be suitable as it is only the structure of the information that needs to be presented to the listener (see Section 3.3). The picture is considerably more complex than indicated above. However this brief description serves to illustrate prosodic features in a language. What these features can add to a spoken presentation of algebra is explored below.

2.6.2 Prosodic Function

Crystal (Crystal 1987, p171) lists six functions of prosody as follows:

1. It signals the emotional attitude of the speaker;

2. It has an important role in the marking of grammatical constructs. The identification of such major units as clause and sentence often depends on the way pitch contours break up an utterance. Several specific contrasts, such as question and statement, or positive and negative, may rely on intonation.

3. Information structure: Prosody can be used to indicate what is new and given information. For example, 'I saw the new blue car'; where stress is put can indicate what is new and given

information. Whether there was a question as to the colour of the car or who saw the car can be indicated by the emphasis on either ‘blue’ or ‘I’.

4. Prosody can work over larger portions of text. Paragraphs can be given melodic shape, resulting in a prosodic coherence.

5. Prosody has psychological effects: ‘Intonation can help to organise language into units that are more easily perceived and memorised. Learning a long sequence of numbers, for example, proves easier if the sequence is divided into rhythmical chunks’ (Crystal 1987, p171).

6. The final function of prosody is as a marker of personal identity and social group.

The rationale for using prosody in the speech output from Mathtalk is its potential for improving Mathtalk’s performance as an external memory. Three of the roles of prosody listed above could be useful in this aim. The principal role of the external memory is to relieve the burden on the reader’s internal memory. In addition to this, the form of the print can determine how the external memory is used in a task.

Prosody cannot directly improve a spoken presentation in respect to holding the information externally. Prosody can, however, make spoken information easier to hold internally. If prosody can reduce the load on the listener’s internal memory, then the spoken presentation will be improved by giving it one of the characteristics of an external memory, even if in an indirect manner.

The main purpose of algebra notation is to show the grouping of symbols and the relationship between those groups. The ability of prosody to indicate syntactic structure could fulfill this role. The other aspects of prosody, such as indicating emotion, identity of the speaker and higher-order text structures, are not relevant to the aim of the Mathtalk program. It is the structure, the grouping of symbols, that the display must convey, not the intention of the expression.

2.6.3 Prosody and Memory for Speech

The ability of prosody to make speech easier to remember is based upon the rhythmic component and the chunking of speech into structurally significant subunits. Prosody gives an utterance structure, dividing it into manageable units of information, and relating those units to each other. This feature is important in the disambiguation of an utterance. The literature also alludes to the function this serves in managing the restricted capacity of short-term memory. Amongst others, Lehiste (1970) proposes that the tone unit is the basic unit of neural encoding. It is assumed that a tone unit represents one unit of short-term memory storage (Slowiaczek and Nusbaum 1985).

The phonological code, described in Section 2.2, is thought to aid the comprehension process at a higher level than recognition of words. The code is thought to organise incoming material into structures such as clauses, phrases, sentences and other meaning units (Slowiaczek and Clifton 1980). This representation is thought to aid comprehension by bolstering short term memory. Rayner and Pollatsek (1989) describe the role of the code as follows:

> ‘There are two possible ways in which a phonological coding could help in speech comprehension. Holding words and word order in working memory. Words are processed very quickly and the limited capacity of working memory would soon be overloaded if they were not chunked together in some way. Words are transformed into a phonological code and held there until meaningful chunks can be passed onto long-term memory’ (Rayner and Pollatsek 1989, p186).

It is recognised that the rhythm of speech or any sound is important in its retention in memory (Deutsch 1982). Baddeley (1992) reports on a mathematician who showed spectacular feats of memory. He could remember very long sequences of numbers with seemingly little effort. His success was largely attributed to an ability to divide a stream of numbers into rhythmic units.

Ellis and Beattie (1986) argue that the phrase or clause forms the high-level order of encoding in speech output. They extend this argument to say that the same subunits of syntactic structure form units of encoding when an auditory stimulus is transformed during comprehension. This links neatly to Lehiste’s notion of the tone unit being the basic unit of neural encoding. Tone units often correspond to clauses. If auditory memory is organised according to the prosodic parameters of speech and syntactic structure, then making the two coincide is likely to facilitate memory for speech.

There is also evidence that pauses provide vital processing time during speech comprehension (Reich 1980). Pauses afford the listener time to process and store information that is divided into significant subunits of the language. A prime example of this is spoken telephone numbers. Telephone numbers that are divided into smaller chunks of numbers, rather than an undifferentiated stream of digits, are more likely to be recalled (Waterworth 1983). The pauses between these chunks are the significant factor. Pitch contour does not significantly increase recall, though is subjectively more pleasant and thought to make the task easier (Waterworth 1983).

2.6.4 Prosody and Demarcation of Structure

As mentioned above, a function of prosody that could be exploited within the Mathtalk interface is its ability to indicate syntactic structure. The following example shows how prosodic cues can give a different structure to utterances:

- The last time we met // Robert was horrible.
- The last time we met Robert // was horrible.

The pauses, indicated by '//', drastically change the meaning of two lexically identical utterances. The utterance is divided into two different sets of tone units or phrases, further emphasised by pitch contour.

Many studies have shown that prosodic information can be used in this way to delineate clauses in an utterance (Streeter 1978; Nakatani and Schaffer 1978; Beech 1991; Ostendorf, Shattuck-Hufnagel, and Fogg 1991). Pauses or durational cues, are seen to be particularly important. Reich (1980) showed that pauses of 300 ms were likely to coincide with clause boundaries and that shorter pauses tended to be found within clauses. If longer pauses were inserted inside clauses, then the utterance became more difficult to comprehend.

As well as the division into tone units, by pitch contour and pauses, corresponding to clauses, other distinct prosodic patterns have been found. There is a tendency for the pitch of the voice to rise fairly rapidly at the start of a phrase, decline slowly throughout the phrase and then fall sharply at the end of the phrase. Where several phrases follow one another to form a sentence, the pattern is repeated for each phrase but with a steady reduction in the average pitch of successive phrases. This pattern was identified by, among others, 't Hart and Cohen (1973) who christened it the 'hat ' pattern on account of its shape.

The general fall in pitch over utterances described above is also known as the declination effect (Vaissiere 1983). The hat-effect works within the declination effect. An utterance made from a succession of tone units has a general decline, but there may be local rise-falls corresponding to the tone units. Within the restriction of the speaker's pitch range, the initial pitch of an utterance is proportional to the length of the utterance. When the length of the utterance would give rise to a initial pitch exceeding the pitch range of the speaker, the initial syllables can be spoken on a sharply rising tone. This feature could also indicate to the listener that a long utterance is imminent.

However, the situation is not as simple and clear cut as described. Prosody has more roles than indicating syntactic structure and it is not simply a component of the syntax of a language. The tone unit does not strictly coincide with any grammatical unit. However in many cases it does correspond to a clause (Crystal 1975; Beech 1991). Most research into prosodic effects in speech is based on discourse between speakers. In these situations only approximately 60% of prosodic junctures correspond to syntactic boundaries (Garnham 1989). However, in oral reading this rises to about 90% (Crystal 1975). In conversation, pauses in particular are used for purposes other than indicating boundaries. For example, pauses may correspond to planning by the speaker (Garnham 1989).

The style of presentation envisaged for the Mathtalk program is more akin to oral reading than conversation. The implication is that, given a set of rules for inserting prosodic cues, listeners could reliably use prosodic cues to apprehend syntactic structure.

2.6.5 Prosody in Spoken Algebra

A few researchers have directly addressed the subject of prosody in spoken algebra. The two major studies Streeter (1978) and O'Malley, Kloker, and Dara-Abrams (1973) investigated spoken algebra to explore the ability of prosodic cues to prevent ambiguity in speech. Algebra notation is rich in examples of truly ambiguous utterances. Unless the parentheses are inserted with special words, the expressions $3x + 4 = 7$ and $3(x + 4) = 7$ are both spoken as 'three x plus four equals seven'.

O'Malley et al. studied the recovery of parentheses from spoken algebra. They used expressions of the type:

$$(a + d)^f \tag{2.2}$$

$$(\frac{s}{l} - r) \times (t^d + e) \tag{2.3}$$

$$(a + ((b + c) \times d)) \tag{2.4}$$

In slow speech, they found that silent pauses of approximately 300 ms were highly correlated with syntactic boundaries, with a success rate of 90%–95%. Furthermore, listeners were reliable in re-inserting these boundaries. Importantly for the Mathtalk program, both experienced and naive listeners were equally capable of using such cues.

Two pauses were distinguished, a short pause and a long pause. A long pause indicated the onset of a parenthesised sub-expression. A short pause was used before least precedence operators.

O'Malley et al. produced a set of rules that successfully re-inserted pauses into expressions:

1. Pauses separate groups or terms;
2. longer pauses separate larger groups, e.g. nested sub-expressions;
3. the length of the pause preceding a group is proportional to the number of nested groups within. Shorter pauses are seen before operators within such groups.
4. Functions such as log, square root and the exponentiation operator act as if they had an implicit pause following them. The group or argument of these functions is then explicitly closed with a pause.

Streeter (1978) extended this work. She described how pitch contour, duration (pausing) and amplitude are used in parsing an expression. A series of three operands (a, b and c) were spoken in a complete set of groupings with parentheses and the $+$ operator. For example, some expressions used were:

$$a+b+c \tag{2.5}$$

$$a+bc \tag{2.6}$$

$$a+(b+c) \tag{2.7}$$

Streeter then tested listeners ability to correctly write down the expressions when heard. The recovery rate was 95%, a similar level to that found by O'Malley et al.

Digitised recordings of the expressions were manipulated, transposing groups from one expression to another. For example the $b+c$ was moved from $a+b+c$ to replace $(b+c)$ in $a+(b+c)$. A similar recovery rate for structure was found in the manipulated expressions, indicating the prosodic cues conveyed structure.

The three parameters being investigated were electronically manipulated to find which were most important in recovery of structure. They found that pitch contour was the most important cue and amplitude (or stress) the least. That pitch was the most important cue is in contrast to the findings of O'Malley et al., but they note that silent pauses become more significant at slower speech rates. Streeter, as did O'Malley et al., found that listeners were reliable at parsing *complex* algebra expressions containing only prosodic cues for syntactic boundaries.

That prosodic rules can be assigned to boundaries in algebra expressions indicates that a prosodic display of spoken algebra is possible. More importantly, these studies have shown that listeners can recover the structure of spoken algebra from these cues alone.

2.6.6 Prosody and Speech Synthesis

For a system that uses prosody in its display, two questions needed to be asked: Can prosody be added to synthetic speech and what effect does that have in the comprehension of synthetic speech?

Values for the prosodic parameters in a text representation of an utterance can be added by hand to emulate human speech. Murray, Arnott, and Newell (1988) showed that a wide range of human emotions can be generated with commercial voice synthesisers. Elovitz, Johnson, McHugh, and Shaw (1976) also showed that prosody can be added by hand with significant results on comprehension and satisfaction. The accounts described in Section 2.3 also described how higher order prosodic cues can be added to synthetic speech to increase comprehension and indicate structure.

The automatic insertion of prosodic cues into synthetic speech is more difficult. A speaker unconsciously breaks an utterance into units of information (Halliday 1970). When reading written material, the prosodic content is decided upon by the speaker. Punctuation marks give some indications (Edwards 1991). It is this sort of information most commercial speech synthesisers use to insert prosody in text-to-speech synthesis. However, the burden of the work is placed upon the reader's interpretation. As Edwards (1991) points out, this is how poetry and plays can be given different meanings by different performers.

Knowing the correct prosodic and syntactic structure of an utterance depends on knowing both the speaker's attitude and the deeper meaning of an utterance (Crystal 1975). Attempts have been made to determine rules for assigning prosody based purely on the syntax of a sentence, notably by Chomsky and Hala (see Bolinger 1972). These have failed to give a full account of prosody, leading Bolinger to state 'accent is predictable, if you're a mind reader' (Bolinger 1972).

So a full account of prosody is difficult to achieve, but if the structure of an utterance is known, a partial account of prosody can be given. If the computer's representation contains the same information, the rules described above could be assigned to the utterance. This would give prosody associated with syntax, but not that associated with the intention of the expression.

2.7 Browsing

The active nature of reading comes from the control over the information flow afforded by the form of the print on the external memory and the speed and accuracy of the visual system. The second theme of this research is to address control of information flow to take advantage of improved spoken presentation and make listening reading active.

The mechanical aspects of reading described above have an intuitive counterpart in the broad process known as browsing. Kwasnik (1992) describe how browsing encompasses the factors needed:

> 'browsing is not a passive activity…[the user] is in charge of the direction, pace, and depth of the search.'

A simple definition of browsing is *movement through an information space*. In the most general sense this is the nature of selection of what to read from a page of print. This does not mean that the process of browsing is the same as reading; it is simply a mechanism of choosing what to read. Whatever the selection mechanism, the comprehension during reading is performed by the reader.

Kwasnik (1992) put the case for choosing browsing to emulate reading succinctly, '…humans are able to invoke a variety of mechanisms to deal with poorly structured and ambiguous stimuli. One of these is the activity known as “browsing” ’ (Kwasnik 1992, p191).

As far as the listener is concerned, a spoken algebra expression may well be a poorly structured and ambiguous stimulus. Apart from the simple grouping ambiguity described above, the issue of cognitive load is important. It is not simply how much information can be delivered to the listener, but the pace of that interaction. A large expression delivered in a single utterance may well overwhelm scarce mental resources needed, not only for retention, but for integration of parts of the utterance and understanding of the content. Part of the active nature of reading is the ability to control flow of information or pace of interaction according to the difficulty of the text. An appropriately designed browsing interface will afford the listening reader the opportunity to add structure to the interaction with spoken algebra as well as to receive that information at an appropriate pace.

Browsing is a difficult skill to define. In many ways it is a self-evident behaviour (Kwasnik 1992). Browsing is described as searching, scanning, navigating, skimming, sampling and exploring. Browsing can be a ‘berry-picking’ activity (Bates 1990) or fact retrieval (Marchioni and Shneiderman 1988).

Both these activities are seen during reading, either skimming or reading for detail. For the reader of algebra notation, the notion of exploration would be useful to invoke. In Section 2.2 it was seen how much movement backwards and forwards took place when reading algebra. As well as the basic complexity of the information, algebraic manipulation tasks demand such movement.

2.7.1 Components of Browsing

Kwasnik defines the functional components of browsing as:

Orientation is the learning of the structure and content of the system.

Place marking is the marking of a view for potential second consideration. Place marks are mental land-marks.

Identification browsing relies on identification of items of interest and disinterest. This is the ability to surmise the content of a view. The view is recognised by some salient characteristics of the view.

Resolution of anomalies this occurs along with people’s desire to give structure and orientate. They do it as they go along even if they do not need too.

Comparison browsers make comparisons at all levels. They compare one item to another.

Transitions are eye-movements from one view to another:

movement to something in anticipation of a goal;

movement away after identification and rejection or after success or exhaustion of information.

These functional components of browsing fit easily into the discussion of the mechanical aspects of reading given above. These components immediately suggest several features that will be needed for the presentation of algebra.

- *Orientation* will be the main task of the browsing being designed. The purpose of the system will be to convey the structure of an expression to the reader. (Kwasnik) uses a definition of orientation that is rather strong. Oreintation is usually taken to mean the question ‘where am I?’, that indirectly needs knowledge of the structure, rather than an explicit knowledge of that structure. Nevertheless, when conveying the structure of complex information the notion of orientation will remain of great importance.
- *Identification* would also be important. If the principle of the browsing methods is to present a series of views of the expression, the reader must be able to surmise the nature of the view and fit it into a holistic conception of the expression. This relates back to the need for an overview described by Ernest.
- *Place-marking* would seem to be a useful component to enable a reader to compare different parts of an expression. By having a comprehensible presentation the *reader* will be able to resolve anomalies that exist in other media.
- Perhaps transition will be the most important aspect of this design. Transitions from one view to another would be the basic browsing moves within the system. They would represent the eye-fixations and saccades of the sighted reader. The transitions available in the Mathtalk program should be designed to present the structure of an expression.

(Kwasnik 1992) described the features of the browsing process and it can be seen that these are direct counterparts of features of control of information flow during reading. Studies of browsing activities at higher levels can also offer guidelines on the design of the control in the Mathtalk program.

Bates (1990) proposes a model of browsing based on goals and movements within the information space. A goal or purpose is attained with a series of *strategies, stratagems, tactics* and *moves*.

A move is a basic operation within the system. These could include typing a query or moving from one section of a data-base to another (Bates 1990). The counterpart in an algebra expression would be moving between the structural components of the expression.

A tactic is the choice of one move over another to approach a goal (Harter and Rogers-Peters 1985). The top levels of this model are made from stratagems, which are groups of tactics or moves to achieve a sub-goal. A strategy is the overall task the user is performing. The strategy formed to achieve this goal may be comprised of any number of stratagems, tactics and moves.

Browsing within Mathtalk was designed to enable a person to *read* an expression. As the range of structures possible in algebra is so vast, pre-empting a variety of high-level strategies suited to particular types of expression would not be realistic. Also, readers will vary in how they wish to tackle an expression.

2.7.2 Navigation in Browsing

One of the most interesting, and difficult, problems in browsing is navigation. This relates to the functional components of orientation, identification and transition. When moving through a large information space users often become lost (Norman 1988; Kerr 1990; Nielsen 1990). This is often referred to as the ‘lost in hyperspace’ phenomenon. Information spaces can be so large that users cannot orientate themselves within that space from the information given on the relatively small computer display. Following links to different sources of information leads to a loss of context.

Everyday experience of blind people forms a microcosm of this problem. When sighted users view a large information space through a small window they can lose context in the large, leaving only local context, or context in the small (Kerr 1990). When reading or listening, blind readers only have a small window on the information. The difficulty is due to the paucity of the control over information flow and the small amount of information on offer at any one time. The window must be moved around the display by the reader, a process that is slow compared to visual selection. This means a blind reader can lose context and therefore orientation very easily.

This will be true of a complex information source such as a large, complex algebra expression. Loss of orientation and sense of place within an expression must be addressed in the design of the control of information flow within listening reading.

This section describes the approaches designers have taken to avoid this problem in visual interface design, which may inform the process in the auditory domain.

Borgman (1986) shows that users are more able to navigate through a data-base if they understand its structure. Structure is the physical arrangement or conceptual framework of the system. The structure will affect how people navigate the information space. In algebra, the structure is the arrangement of syntactic components of an expression. Borgman suggests that knowledge of the system’s structure will aid in generating methods of interaction with the system, debugging errors and keeping track of one’s place in the system. So it will be more useful to convey the structure of the expression via browsing, than simply indicate what moves have been made through the expression.

To facilitate the understanding of the structure of the system to be browsed, a user is presented with a conceptual model of the system. A conceptual model is usually presented to the user in the form of a map (Beard and Walker 1990) or a diagram of the structure (Borgman 1986). This could be thought of as an overview or glance. Beard and Walker suggest that provision of a map reduces the load on one browsing component. This is the cognitive, 'where am I and where am I going' browsing function rather than the motor tasks involved.

These findings are backed up by Kerr (1989) who found that a good conceptual model was more useful in maintaining a sense of place in a structure than the navigational aids provided by a system.

> 'A study of strategies, textual, graphic and colour, for cuing users to their location in a data-base showed that the presence or absence of physical cues was less important to successful searching than the user's ability to represent internally the structure of the information' (Kerr 1990, p511).

It is the structure of an expression that Mathtalk needs to convey to the listening reader. Understanding the structure is part of the reading process. To use the browsing moves provided by Mathtalk effectively, the listening reader will need to have some idea of the overall structure of the expression. Having this preview or glance will also aid in navigation and orientation within the expression.

In a speech environment the extra cues provided for navigation could be a hindrance. Rosson (1985) describes access to a data-base via a telephone link. A synthetic voice gave feed-back. After complaints about the voice quality, most users cited lack of orientation as a major problem. In an auditory interface it was difficult to give the user a conceptual model of the data-base's structure. A similar problem is likely to develop in presenting algebra notation. Rosson suggests some answers to the problem, including explicit confirmation of moves (without increasing speech overheads). Rosson suggests this could be done by increasing the amount of implicit navigational information, by varying voice or message type according to place.

Visible cues are less demanding and more avoidable than spoken cues. The suffix effects described above show how extra speech cues could disrupt the primary information in reading. The limitations of short-term memory combined with the lack of control over attendance to the cues suggests that the provision of the structure should be of highest priority. Other navigational cues should be designed to avoid any disruption of the reading.

2.7.3 Program Browsing

Reading computer program source code has some interesting parallels to a blind person reading algebra notation. Both are terse, informationally dense material. Both can have complex, but rigidly defined structures. A computer program will be much larger, covering many pages. The fact not all the structure is ever present means a sighted person reading a computer program can encounter the same problem in apprehending overall structure as a blind person reading an algebra expression.

When viewing source code, programmers are not simply reading the text, they are also trying to comprehend its structure (Shneiderman, Shafer, Simon, and Weldon 1986). To do this more information is needed than is available on the display. Shneiderman et al. (1986) review strategies for increasing the amount of information available on the screen. The design strategy is to provide several different views of the program source code. Similar needs are apparent when reading algebra notation.

Factors such as data availability, its complexity and size of the display have to be accounted for in the design of the display (Shneiderman et al. 1986). Such designs include splitting large screens to fit two sections of code onto the same display, providing more ‘context in the large’ and reducing the need for tedious navigation. This increases time allocated to comprehension. Embedded display allows information about any item to be shown without movement. This removes one opportunity for becoming lost in the information.

Synchronised displays allow a reader to compare two documents at once, again reducing cognitive load on navigation and remembering hidden information (Shneiderman et al. 1986). Perhaps the most successful strategy is the hierarchical browser. This is ‘a representation of the high-level information structure that may be used to access the source code of a program or other text. It is easier for a programmer to find the design scheme in the structured elements than in the bare source code’ (Shneiderman et al. 1986, p10). The map provides context in the large, and aids navigation. However, users still became lost at lower levels of the structure.

Unfortunately these techniques are very visual, though auditory equivalents may be found. The problems encountered are the same and the fundamental answers are also the same. The amount of information available to a user has to be increased. However the information should be presented in a manner that reflects the information structure and in a way that does not overload the user. The user must also be able to control this flow of information.

2.8 Conclusions

From the review of reading, some important conclusions and concepts emerge. The first is that it is the mechanical aspects, not the cognitive processes of comprehension, which must be the target of the design of the Mathtalk program. It is the control over the selection of information afforded by the fast and accurate control of information flow that is not available to a listening reader. This makes listening reading passive, tedious and mentally taxing.

The fine control characteristic of visual reading is only possible because of the external memory. This permanent record of information frees the resources of the internal memory for the comprehension process. The form of the print on the external memory makes the selection process easier, from higher-level document structure, down to the parsing of an algebraic expression. An external memory can also be useful in prompting the reader to use certain procedures stored in long-term memory.

The themes of control and external memory are the guiding principles of the design of the Mathtalk program. By addressing the poor external memory of a spoken presentation and the resultant lack of fast and accurate control, passive listening can be transformed to active reading.

The solution to these two fundamental problems lies in the design of the user interface to a machine representation of algebra expressions. The design principles set forth in this thesis aim to enable usable, active listening reading. This approach is the counterpart to that of Raman in his work on the ASTeR program. Raman's work concentrated on the provision of a rich internal representation of algebra notation on which a user interface must be based. The AFL also provides the tools for implementing the design principles for usable listening reading. In the long-run the two approaches should be combined to give usable access to the widest possible range of algebra notation.

The addition of prosody to the speech output and the use of browsing were proposed as solutions to these problems. The prosodic component of speech can be seen to indicate the structure of an utterance and aid in its retention in memory. Thus addition of prosody could make the presentation in speech have some of the qualities of an external memory. Not only should the presentation indicate the structure, but it should do so in a way that aids the parsing process. The addition of prosody could act like the formatting seen in print.

Both the literature on reading and browsing indicate the importance of an overview. During browsing having a model of that which is being explored is important for orientation and navigation. In reading it is important in planning or creating expectancies for the algebra reading task ahead. The provision of a glance is one of the objectives of the design of the Mathtalk program.

Browsing can be seen to be the counterpart of the selection process in visual reading. It is not reading in itself, but aids the comprehension process by only selecting for output what is needed to move the comprehension process forward.

For adequate control, the reader must be able to visit all parts of the structure of an expression with speed and accuracy. It is the reader who does the reading, not the machine. This means that the control mechanism must not be prescriptive about how an expression should be read. The system must be flexible enough to allow a variety of strategies, strategems, tactics and moves to be used. By paying attention to the components of browsing described by Kwasnik this can be achieved.

The following chapters explore the designs used to achieve these objectives; the evaluation of each of the components of the Mathtalk program and the final evaluation of the complete Mathtalk program.

Chapter 3

Speaking Algebra Notation

3.1 Introduction

In this chapter the answers to two basic questions in the design of the user interface for reading algebra notation are investigated: What information to present and how to present that information in speech? The aim of the design process was to improve the utility of a spoken presentation as an external memory. The aim was to convey only that information contained within the printed expression: That is the structure and content of the algebraic expression. In addition the mental workload associated with the listening process described in Section 2.3 should be reduced.

A description of the scope of algebra notation presented and the target user group are given, together with a rationale for these choices. Then the form and function of standard algebra notation are discussed with a view to answer the question 'what information to present?'.

In the rest of the chapter the question of 'how to speak the chosen information?' will be investigated. Two methods of presenting algebra notation are investigated. First, a subset of rules from Chang's (1983) method of presenting algebra in speech are developed that are consistent with the type of information and the scope of algebra presented. The advantages and possible disadvantages of this approach are discussed. The second approach to presenting algebra notation is to use prosodic cues to indicate algebraic structure. A set of rules to accomplish this end were developed. An experiment was performed to investigate the effect of adding prosody to the synthetic speech to compare a prosodic presentation to the more traditional method of inserting lexical cues.

The chapter concludes with a summary of the approach taken to presenting algebra and a discussion of the effectiveness of prosody to indicate grouping within an expression, enhance the retention of lexical content and reduce the mental workload involved in the task.

3.2 What Notation to Speak and to Whom

In this research only the core of standard algebra notation was tackled. This is letters, numbers, a basic set of operators (see Table 3.1), radicals of arbitrary degree, superscripts, parenthesised sub-expressions and fractions. The user interface is capable of dealing with expressions containing any of these objects, to an arbitrary complexity.

This core of the notation forms the basis for most scientific and mathematical notations. The principles used in the design of this user interface should be extendable, not only to a wider set of algebra notation, but also to other notations based on standard algebra notation and structured, complex information.

The design of the Mathtalk program aims to emulate the reading process based in school mathematical tasks. This is the typically paper based exercise , where an expression is read in the context of a target solution. For example, an expression has to be read in the context of a goal such as ‘solve the equation $3x+4=7$ for x’. It is hoped that the design principles described here will form the ground work for further research that will enable blind school-children and students to read, write and manipulate algebra notation. Such tasks depend on the reader apprehending and then transforming the structure of an expression according to a set of rules. The Mathtalk program aims to enable such users to perform the reading part of these tasks and apply their own knowledge of the rules. As these tasks are typically carried out with pencil and paper it is this paradigm that was used as the background to the design process.

The Mathtalk program will only be used to present correctly formed and complete expressions. When the user interaction develops into reading, writing and manipulation the user will have to deal with incorrectly and partially formed expressions in the reading process. How to present poorly formed expressions will present interesting design issues that will build upon those derived from this work.

It would be appropriate to be able to read, write and manipulate algebra notation with similar functionality or utility that a sighted person does with printed algebra. There is a need to be able to read algebra notation in an equivalent manner to that in which paper is used. The development of teaching aids or manipulation tools such as Mathematica could build upon the design principles presented in this thesis, but will be a future research project. However it should be stated that it should be possible for the algebra to be used in an equivalent, but not the same, way. Computers offer the potential to develop different and powerful functionality than that which is available with paper. The design of this user interface will take advantage of this potential, without contravening the design principles outlined for the presentation of algebra notation.

3.3 What Information to Speak

The first question that must be asked in the development of a user interface to read algebra notation is what information must be presented. Algebra notation is used for the communication and manipulation of mathematical concepts. The user interface therefore must have the same function. 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. The Mathtalk program was designed to replace the role usually performed by the paper or external memory. If the missing functionality of external memory 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.

What information is present on the paper and what does the reader bring to the reading interaction? The expression:

$$E = mc^2$$

can be used to explore the different levels of information associated with an expression written in standard algebra notation and its reader.

1. Types of symbols: letters, a relational operator and a numeral. The presentation indicates what symbols are in the expression and the reader uses his or her knowledge to identify their type and meaning, for example that $=$ is a relational operator expressing equality.

2. the association between the symbols: A quantity E is equal to the quantity m multiplied by the square of the quantity c. This correct parsing is achieved by the application of the reader's knowledge of algebra syntax to the presentation on the page and facilitated by the style of presentation on the page. It should be noted that the reader can also parse incorrectly, for example that E is equal to the product of m and c, which is then squared.

3. A knowledge of what the symbols mean in a wider context, that is, that 'energy is equal to mass times the square of the speed of light'. Another interpretation could be that $E = mc^2$ is a quadratic equation with a variable c. This interpretation is based on the application of the reader's wider knowledge of mathematics or physics. This knowledge is not inherent in the presentation itself.

4. A deep understanding of the physics of energy mass equivalence or a misunderstanding that the equation is something to do with Einstein and relativity. This is a deeper understanding

> or misunderstanding of what is meant by the presentation due to the level of the reader's knowledge. This is knowledge that the user brings to the page, it is not necessarily present on the page.

Printed standard algebra notation presents the grouping or association of symbols within an expression. Delimiting symbols, such as parentheses group certain objects together that must be dealt with in a certain way. Parentheses and fraction lines delimit the scope of operations such as multiplication and division. The way an object or group of objects are placed as superscripts delimits the scope of the operation denoted by that placement. The manner in which the notation is written unambiguously groups the objects in the expression so that the reader can apply his or her knowledge of the syntax of algebra to parse the expression. As Kirshner (1989) described, the spatial rules encode the order of precedence in the style of printed algebra. This implicit encoding of precedence and explicit grouping facilitate correct parsing of an expression, given that the reader has knowledge of that syntax.

This presentation of the grouping of objects within an expression is the main function of printed algebra notation. This functionality must be preserved within a speech based presentation. This principle can be further refined by examining what information the reader brings to the notation.

The symbols $2x^2$ may either be correctly parsed as $2(x^2)$ or incorrectly as $(2x)^2$. The print presentation facilitates correct parsing by instantiating the order of precedence with different spatial cues. However, it is the reader who knows that horizontal juxtaposition indicates multiplication and that diagonal juxtaposition indicates exponentiation. The presentation displays the grouping of the symbols unambiguously, but not the meaning of that positioning. This principle should apply to the speech presentation: enabling parsing, but not explicitly indicating the semantics of the grouping.

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. It 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. It is proper in the context of school education that a reader should be able to misinterpret as well as interpret correctly. It was the aim of this research that a user interface be designed that enables algebra notation to be read in an equivalent manner to printed algebra notation.

From this analysis emerges a basic design principle: That 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 Mathtalk would not 'read to' a blind person, but that the blind person would do the reading.

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: ‘Two x squared’, ‘two times x squared’, ‘two x to the two’, ‘two x to the power two’, ‘two x superscript two’. These spoken forms move through a spectrum of interpretations from mathematical interpretation ‘squared’, to syntactic interpretation ‘to the two’ and finally to a simple presentation of grouping ‘superscript two’. In speech $2x$ is rarely presented as ‘two times x’, so does not present a problem, however, when parentheses are used the word ‘times’ is often inserted as a cue to indicate the onset of the parenthesised group; for example $3(x + 4)$ can be spoken as ‘three times x plus four’. This does give the listener some syntactic interpretation, but may be useful, first as a cues to the onset of groups and also that it is usually, making the utterance flow and be listenable. As will be seen later a global principle of minimal syntactic interpretation is applied, but sometimes compromise is needed so that the best form of presentation is used.

Another example is the fraction line; this is usually spoken as ‘over’, which is a description of the visual presentation that has come to mean the operation of division itself. In this case using the word ‘over’ is acceptable within the principle of non-interpretation. Using the tag ‘fraction’ in the same context does give some interpretation. As the mode of display is developed in this chapter some compromises will be made and explanations will be given as they occur.

Some more interesting decisions have to be made for notation outside the scope of this thesis. Three examples will be discussed here: $f(x)$, $A \cup B$ and $\frac{dy}{dx}$. A usual notation for a function f with parameter x is to write $f(x)$ and this would be spoken as ‘f of x’. The problem is that two identical styles of visual presentation: $3(x + 4)$ and $f(x)$ have two syntactic interpretations, and consequently are usually spoken in different ways. Several questions can be asked to aid the decision for spoken presentation: Is the notation so familiar that some syntactic interpretation is acceptable? Does a misinterpretation in presentation matter? Is there a reasonable, flowing, listenable non-interpretative spoken presentation that is acceptable?

A similar problem arises with the presentation $\frac{dy}{dx}$ that may be presented as a fraction, when it is not a fraction. A minimal utterance of ‘d y over d x’ is syntactically non-interpretive, but ‘ over’ has come to mean ‘divided by’. A student familiar with calculus may not be troubled by such a presentation. A more interpretive description such as ‘the differential, d y, dx’ is not acceptable for the non-interpretation approach. Such examples highlight a problem with spoken presentations: Print on paper has an almost infinite variety of symbols and arrangements available for displaying meaning; in contrast, natural language may lack adequate words to give the desired presentation. For this reason compromises to non-interpretation will have to be made.

A final example at this point is $A \cup B$, that in set theory represents ‘A union B’. This spoken presentation, which interprets the symbol $\cup$, may be acceptable when the symbol’s meaning in the context of set theory is familiar, just as it is reasonable to judge that $+$ is readily understood by all students at secondary level or above, so saying ‘plus’ would not compromise a non-interpretation approach, whereas ‘vertical line with horizontal line crossing’ would not be an acceptable, non-interpretive presentation. It could also be ambiguous. It may not be acceptable when learning, where a question such as ‘what does the symbol $\cup$ mean?’ may be compromises when the presentation says ‘union’. The only problem is that, like ‘superscript’, the association between the name or tag must be learned, but this is also true of a visual display.

For a spoken display of algebra notation, in the context described, the approach of simply displaying grouping with minimal syntactic interpretation should give a consistent display. Where these principles are contravened design decisions can be made by asking the questions outlined above.

3.4 Presenting Algebra Notation with Lexical Cues

In Chapter 2 a description of Chang’s rules for adding lexical cues to spoken algebra to disambiguate grouping was given. In this section a subset of rules will be extracted and developed for use with the core of algebra notation displayed with the Mathtalk program. The method described has remained consistent throughout this work, however, some of the words have changed as the principle of minimal or weak interpretation has developed.

3.4.1 Names of Objects and Whole Expressions

Letters, both Roman and Greek, present no problems in speech, except case. Chang suggests that either the symbol name is preceded by the case of the letter or lower case is accepted as default and the word ‘capital’ is used as a prefix for upper case. A principle can be developed even from this choice. To reduce the number of lexical cues to a minimum, the most common state is used as a default. Reducing the number of cues should reduce the amount of material both to be spoken, processed and remembered by the listener, making the task easier.

Strings of digits are to be spoken as numbers rather than numerals. This makes no difference for the numbers 0–9, but speaking 13 as ‘thirteen’ rather than ‘one three’ should make the listening task easier by not making the listener convert ‘one two three’ into ‘one hundred and twenty three’, by storing all the digits so that the place value of the ‘one’ can be known. The decimal point will be spoken as ‘point’ and numerals in decimal places will be spoken as single digits.

Operator	**Utterance**	**Relation**	**Utterance**
$+$	plus	$=$	equals
-	minus	$\neq$	not equals
$\times$	times	$<$	less than
$\div$	divided by	$\equiv$	equivalent
$\pm$	plus or minus	$\approx$	approximately equals
$\mp$	minus or plus	$\sim$	similar to
$*$	star	$>$	greater than
		$\leq$	less than or equals
		$\geq$	greater than or equals

Table 3.1: Set of operators used in Mathtalk and their spoken form.

Some decisions have to be made with the printed operators. The symbol '+' is almost exclusively spoken as 'plus'. Like the fraction-line the symbol $+$ has a name synonymous with its operation. The word 'plus' offers some interpretation, but seems a senseless imposition to make the listener interpret 'horizontal cross' as 'plus' when no other name is used and the meaning is overlearnt by an early age in education. A more interesting choice arises with the operator $-$. Should this be spoken as 'minus' or 'dash'. The second is more descriptive and less interpretive, but the former is preferred, to be consistent with 'plus'. The names 'less' and 'take away' will not be used. Similarly $\pm$ is spoken as 'plus or minus' and distinguished from $\mp$ by reversing the order of speech to 'minus or plus'. The speaking of operators offers some interpretation and some compromise has been made between interpretation and what may be called listening *legibility*. A summary of symbols and their spoken form are shown in Table 3.1.

Unary operators pose another decision. Should $-b$ be spoken as 'minus b' or 'negative b'? The latter distinguishes the unary operator from the binary, but does it offer too much interpretation by prompting the listener to treat the 'b' as a negative value? To be true to the cause of minimal interpretation the Mathtalk program will speak the unary operators in the same manner as the equivalent binary operator. This has the additional virtue of not misleading a visually disabled reader into thinking there are two different symbols for the unary and binary $-$ symbol.

3.4.2 Sub-expressions and Radicals

The phrases suggested by Chang: 'begin quantity' and 'end quantity' are used to delimit the start and end of parenthesised sub-expressions. So the expression $3(x+4) = 7$ is spoken as '3 times the quantity x plus four end quantity equals seven'. An alternative to the word 'quantity' is 'group'. The word 'quantity' might imply that the contents should be regarded as one mathematical entity, where the word 'group' might imply less, that the symbols are simply grouped together and the reader then has to decide that the group be treated as a 'quantity'.

Chang uses the following method to reduce the number of cues: If the end of the group is implicit, because of another feature of the expression, then the final lexical cue may be omitted. For example in $3(x+4)=7$ the $=$ can only occur at the base level, so the sub-expression must have finished. So that expression is spoken as ‘three times the quantity x plus four equals seven’. The end of an expression also ends all groups. So $y=(x+2)(x-2)$ may be spoken as ‘y equals the quantity x plus two end quantity times the quantity x minus two’.

The use of the radical symbol $\sqrt{\ldots}$ can also group items together. The choices for speaking the symbol are ‘radical’, ‘root’ and ‘square root’. The first two are synonymous, but the word radical is not in common usage. The lexical cue ‘square root’ is probably the most common form, but does offer some interpretation to the listener. The representation has developed from the more interpretive ‘square root’ to a more presentational form. Given that a description of the visual characteristics of the symbol are as untenable as for other symbols, the word ‘root’ will be used to denote the radical symbol.

No order in the superscript position of the radical denotes the second order root. Visually disabled readers will learn this convention as readily as their sighted counterparts. When a higher order is present, for example $\sqrt[n]{\ldots}$, then the order n is spoken in the form ‘root n of ....

The scope of the radical symbol can vary from one symbol to many and is denoted by the length of the radical symbol. Invoking the principle of reducing the amount of speech, a default scope of one item is used for the radical symbol. The scope is extended using the same mechanism for parenthesised sub-expressions; thus $\sqrt{b^2-4ac}$ is spoken as ‘root of the quantity b super two minus four a c end quantity’ and $\sqrt{2}$ is spoken as ‘root of two’.

3.4.3 Fractions

The mechanism chosen for describing fractions takes two forms for the Mathtalk program. Fractions are preceded with the lexical cue ‘the fraction’, followed by the phrase ‘numerator’. Then the contents of the numerator are spoken. The lexical cue ‘denominator’ closes the numerator and commences the denominator. After the contents of the denominator have been spoken the phrase ‘end fraction’ can be inserted.

The same rule for closing the fraction operates as in sub-expressions. The use of the word ‘numerator’ might be redundant, but was retained to match the use of the word ‘denominator’.

The Expression 3.1 will be spoken as:

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a} \tag{3.1}$$

> ‘x equals the fraction numerator minus b plus or minus the root of the quantity b super two minus four a c denominator two a.’

Notice that the lexical cue ‘denominator’ terminates both the numerator and the root, negating the need to explicitly end the scope of the radical symbol.

Where both the numerator and denominator are only one term the number of lexical cues may be reduced. These fractions will be referred to as *simple* fractions. Those described above, where either numerator or denominator contain more than one term, will be described as *complex* fractions. For simple fractions the scope of the division operator or fraction line is assumed to be only the adjacent terms and the start and end of the fraction need not be made explicit.

Simple fractions are a situation where the principle of reducing cues can be used. For example the fractions $\frac{x}{y}$ and $\frac{2a}{4b}$ are spoken as ‘x over y’ and ‘two a over four b’ respectively. Some ambiguity may arise when a fraction forms a superscript or a superscript appears on the last item in the numerator. For example $x^{\frac{1}{2}}$ would be spoken as ‘x super 1 over 2’. This may be resolved as $\frac{x^1}{2}$ or $x^{\frac{1}{2}}$. A small extension to the rule for simple fractions states that the phrase ‘the fraction’ should precede a simple fraction that forms a superscript. The absence of the other keywords denotes the fraction as simple. Not using the prefix ‘the fraction’ means that the listener will have to backtrack to include the previous term in an adjusted internal representation. This extra workload has to be weighed against the decrease in the number of lexical cues that will speed up the presentation and reduce the amount of processing to be done by the listener.

At this point a compromise has to be made against the principle of no syntactic interpretation. The words ‘fraction’, ‘numerator’ and ‘denominator’ have to be used to disambiguate the structure where one set of objects are placed above another, separated by a horizontal line. No other reasonable lexical cue exists that would describe the positional and grouping information in a terse and informative manner. The language does not contain a suitable word to describe the presentation of a fraction, except the word ‘fraction’. For the range of algebra notation to be spoken by the Mathtalk program the use of these cues will not cause confusion, as no other meanings use the same construction (calculus is not included).

3.4.4 Superscripts

In print a superscript character usually means exponentiation and this is reflected in the number of ways of presenting such a form in speech. See the example $2x^2$ given in Section 3.3. In some situations a superscript may not denote exponentiation (in this context accents are not counted among superscripts). For example in second order derivatives ($\frac{d^2y}{dx^2}$). Many of the methods for speaking superscripts imply exponentiation. For the target user group, such interpretation would probably not offer undue assistance. Nevertheless, for consistency with the principle of non-interpretation, such syntactic interpretations should be avoided where possible.

The method for presenting superscripts within Mathtalk has developed from a more interpretive method of saying ‘to the’ to initiate a superscript and ‘end power’ to terminate. Superscript two and three were spoken as the exceptions ‘squared’ and ‘cubed’. Over time this has been reduced to the less interpretive form given below.

The presentation used is the lexical cue ‘superscript’, which is shortened to ‘super’ to increase speed and reduce verbal clutter. The phrase ‘end super’ is used to terminate superscript groups of terms. This cue only describes the grouping of objects in the expression and does not give any syntactic interpretation. It will be up to the reader to learn the association, just as the sighted reader learns the spatial association.

Chang offers the approach of implicitly ending superscripts that only contain one term and only using the lexical cue to end superscripts with more than one term. This is the system adopted in Mathtalk and is consistent with methods used in the other constructs discussed above. So, $x^n + 1$ is spoken as ‘x super n plus one’ and x^{n+1} is spoken as ‘x super n plus one end super’. Objects such as fractions that are present as superscripts may be regarded as single items and therefore do not need the terminating lexical cue. For example $x^{\frac{1}{2}}$ would be spoken as ‘x super the fraction one over two’ .

When a parenthesised group or fraction has a superscript, the word ‘all’ was inserted before the usual lexical cue. This was used to emphasise that the superscript governed the whole of the group to which it was attached.

The general rule is that all superscripts are initiated with the lexical cue ‘super’, implicitly terminated if only one term or object is contained in the superscript and terminated with the lexical cue ‘end super’ if more than one term is present. A listener only knows that a superscript is simple when there is no ending cue to indicate that it was complex. Conversely, as simple superscripts are most common in school algebra, an assumption of simplicity will stand until a cue indicates otherwise. This means there is an ambiguity in the presentation, that may be resolved, but may present extra work for the reader. Again, the reduction of lexical cues took a higher priority than an immediately explicit indication of complexity like that seen in speaking radicals.

3.4.5 General Rules

The following list of rules summarises those rules to be implemented in the Mathtalk program and contrasts some of the differences with the wider setoffered by Chang. Following the list, some of the general principles used in developing this subset are expanded.

- For Latin letters, Mathtalk only prefixes uppercase with a tag. For example ‘capital a’, leaving lowercase ’a’ unadorned. Chang offers the choice of prefixing both upper- and lowercase letters.

- For sub-expressions, Mathtalk uses only the tag ‘quantity’. For example, $3(x+4)=7$ is spoken as ‘three times the quantity x plus four end quantity equals seven’. Chang offers the choice of replacing ‘quantity’ with ‘parentheses’, ‘quantity’, ‘the sum’ or ‘the difference’.

- Mathtalk speaks simple fractions (those with a single term in both numerator and denominator) with only the word ‘over’ between the two terms. For example $\frac{1}{2a}$ is spoken as ‘one over two a’. Chang optionally prefixes such a construct with the phrase ‘the fraction’. Given the rule below for complex fractions, this can lead to ambiguities.

- Complex fractions (those with more than one term in either numerator or denominator) are bounded with lexical cues. For example, $\frac{x+1}{x-1}$ is spoken as ‘the fraction, numerator x plus one, denominator, x minus one, end fraction’. Chang optionaly omitts the cues ‘numerator’ or ‘denominator’, replacing the latter with ‘over’.

- In Mathtalk a relational operator or the end of the expression always removes the need for a closing tag for sub-expressions, root and fractions. This is optional in Chang’s rules.

- Roots are enclosed with the lexical tags ‘the root’ and ‘end root’. For example, $\sqrt{b-4ac}$ is spoken as ‘the square root of b minus four a c’. Roots of higher order are spoken as ‘root *n* root of *m*’, where *n* is the order and *m* is the radicand.

- Simple roots are spoken without an end tag. For example, $\pm\sqrt{2}$, is spoken as ‘plus or minus the square root of two’. Later this tag was reduced to ‘the root’ rather than ‘the square root’. Similarly, a ‘cube root’ became ‘root three of …’.

- Initially Mathtalk used the cue ‘to the’ to indicate exponents. Later this was replaced by ‘super’ (shortened from ‘superscript’) to comply with minimal interpretation. The expression x^{n+1} is spoken as ‘x super n plus one, end super’.Chang offers a selection of lexical cues: ‘to the’, ‘exponent’, ‘to the power’ and ‘superscript’. In Mathtalk, complex exponents are terminated with the word ‘end’ followed by a repetition of the salient word from the opening tag. Chang leaves the choice open of how to close superscripts.

- The word ‘all’ can be used with the opening superscript cue, when the superscript governs a complex object. For example, $(a+b)^c$ is spoken as ‘the quantity a plus b, end quantity, all super c’. Chang also offers this cue to emphasise the scope of fraction lines. After the evaluation, this cue was omitted.

The notion of simple and complex notation can be introduced to give a general rule for the use of lexical cues to disambiguate spoken algebra. The complexity is the degree of syntactic structure within an expression. The presence of a syntactically complex subunit or object will make an expression complex. A complex object has more than one term grouped by explicit parsing marks or spatial location. A term is a group of one or more operands separated by a least precedence operator. At points where an expression becomes complex, then lexical cues need to be used to delimit the scope of those groups. For example the complex fraction and the parenthesised sub-expression.

Fractions, superscripts and roots form an exception where their presence does not necessarily denote complexity, but scope needs to be presented. Roots and superscripts are only explicitly terminated when more than one term is involved, that is, they are complex. For all the constructs used in the Mathtalk program, complexity of structure may be used to guide the insertion of lexical cues.

Determining the complexity of an expression is proposed by Ernest (1987) as one of the first stages in the process of reading an expression. The presence and scope of structures in an expression will form a large part of this complexity. Presenting the structural complexity in a clear, unambiguous manner is a basic requirement for the Mathtalk program.

Another general rule is to reduce the number of lexical cues by speaking structurally simple objects unadorned with extra lexical cues. More reductions can be made by letting the most common form act as the default.

A further reduction in the use of cues was made by allowing structures higher in the level of nesting to terminate those inside, for example, Expression 3.1 presented with lexical cues in the discussion of fractions.

An attempt has been made to provide no syntactic interpretation when using these cues to speak algebra. The least interpretive of Chang’s choices have been chosen, made simple and consistent. In some cases it proves difficult to avoid some interpretation. Many of the symbol names describe their use, so some interpretation is necessary for a reasonable presentation. No syntactic interpretation is the ideal, but compromises are made on several fronts: When there is no alternative name; to provide easy flowing speech; or consistency with some earlier compromise.

This method of presenting algebra notation has the virtue of making the grouping within an expression explicit. These rules principally cover the set of explicit parsing marks described by Kirshner (1989), but take little account of the implicit parsing cues.

When either expressions containing lexical cues or plain expressions are spoken with a speech synthesiser a pauseless stream of speech, with little emphasis or pitch change emerges. This means the listener has to rely entirely on the inserted lexical cues and the operators present to parse the expression. Given that spatial cues help visual parsing, their lack may mean auditory parsing is more mentally taxing.

A large number of extra words can be inserted into an expression to ensure it is spoken with unambiguous grouping. Unfortunately the addition of extra words may increase the burden on the listener's memory resources. The quantity of information, together with the nature of the synthetic speech may make this form of presentation difficult to use.

3.5 The Prosodic Alternative

Section 2.6 described some of the roles of prosody in human speech. Some of these capabilities used in spoken algebra could improve the presentation. The three functions of prosody of interest were its ability to indicate syntactic structure; presenting the information structure and the psychological aspects of improving comprehension and retention.

The studies described in Section 2.6 indicated listeners could use prosodic cues to recover structural information from spoken algebra. If prosodic cues could be added to an arbitrary expression presented by Mathtalk, then the structure of the expression could be displayed. At the same time, the number of lexical cues used could be greatly reduced. This could avoid disruption of retention by the suffix effect. The division of the utterance into units of information, together with rhythm could increase retention of content. Other cues, such as the declination effect could indicate the length of an expression. Such benefits could also decrease the mental workload associated with the listening task.

The problems of adding prosody to synthetically spoken text were discussed in Section 2.6. Without knowledge of the structure and intention of an utterance, it is impossible to give a full prosodic account of that utterance. Some of these issues can be avoided in Mathtalk. In algebra, the structure of the expression is known and captured in the internal representation of the Mathtalk program. The prosodic cues only need to be added to the algebraic utterance to indicate structure. The addition of prosodic cues to indicate the intention of the expression, through knowledge of the semantics of the expression, is beyond the scope of the Mathtalk program. The addition of such cues would also contradict the principle of non-interpretation. What is needed in the Mathtalk program are those prosodic cues that indicate the structure or grouping of an expression. Given that the grouping was captured in the internal representation of the Mathtalk program and a set of rules could be derived to add prosodic cues to these boundaries, then a prosodic presentation could be generated.

Most commercially available speech synthesisers are capable of manipulating the speed, pitch, amplitude, duration and timing of generated speech (Edwards 1991). This means that prosodic cues can be added to an utterance of any algebraic expression if the rules for adding those cues are known. An advantage of using speech synthesisers is that any rules can be implemented consistently. Consistency of use of prosodic cues is not a feature of human speakers (Crystal 1987). In addition prosodic cues may be exaggerated so that any information imparted may be made more obvious.

In the following sections the rules for algebraic prosody already developed by O'Malley et al. and Streeter will be extended. The prosodic cues derived will be drawn up into a set of rules and implemented within the Mathtalk program. Finally, the utility of these cues will be compared to the lexical cue method of presenting algebra described above.

3.5.1 Extending the Rules for Algebraic Prosody

Streeter demonstrated that listeners could reliably recover parentheses from spoken algebra using prosodic cues. However, these rules need to be confirmed and extended for the purposes of the Mathtalk program.

The study of O'Malley et al. used 50 expressions. The investigation was solely interested in defining a set of parsing rules to re-insert parentheses into spoken algebra. The expressions did include some simple fractions, superscripts and functions, but the focus remained on parentheses. O'Malley et al. mention that pitch contour, duration and amplitude were thought to be useful cues for deriving syntactic structure, but present no rules for their use.

Streeter used a set of eight expressions to explore the potential of prosodic cues to disambiguate an algebraic utterance. The expressions were variants of the forms: $a + e + o$, aeo and $a + (eo)$ etc. Fractions or superscripts, in either simple or complex forms, did not appear in the investigation. In Streeter's study, pitch was found to be the strongest cue, followed by duration (the equivalent of pausing in O'Malley et al.'s study), with amplitude acting as a minor cue for recovery of structure.

In both studies the reported expressions used were short and the cues described did not include the global pitch changes reviewed in Section 2.6. Further investigation is needed to see if these and other cues are present in spoken algebra. Another result of using only short expressions was that the effect of length itself, the presence of multiple structures, either nested or in series, was not apparent. It is possible that prosodic cues either do not exist or are not capable of indicating such complex structures.

The rules described by O'Malley et al. and Streeter were for American English. An investigation into the prosodic rules for British English was needed, not only to extend the rules, but also to confirm that the rules were similar or the same to those for American English. Whilst the prosodic cues used by Mathtalk need not be fully accurate, the auditory display would be enhanced if the prosodic cues were at least familiar, therefore more intuitive and learnable.

Method

This investigation into the structural prosody of spoken algebra did not attempt to be exhaustive. The aim was to confirm the pre-existing rules described earlier and to suggest trends or features used in spoken algebra above those already known. If the same prosodic features used in natural language also appear in spoken algebra, then those features may easily be included in rules for algebraic prosody. If, however, prosodic behaviour contradicts that of natural language, then further investigation would be necessary.

For this investigation a set of twenty four expressions were devised. These may be seen in Table 3.2. These expressions represented the core of algebra notation used in the Mathtalk program. Where necessary, constructs were present in both their simple and complex forms. These expressions, as far as possible given the small number of expressions, attempted to use the structures on their own and in combination. The combinations were used to explore the effects of one structure on how another was spoken. The structures were also presented at different positions within the expressions to explore the effect of position on prosodic cue used.

The expressions appear in pairs that contrast simple and complex structures. For example a pair may be $3x + 4 = 7$ and $3(x + 4) = 7$. Contrasting the cues used in each expression indicates how they may be discriminated in speech. Expressions 3.2–3.9 show superscripts in different contexts. Expressions 3.10–3.21 show parenthesised sub-expressions and Expressions 3.22–3.25 contain simple and complex fractions.

Two experienced speakers of mathematics were used in the study. Both were native speakers of British English. One was an ex-school teacher of mathematics and one was a postgraduate student of mathematics, with teaching experience. Two recordings on high quality tape were made for each speaker. Separate recordings were made for each participant to reduce effects of fatigue and memory for individual expressions. Each expression was printed on a separate card. The expressions themselves were shuffled so that pairs did not appear together and different orders were used in each recording.

The speakers were asked to read the expressions as if they were addressing a class of sighted students and were pointing to a written expression on a blackboard. They were asked to present the expression in a neutral manner, that is, not to indicate any of the mathematical intentions of the notation. This was an attempt to bring the speakers' presentation into line with the non-interpretive approach taken in Mathtalk. They were asked to use extra lexical cues only when they were

$$x^n + 1 \tag{3.2}$$

$$x^{n+1} \tag{3.3}$$

$$x^4 n \tag{3.4}$$

$$x^{4n} \tag{3.5}$$

$$x_4 \tag{3.6}$$

$$x^4 \tag{3.7}$$

$$y = ax + bx + c \tag{3.8}$$

$$y = ax^2 + bx + c \tag{3.9}$$

$$ab + c - ef - g \tag{3.10}$$

$$a(b + c - e(f - g)) \tag{3.11}$$

$$a - b + c \tag{3.12}$$

$$a - (b + c) \tag{3.13}$$

$$3x + 4 = 7 \tag{3.14}$$

$$3(x + 4) = 7 \tag{3.15}$$

$$x + y^3 \tag{3.16}$$

$$(x + y)^3 \tag{3.17}$$

$$-(a + b) \tag{3.18}$$

$$-a + b \tag{3.19}$$

$$a + ba + b \tag{3.20}$$

$$(a + b)(a - b) \tag{3.21}$$

$$1 + \frac{x}{y} + 4 \tag{3.22}$$

$$\frac{1 + x}{y + 4} \tag{3.23}$$

$$\frac{a}{b} \tag{3.24}$$

$$ab \tag{3.25}$$

Table 3.2: The expressions used in the investigation into algebraic prosody.

thought necessary. This was an attempt to reduce the number of cues to a minimum and only use those cues in ‘common’ usage that may make an expression flow more easily.

The recordings were analysed, by experts in the linguistics department at the University of York, for pitch, timing and amplitude. Changes in these three parameters were related to the individual syllables in the expressions. Pauses were measured in milliseconds (ms) after a particular syllable. Pitch was measured in Hertz (Hz) at either the beginning, end or mid-point of a syllable. Where pitch changes were approximately linear only the beginning and end points of the linear pitch contour were recorded. Amplitude was measured and categorised as either low or high[1]. This high-level analysis was deemed suitable for the simplistic model of prosody that would be implemented within the Mathtalk program.

Only one recording was analysed in full. By observation, all four recordings were consistent, so reducing the need for detailed analysis of each recording.

The full data set may be seen in Section A.1 of Appendix A. The results below are divided into different classes of prosodic effects and examples illustrating these features can be seen in Appendix A.1.

Global Pitch Changes

Information on pitch was recovered for 23 of the 24 expressions. All but one of these showed a decrease in pitch over the utterance. The one deviation from this trend, x_4 (Expression 3.6), can be treated as a special case. The subscript was spoken with a distinct fall-rise tone, which resulted in the final pitch being higher than the initial pitch. This expression was included only for contrast with the superscript; subscripts were not to be presented by the Mathtalk program.

For all the other 23 expressions, the mean initial pitch is 159 Hz, with a standard deviation of 20 Hz. The mean final pitch is 110 Hz, with a standard deviation of 8 Hz. The higher initial pitch has a much greater spread than the terminal pitch. The low spread for the terminal pitch indicates that the speaker tended to finish an utterance at a constant pitch.

A value of 0.2 for Pearson's correlation coefficient indicated that the length of an expression, in syllables, was not associated with high initial pitch. However, removing the result for Expression 3.10, moved the value for ρ to 0.6, which was highly significant. Expression 3.10 had the first two syllables spoken with a sharply rising tone. These rising tones may be a result of the expression's length (11 syllables), exceeding the speaker's pitch range and causing the rising tone to be used (see Section 2.6). Table A.1(c) shows the data for this expression.

This information in the declination effect provides two potentially useful cues for the listener. If initial frequency is determined by the expression's length, then the listener may be able to use this

[1] These recordings were made and analysed with the help of Professor John Local of the Department of Language and Linguistic Science at The University of York, UK

information to anticipate the expression's length. Secondly, as the speaker's tone approaches the consistent terminal pitch, the listener could also anticipate the end of the expression.

Pitch Changes within the Term

Two other pitch trends make themselves apparent within this overall pitch decrease. If the expression has no base-level operators, the pitch fall is roughly linear from start to finish. For example, see Expressions 3.17 and 3.3. The first syllable of a term may act as the tonic, over which most of the pitch change occurs, as described by Halliday (1970). For ease of implementation within the Mathtalk program the pitch fall was taken as linear, in spite of the implicit inaccuracy compared to 'natural language'.

When operators occur at the base-level, the pitch falls in a series of steps, interrupted by pauses. This is the same as the 'hat effect' described in Section 2.6. Inspection of the pitch changes indicate that pitch remains level or rises on a base-level operator. However, at the final base-level operator in an expression, the pitch fall becomes linear once again. This definite fall at the start of the operator, rather than at the first operand of the term, gives a sharp pitch fall at the end of an expression. This pitch fall accounts for, on average, 34% of the total pitch fall within an expression. The pattern of pitch changes can be seen in Figure 3.1(a).

This pitch behaviour is well documented within natural language and has been called the 'hat effect' ('t Hart and Cohen 1973). Each phrase within an utterance is spoken with a rise and a longer fall in pitch. So structurally simple expressions are divided into units of repeated pitch contours. The lack of a rise, or a rise from a lower initial pitch, at the onset of the last unit is also well known. The sharp pitch fall is thought to indicate the imminent end of the expression. This could be a useful cue for the listener to anticipate the closing of the expression.

Temporal Changes at Base-level Operators

Pauses are consistently seen at base-level operators. Pauses of mean length 250 ms are placed before $+$ or $-$ operators occurring at the base-level. This pause associates the operator with the following term, rather than the preceding term. This association of operators was observed in both studies reported earlier.

A pause of similar length can be placed before or after an $=$ operator. The optional placement of the pause associated with the $=$ operator may depend on the size of each side of the equation. In the example expressions, the pause is placed on the longer side of the expression (Table A.1(i) and A.1(o)). Such a cue could allow a listener to anticipate that a large amount of information is to follow. Such a cue could prove useful in a speech display where the amount of information cannot be surmised quickly. Unfortunately there are not enough examples to be sure of this rule.

These pauses, together with the pitch changes described above, divide an expression into terms. The term becomes the tone unit of spoken algebra, the basic unit of information. The pitch rise at the start of a term indicates the start of new information (Halliday 1970), in this case the term. The information structure is the structure of the expression. This means that spoken algebra is presented to the listener divided into subunits corresponding to the first parsing points, in much the same way that large amounts of white space divide printed algebra into terms (Kirshner 1989). Figures 3.1(a) and 3.1(b) show the patterns of pauses at terms and a parenthesised sub-expression.

It is assumed that these pauses are used to chunk an expression into manageable units. This chunking should make the task of retention, and thus integration, of information more easy. The placement of the pause before or after an ‘equals’ should indicate where the bulk of information lies within an expression.

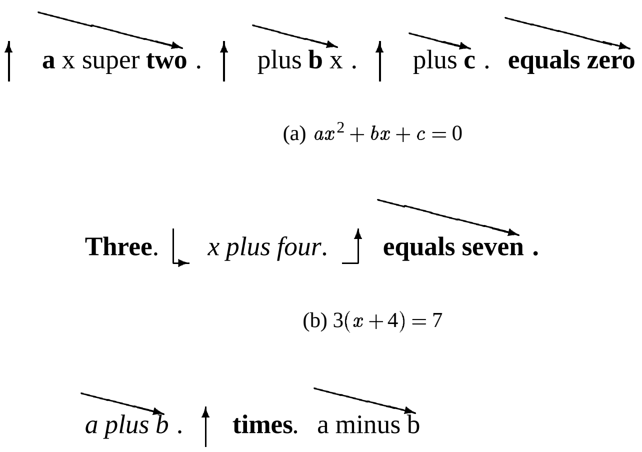
These units of information are phonological units or tone units. Tone units are separated by pauses and each carry a pitch change. When spoken as one tone unit the pitch is a linear fall from start to finish. This is also the case for an expression containing several tone units. The pitch falls from the first operand to the last. However, after the pause, there is a slight pitch rise across the operator to the start of the next term. This is the same as seen in one of the speaker’s in Streeter’s study.

So, in an expression or utterance holding more than one tone unit, pitch falls throughout the expression, but is couched in a series of short rises and longer falls. This pattern is broken for the last term. In this case, the rise associated with the operator is omitted, resulting in a sharp fall from the start of the last operator to the end of the term. Most significantly, all the effects described so far appear in regular English and appear in similar contexts, as described in Section 2.6.

Pitch and Timing Changes at Superscripts

Contrary to expectations, the speaking of superscripts was not associated with a pitch rise. The literature suggests that a high pitch is associated to a high physical position by listeners (Mansur, Blattner, and Joy 1985), and therefore, perhaps also by speakers associating the raised superscript with a pitch rise. This was not the case with superscripts in spoken algebra. The onset of an exponent was indicated lexically with the words ‘to the’, which followed the usual trend of falling pitch within a term. The contents of both simple and complex superscripts also followed the falling pitch of the whole term. For example, compare Expressions 3.2 and 3.3, and Expressions 3.4 and 3.5. Figure 3.1(a) shows the pitch change within a term that includes a simple superscript.

The presence of this lexical cue (‘to the’) may negate the need for an additional prosodic cue. If the speaker were forced to indicate a superscript prosodically a pitch rise may indeed be the correlate.


a x super two . plus b x . plus c . equals zero
Three. x plus four. equals seven .
a plus b . times. a minus b

(a) $ax^2 + bx + c = 0$

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

(c) $(a+b)(a-b)$

Figure 3.1: Representations of the timing, pitch change and amplitude in three spoken algebraic expressions. The arrows show the trend of pitch change; periods indicate pauses; *italic* typeface indicates increased speed and **boldface** indicates increased amplitude.

Including the superscript in the current term, using the pitch contour may be a higher priority than indicating the fact it is a superscript, a feature denoted lexically. The pitch contour also binds the superscript within the group to which it is attached (the term) and indicate its lower precedence.

One pair of expressions (3.4 and 3.5) was used to explore what happened when a superscript was not followed by a base-level operator. The results for $x^4 n$ and x^{4n} are shown in Tables A.1(e) and A.1(f). In the expression $x^4 n$ there was no pause separating the superscript from the following operand. There was a linear pitch fall from the beginning to end of each expression. Instead, unusually, the superscript is not emphasised, but the ‘the’ of ‘to the’ is emphasised. In both expressions the n is emphasised, which would not normally be the case with non-initial operands; so it is, perhaps, being signaled as unusual. A major feature discriminating one expression from the other is that the 4 of x^{4n} was linked to the following n. So the speech was ‘fouren’ rather than ‘four n’. These cues may be too subtle for listeners and to reliably implement, even though the prosody can be exaggerated in a speech synthesiser. A simpler rule of inserting a pause after the superscript was used in the Mathtalk program. Inserting a pause was the corollary of not blending the borders within the superscript $4n$.

Expressions 3.16 and 3.17 contrast the use of a superscript on a single operand or a group of objects. A lexical cue ‘all’ was used to indicate that the superscript was attached to a group not a single operand, as shown in A.1(r). As described below, some objects were grouped together by tempo and pitch, which should indicate the scope of the ‘all’ before the superscript. An additional cue was added to help resolve scope. A complex object was often terminated with a pause (see below). Advantage was taken of this feature to put the superscript attached to such an object 'outside' the expression by placing it after the pause. There were not enough examples of superscripts on grouped objects to be sure that the observed rules were the only ones in use. However, the insertion of a pause was thought to be a suitable cue to close the grouping of the previous objects.

Changes for Sub-expressions and Fractions

Both pitch changes and timing are used to indicate the onset and extent of a sub-expression. Within a sub-expression the rules governing pitch contour and timing are somewhat different from those at the base-level.

Sub-expressions should be divided into two classes. Those that follow an implicit multiplication operator, and those that follow printed base-level operators. Within these types, the position of the sub-expression within the whole expression is important. The two positions to consider are terminal and mid-expression.

In print algebra notation the multiplication operator before a sub-expression is usually implicit, indicated by horizontal juxtaposition. However, the speaker used in this analysis usually inserted the lexical cue 'times' at this point (see Expression A.1(d)). This lexical cue was preceded by a pause of the length described above. Following the lexical cue there was a pitch fall. This fall was greatest in Expression A.1(p), where 'times' was not spoken. Here, the pitch fall was 81 Hz. To emphasise the level change, in the synthetic speech presentation, the lexical cue and a pitch fall will be used. Both the pause and the pitch fall signal the onset of a parenthesised group. The pause was observed in O'Malley et al.'s study, but the pitch fall was not described. In their study, a pause could both precede and follow the lexical cue 'times', in the same fashion as two single pauses were observed either side of an operator before a sub-expression.

If the sub-expression is at the beginning of the utterance, the pitch fall occurs over the first syllable (see Expression A.1(v) and Figure 3.1(c)). That is, initial sub-expressions have a falling tone with the tonic on the first syllable. By inspection, the pitch within a mid-expression parenthesised group was flat, compared to changes at the base-level (Expression A.1(p) and Figure 3.1(b)). However, sub-expressions occurring at the end of an expression show a linear fall, equivalent to that for the initial sub-expressions, (see Expression 3.21) This can be generalised to a rule that states: pitch within a sub-expression is flat, except when a term within the group is at the start or end of an expression. Sub-expressions behave much like whole utterances in which no base-level operators occur.

In addition to the flat pitch, pauses do not occur within the sub-expression, except in very long or nested sub-expressions(see Expression A.1(d)). This was in contrast to the observations of O'Malley et al., where pauses were seen at all printed operators, but sub-expressions were bounded by longer pauses. The lack of pauses in complex groups and the generally shorter pauses may be a result of faster speech in this study. O'Malley et al.'s participants were asked to speak slowly, but no such instruction was given to the speakers in this study.

In Expression 3.11 there are two levels of sub-expression. Pauses occur within the first level, but not the second. This may be a physiological requirement for the speaker to breathe. The whole sub-expression may be too long to omit all the pauses or there is a rule that means only the deepest level is spoken without a pause. Unfortunately pitch information was not recovered for this expression.

These features show sub-expressions grouped together strongly by both pitch and tempo. The pauseless speech and the lack of raised amplitude on syllables (see below) alter the rhythm of spoken sub-expressions, making them appear as a single unit. This strong binding of sub-expressions into a single unit may make them easier to recognise as a distinct unit for the listener.

Only two expressions had $+$ or $-$ operators followed by an opening parenthesis. In both cases the operator was preceded and followed by a pause, in the same manner as 'times'. In Expression A.1(n) the following pause was 216 ms and in Expression 3.18 it was 252 ms. These pauses do not differ significantly from those preceding the same operators at base-level. This rule, that a pause follows an operator, to indicate a sub-expression, was also observed by O'Malley et al.. Adding the pauses before and after the operators adjacent to sub-expressions gives a 'double pause' that O'Malley et al. suggested indicated nesting of a group of symbols within the whole sub-expression.

Complex fractions are treated as two sub-expressions separated by a spoken operator 'over'. The fraction $\frac{1+x}{y+4}$, could easily be represented as $(1+x)/(y+4)$ The fraction would also be spoken as if it were the expression $(1+x)(y+4)$, except that the word 'over' indicates a fraction, instead of the word 'times' indicating horizontal juxtaposition. The pattern of prosodic cues is the same as seen in Figure 3.1(c).

Compare Expressions A.1(v) and A.1 for the multiplication or division of two parenthesised groups. A long pause (333 ms) is placed before the 'over', which has the first syllable spoken with raised amplitude. The 'over' is also followed by a pause. The 'over' is spoken at a higher pitch than either sub-expression. This is similar to the 'times' example. This raising of pitch makes the division between numerator and denominator more apparent and also implicitly refers to the higher-level on which the division operator lies.

The lexical clue ‘fraction’ was not used. That the utterance is a fraction is only indicated when the major operator is reached. This means that the listener has to ‘back track’ to find where the fraction construct started, rather than knowing from the outset. If the grouping of the numerator is very strong this may not be too much of a problem. A balance has to be struck between using too many cues and overloading the listener and using too few and making the structure difficult to apprehend.

Simple fractions were spoken with a linear pitch fall from start to finish (A.1(w) and A.1(k)). The fraction-line or slash was spoken as ‘over’, and no pauses were used within the fraction. The operator ‘over’ was not emphasised by amplitude or pitch as was the case in complex fractions. Simple fractions seem to be spoken more like ordinary terms (for example, ab^2), than complex fractions.

Amplitude Patterns

The general rule throughout the expressions is that the first operand within a term is stressed by increase in pitch and amplitude. Operators are only stressed in certain cases. The $=$ operator was only spoken with raised amplitude once. In Expression A.1(p) the $=$ was emphasised after the close of a sub-expression, perhaps to emphasise the return to a higher level. A decision was made to emphasise all equality operators. As the relational operator forms the root of the parse tree its presence is important cue to the listener. For this reason it was stressed in Mathtalk’s prosody to make it more prominent.

In all expressions a definite pattern of stress was seen. The first operand of each term was stressed giving a simple rhythm to an expression. In Halliday’s description, each algebraic term would form a foot, the basic unit of rhythm, with the first syllable salient. In speech, the inter-stress interval establishes the rhythm. This fits in with the notion of each term forming a tone unit, with the tonic on the first syllable. This also correlates with the notion of the term as being the basic unit of information. Establishing a rhythm in a spoken expression, however crudely, may well aid the listening reader by making the expression easier to retain.

Operands within terms had the same amplitude patterns as those outside complex objects. Another decision was made here to make the grouping within Mathtalk’s model of prosody more prominent. Objects within complex objects were spoken without any raised amplitude, and the breathiness of the voice was increased. This made the grouped objects appear as an ‘aside’. Advantage was taken of the ability of exaggerating the prosodic cues when using a speech synthesiser. It was hoped that not using raised amplitude within complex objects would not interfere with the establishment of a rhythm within the speech. The pattern of amplitude changes can be seen in Figures 3.1(a) to 3.1(c).

The only cases in which non-equals operators were stressed seems to be for purposes of contrast.

For example, $(a + b)(a - b)$ Expression 3.21, in which the 'minus' is stressed, presumably for contrast with the 'plus'. Such emphasis might draw the listener's attention to this feature, thus indicating its mathematical significance. In complex fractions the first syllable of 'over' is stressed. Again, this is presumably to contrast it with the same operator in a simple fraction, which is not stressed. This emphasis may help to establish the whole construct as a fraction, rather than a sub-expression, as the prosodic structure may suggest.

Amplitude may have been used to contrast superscripts in x^4n and x^{4n} (see above). The unary operator in Expressions 3.18 and 3.19 may have been emphasised to contrast it with the binary equivalent.

The example of using stress for the fraction line would be legitimate within the constraint of presenting only structural information. The first example, by indicating mathematical comparisons between the two parenthesised groups, is presenting some interpretive information. This latter case will be avoided. The other uses of stress could be useful, for example, for indicating when operands were at the base-level of the expression.

The earlier study by Streeter (1978) suggested that amplitude was the least important of the three cues for disambiguating structure. Rhythm patterns are influenced by stress (Halliday 1970) and rhythm has important implications in the perception of an utterance (see Section 2.6). For this reason alone, stress patterns should be included in the Mathtalk presentation, but any effect in the apprehension of structure would also be useful.

3.5.2 Conclusions and Discussion

Major prosodic features used in speaking algebra notation were correlated with syntactic features with relative ease. All the features described were consistently associated with the structure of the expression. In the majority of instances the temporal, pitch or amplitude cues were explicable in terms of an expression's structure. Amplitude seemed the most variable cue in this association, often being used to mark points of contrast. This apparent association of prosodic cues with syntactic structure offers an opportunity to develop an alternative method for presenting spoken algebra in synthetic speech.

The algebra notation makes the grouping of an expression explicit so the grouping structure within an expression is known. The prosodic cues described in this investigation were associated with this grouping, regardless of any mathematical meaning associated with the grouping. Thus, a simple model for the prosody of algebra could be implemented by making large-scale global or local prosodic changes at base-level operators and the boundaries of complex objects. At operands only minor pitch changes need to be effected.

The range of expressions covered in this experiment was not wide enough to be absolutely sure of all the rules described. The investigation has shown there is a firm foundation from which a set of rules for basic prosody of algebra can be implemented. A more extensive investigation of algebraic prosody would yield a better model. However, two reasons support the view that an implementation of prosody to indicate algebraic boundaries is possible. The rules described here tally with those described by other researchers and behaviour found in regular English (see Section 2.6). The second reason is that the model need not be perfect. The general behaviour of the cues has been described. They can be extended by applying the rules to new situations, transferring rules from regular English.

The model does not need to be ‘natural’. The purpose was to convey the structure of the expression to the listener. It would be helpful if the resulting prosody was ‘natural’ sounding, but less natural rules could probably be learnt by listeners if they were the best way of presenting the structure.

This study both supports and extends the rules described in earlier work (Streeter 1978; O’Malley, Kloker, and Dara-Abrams 1973). Like O’Malley et al., pauses seem to be a major indicator of group boundary. Some differences were apparent. Pauses were generally not seen within complex fractions, superscripts or parenthesised sub-expressions. Pauses were seen before sub-expressions nested within sub-expressions. Like O’Malley et al. longer pauses may indicate level of nesting.

More definite information has been derived for pitch behaviour over the whole expression and within a term. Additional information about the pitch behaviour of groups of symbols in different positions within the expression were also described. O’Malley et al. cite Pike (Pike 1945) as predicting sub-expressions being spoken at a higher pitch than the surrounding text. This was not seen. While this description of algebraic prosody is incomplete, these results, taken with pre-existing knowledge of intonation within regular English should be able to provide a complete set of cues for the algebra to be presented by Mathtalk.

This study was successful in demonstrating that the prosodic cues of timing, pitch contour and amplitude could be associated with structural boundaries within an algebra expression. Even with the restricted scope of the algebra used in the Mathtalk program, the rules derived above were not fully comprehensive and inferences had to be made about some cues and some observations redesigned to make the cues simpler to implement. It would be better that the prosody used in the Mathtalk program were as natural as possible, to make the teaching easier and the presentation more ‘pleasant’ and intuitive to use. Given the simplistic nature of this approach optimising some cues will be necessary.

3.5.3 Design Rules for Algebraic Prosody in Mathtalk

The following list of design rules can be derived from the observations above for implementation in the Mathtalk program:

1. A pause of approx 300 ms occurs before the explicit binary operators occurring at the base-level.
2. These pauses do not occur within sub-expressions, superscripts and fractions.
3. A pause of approx 300 ms occurs on the side of a relational operator juxtaposed to the longer side of an expression. By default the pause is before the operator.
4. Pitch declines throughout the expression.
5. Initial pitch is proportional to the length of the expression in syllables. If the length of the expression exceeds pitch range, then the first syllables are spoken with a rising tone encompassing almost the entire pitch range. (This latter feature was not implemented successfully. Instead, if the top of the pitch range was reached, pitch was held at that level until the structure of the expression allowed it to fall.)
6. Pitch fall is linear through the expression, unless broken by a base-level operator or complex group appearing at the base-level.
7. Pitch rises after such an operator.
8. Pitch fall within a term is linear.
9. Pitch fall is very sharp over the final term. This is because there is no rise at the beginning of the final term.
10. All utterances terminate at a fixed pitch, towards the base of the pitch range.
11. The first operand of a term is stressed by pitch and amplitude.
12. Operands within superscripts are also emphasised by amplitude.
13. Operands are not emphasised within sub-expressions and fractions. No other syllables are emphasised, except the *e* of *equals*.
14. Superscripts continue the linear fall of the term to which they are attached.
15. In complex superscripts, no pause is found. A pause terminates a superscript, if one is not supplied by a following operator.
16. Sub-expressions are spoken without pause and slightly faster.

17. The contents of complex objects are spoken with increased breathiness.
18. Complex objects at the start and end of an expression have a linear pitch fall. Those in the middle of an expression have a flat pitch.
19. For central sub-expressions, the start and end are marked by a pause (300 ms), a large pitch fall (two or three times that from operand to operand), and a slightly smaller pitch rise marks the end of a sub-expression.
20. Complex fractions are spoken in a similar way to sub-expressions. The operator ‘over’ is stressed and surrounded by pauses of 250 ms. The ‘over’ is also spoken at a higher pitch.
21. Simple fractions are spoken with a linear fall, no pauses and no stress on the operator.
22. The first operand in numerator and denominator are stressed in simple fractions.

Some printed features are replaced by lexical cues. A superscript is preceded by the phrase ‘to the’. A coefficient followed by a sub-expression has the word ‘times’ inserted. Both these cues could be said to adding some interpretation of the grouping, informing the user what that grouping means. The ‘times’ is retained as an extra cue to confirm the onset of a parenthesised group. The cue ‘to the’ was also retained. The speaker analysed in this study also used the cue ‘all’ when spreading the scope of a superscript over a complex group. This cue was also retained to add information to the prosody information.

In a restricted investigation of this sort, not all possible combinations of algebra notation will have been tested. The rules stated may not account for all these combinations. The restricted nature of this study did not allow all combinations of structure to be investigated. This was particularly true of nesting one type of structure within a similar form or within a different type.

Having found a series of simple rules for presenting complex groups, a problem arises. In general, complex groups are bounded by pauses and a pitch change. Within the group there are no pauses and the direction of pitch does not alter; it is either flat or falling/rising. If other structures are placed in such situations, the cues that may be used are severely limited. The lack of available cues in such a situation may mean that complex structures will be more difficult to present and perceive.

The nested sub-expression in Expression 3.11 indicates that long pauses may be used to indicate nesting, as proposed by O’Malley et al. In such a situation, pauses could be re-used within the complex group. Expression 3.11 also shows that a further pitch change may be used to group objects within another complex group.

Implementing such rules on a speech synthesiser has some advantages. Speed and pitch can be increased and held at consistent levels that would be unreasonable for a human speaker. This advantage will be exploited to exaggerate some cues, in order to retain a simple use of pauses at base-level operators and to mark complex groups.

3.6 Evaluating the Prosodic Component

The aim of this experiment was to compare the ability of prosodic cues and the lexical cues to present the structure of an expression. It was hypothesised that the prosodic cues would be at least as good as the lexical cues in indicating the structure of complex expressions. A second hypothesis was that lexical cues may disrupt the retention of the content of an expression. The content of an expression are the symbols that are arranged within the structure of an expression: Fractions, sub-expressions, superscripts and terms.

The idea that addition of prosody reduced the mental workload of the listening task was the final hypothesis investigated in this experiment. By dividing an utterance into meaningful chunks of information, which correspond to the structure of the expression, and affording the listener processing time by use of pauses, it was thought that prosody would reduce the mental workload associated with the listening task.

The pattern of stress within the prosodic utterance adds a rhythmic component to the spoken expression, making it easier to remember (Baddeley 1992). The reduction of lexical cues will reduce the volume of verbal information to be processed by the listener, also potentially reducing the amount of mental work to be performed.

Wright and Monk (1989) note a dissociation between qualitative and quantitative measures in usability. It is possible for users to perform well on a task, but find the task demanding and frustrating, taking more effort than the user expects. Simple measures of speed and accuracy might rate the interface highly, but the users' subjective rating reveal usability problems with the design. In this case, the lexical cues make the structure explicit, but the extra words may mask any advantage.

Studies of mental workload have attempted to capture this dissociation. Hart and Wickens (1990, p258) define workload as '…as the effort invested by the human operator into task performance; workload arises from the interaction between a particular task and the performer'. The assumption is that performing a task requires cognitive resources and that these resources are finite. As a task becomes more difficult, more of these finite resources are used in achieving the same level of performance. The ability of humans to devote more resources to achieve the same task can mask usability problems in an interface.

Ratings of mental workload were used in this experiment to capture the potential difference in workload implied between the lexically presented expressions and those presented prosodically. The NASA Task Load Index (TLX) (NASA Human Performance Research Group 1987) was used to assess subjective mental workload in this experiment. This offers a quick, non-intrusive method of assessing the subjective mental workload associated with a task. Given the increased load placed on internal memory due to the lack of an external memory, reducing mental workload by making structure easier to apprehend and the speech easier to remember is an important goal in speech interface design.

The NASA Human Performance Research Group (Hart and Staveland 1988) analysed workload into six different factors: Mental demand, physical demand, time pressure, effort expended, performance level achieved and frustration experienced. Bevan and Macleod (1994, p143) say that three of the subscales relate to the demands imposed on users in terms of:

1. the amount of mental and perceptual activity required by the task;
2. the amount of physical activity required;
3. the time pressure felt.

A further three subscales relate to the interaction of an individual with the task:

1. the individual's perception of the degree of success;
2. the degree of effort an individual invested;
3. the amount of insecurity, discouragement, irritation and stress felt.

These factors have a direct bearing on the usability of a speech based interface. Three standard usability measures are effectiveness, efficiency and satisfaction (ISO-9241 1993). Effectiveness can be measured by the recall task itself. Part of efficiency is the amount of mental resource used. Reliance on short-term memory by the listening reader means that efficiency in use of mental resources is very important. If fewer mental resources are used, then the efficiency, effectiveness and satisfaction associated with the interface can be increased.

When using TLX participants mark scales for each of the factors shown above. In standard TLX analysis, paired comparisons of each factor give weights for the importance the participant gives to each factor. The factors are multiplied by these weights and a mean taken to give an overall workload value. Byers, Bittner and Hill (1989) proposed that 'raw TLX' of a simple mean of the factors were as reliable as the standard two pass procedure. For the sake of simplicity the raw TLX scores will be used in this experiment. Five of the six factors described above were used in this experiment. One factor, physical effort, was omitted as the experiment demanded no physical input from the participant. This made the use of the raw TLX more pertinent.

3.6.1 Design

A recall task was used in this experiment. Such a task was reasonably ecologically valid. In a real world task, the listener will have to listen to and retain the whole of a spoken expression. This experiment used a single utterance, which would not be consistent with the real world task, but should have given a stronger indication of the problems to be encountered. Even if one form of presentation was better than the other, it was likely that not all the content and structure of large expressions would be recalled accurately after a single utterance.

Examinations of transcripts could provide insight into the types of structure that cause errors; types of error made with retention of content and the size of expressions that may be recalled accurately with a single utterance. Such categories of errors will show what problems have to be designed for in other aspects of the user interface.

The experiment used a split-plot design, with two between-groups conditions and two within-group conditions. In total, there were three conditions, a prosodic condition, where structure was indicated by the prosodic cues described above and by the rump lexical cues described in the prosody experiment.

The *lexical* condition used the lexical cues described in Section 3.4. Only the default prosodic style of the speech synthesiser was used in this condition. Naturally a human speaker using such lexical cues would probably also insert prosodic cues into the speech. This would not be the case with a speech synthesiser. This experiment aimed to determine which method was better to use in such a system. If prosodic cues were good enough on their own, no extra lexical cues needed to be used. If neither prosodic nor lexical cues worked in isolation, but a combination of the two might, then a further investigation would be required.

The third condition was a *no-cues* condition in which neither prosodic nor lexical cues were used. This acted as a control condition within the experimental design. It also gave an indication of the types of mistakes listeners would make when given little or no structural information. This condition gave a benchmark from which to measure the effects of the other two conditions.

The three conditions were split between two groups: A lexical-prosodic group and a lexical-no cues group. Both groups heard a lexical presentation, using the same expressions in each group, then either a prosodic or a no-cues presentation, with the two latter conditions in each group using the same expressions. A lack of significant difference between the two lexical conditions would indicate the lack of consistency between the two groups. The difference in scores between lexical and prosody in one group would show the effect of these presentation styles. Similar significant differences within the lexical and no-cues group would show the effects of those presentation styles.

To ensure any learning or fatigue effect was not responsible for difference to the first condition presented, tests for difference between prosodic and no-cues conditions were made. It would have been preferable to have a prosodic-lexical condition to test this more thoroughly.

The recall style of this experiment gave a rich set of data. Such data were difficult to mark as answers were rarely neither completely correct nor completely wrong. Questions were marked separately for apprehension of structure and retention of content. This enabled the two major hypotheses to be assessed separately. To be correct an answer had to be correct in both structure and content. Given the difficulty of the task neither factor had to be 100% correct. In general the answer had to have most (75%) of an expression's content to be marked correct for content. Similarly the major structural features had to be present, for example, base-level terms and complex structures. This marking scheme was subjective, but the expressions were marked independently to ensure consistency. To mark absolutely right or wrong would have missed many of the facets of the presentation styles. This was particularly relevant when only a single utterance was given.

A NASA Task Load Index (TLX) workload assessment (NASA Human Performance Research Group 1987) was used to provide a subjective rating for the task workload. Participants had to give quantitative ratings for five of the six workload factors described above: mental demand, time pressure, effort expended, performance level achieved, and frustration experienced. The factor of physical effort was not recorded as the task required no movement by the participant apart from writing down the recalled expression. 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. This was an added extra, giving a subjective preference for a condition. That a listener finds one condition easier than another gives some indication of the success of that condition. This was the usability measure of satisfaction.

Materials

Two matched sets of 12 expressions were presented. All expressions contained one or more fraction, parenthesised sub-expression or superscript. The fractions and superscripts could be either complex or simple. In the lexical set the scope of the complex objects were delimited using the lexical cues, as described earlier. The second set had these boundaries indicated by using only the prosodic cues described earlier and the minimal set of lexical cues. A second version of this set was prepared, with neither lexical nor prosodic cues (the no-cues condition). The only cues present in the no-cues condition were the minimal lexical cues present in the prosodic condition. The expressions used can be seen in Table 3.3.

The expressions were designed to represent those that may be found in an 'A' level algebra

	Condition	
Number	Non-prosodic	Prosodic
1	$(y+6)(y-6)$	$x=x^{n-2}+7x$
2	$y+\frac{6}{y-4}$	$x+5(x-5)$
3	$y=(x-9)^{k+3}$	$(x+2)^2$
4	$3^{y+6}=5$	$\frac{x+3}{x-6}$
5	$(\frac{4y}{7(y+2)})(y+9)$	$5+(x^2-9)^x+5$
6	$9(y-6)+3(y+7)^7=3$	$x=(y+5)^n-2$
7	$y=2(y^4-8(y+5)$	$(x+7)^{\frac{1}{2}}=y$
8	$\frac{2}{3}(3x+9)=x+3$	$4(x-9)+5(x+7)^2=0$
9	$(x-9)^3$	$\frac{1}{2}(2x+4)=x+2$
10	$3+(y^6-5)^{y+7}$	$\frac{3x}{(x+4)(x+7)}$
11	$(y+1)^{\frac{2}{6}}=x$	$x=3(x^2-8x+1)$
12	$y=y^n-9+4y$	$3(x+4)=7$

Table 3.3: Questions for the prosody evaluation experiment. Both conditions are shown in the order of presentation. The prosodic condition stimuli were those used for the *no-cues* condition.

manipulation exercise, for example Bostock and Chandler (1981). The expressions in each group were written to be similar but not duplicated. This was to avoid possible memory for expression form from one condition to another by the participants. The expressions were matched by independent assessors so that overall complexity was matched, rather than pair-by-pair matching for complexity.

Participants

The overall scheme for the evaluation of the Mathtalk program was as follows: There were three major components, the prosodic display, the browsing and the audio glance. Each of these components was designed and evaluated separately before a final evaluation of the integrated Mathtalk program. Blind participants were only used for the final evaluation. Sighted participants were used to evaluate the separate components. A large, homogeneous set of blind participants was difficult to find. The participants needed to be familiar with algebra notation to a reasonably high level and to have a high-level of computer skills. No such pool of participants was available locally and time and financial resources did not allow such participants to be brought in from a wider area. It was thought that sighted participants would allow for basic usability evaluation, but a valid testing of the final Mathtalk program would need blind participants. Using blind participants also presented practical problems of writing algebra expressions recalled from the synthetic speech.

It was assumed that the sighted participants had the same memory and hearing characteristics as potential blind users. There is little evidence that blind people, even congenitally blind, have enhanced memory capacity or hearing skills (Lowenfeld 1980). If this were true, it would be unlikely to compensate for the passive listening task that is needed when listening to spoken algebra. If sighted people's performance on the task can be improved then this is also likely to be true for blind users.

Two groups of twelve sighted, normally hearing participants were used. 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 with neither lexical nor prosodic cues (LN group).

None of the 24 participants originally had any experience with speech synthesisers. Before the full experiment began each participant was given extensive experience by listening to polynomial expressions spoken by the speech synthesiser used in the experiment. This procedure was used to ensure homogeneity in the participant group. This should have reduced the effects of learning during the first set of expressions in each group. All participants had mathematics qualifications of ‘O’ Level or above and varied from daily use of mathematics to infrequent use.

Equipment

A Berkeley Speech Technology Best Speech synthesiser (Berkeley Speech Technology 1986) was used to speak the expressions. The prosodic cues were inserted into text string representations of the expressions by hand. Appropriate software was not available for a software implementation. For the prosodic condition the appropriate commands were used to specify timing, pitch, emphasis etc (Berkeley Speech Technology 1986). A symbolic representation of the prosodic form may be seen in Figures 3.1.

For the non-prosodic conditions, the expressions were sent unadorned to the speech synthesiser. No punctuation was present in the string, giving the synthesiser no information on which to base any prosody.

Procedure

Each participant had the overall design of the experiment explained from a script. Three example expressions were used to illustrate the presentation methods. These expressions can be seen in Section A.2.1. These were spoken by the experimenter, in the appropriate style, and each point of the presentation explained and general rules given. Then the same expressions were presented using the speech synthesiser. After each presentation the experimenter spoke the expressions again. This procedure was repeated until the participant was happy about the presentation style.

The expressions were presented one at a time. The participant was asked not to write the expression down until the presentation was over. The expression was not repeated. The participant wrote down

	Group			
Factor	**LP**		**LN**	
	Lexical	**Prosody**	**Lexical**	**No-cues**
Structure	0.67	0.88	0.62	0.37
Content	0.52	0.79	0.45	0.67
Overall	0.49	0.76	0.4	0.35

Table 3.4: Mean proportion of correct answers (n=12) for structure, content and overall scores for each condition in each group. (LP = Lexical prosody condition; LN = Lexical no-cues condition).

his or her recall of the expression and was told to use either question marks or ellipses to denote any missing objects from the expression. The experimenter remained silent until the participant indicated he or she had finished. There was unlimited time in which to write down the expression.

After each condition the participant was given a set of scales for marking the TLX scores. A sample scale for the TLX evaluation can be seen in Section A.2.3. A set of explanations was available for each factor (see NASA Human Performance Research Group 1987). After the second condition the participant was asked to mark a final scale giving preference for each condition.

3.6.2 RESULTS AND DISCUSSION

The answers were marked separately for recall of structure and content. Table 3.4 shows the mean proportion of correct answers for the factors of structure, content and the overall scores for each condition in each group. Student T tests were used to test for a significant difference between the means. The results of these tests are summarised in Table 3.5.

These tests (see Table 3.5) showed that participants performed significantly better on the recall task, for both structure and content, when hearing expressions presented with prosodic cues than lexical cues. The test for significant difference between recovery of structure for the prosody vs no-cues comparison was not performed. As the prosodic condition proved better than the lexical and the lexical better than the no-cues condition, this test was unnecessary. Overall those using lexical cues did not perform better than when using no cues at all. However, when using lexical cues more structure was recalled than with the no-cues presentation and this situation was reversed with more content being recalled under the no-cues condition.

The two lexical conditions were not significantly different, indicating that the two groups were comparable. The prosodic and no-cues conditions were significantly different on all factors, suggesting that the improvement due to the addition of prosody was not due to any learning effect. Even though the stimuli were the same for the prosodic and no-cues conditions, the style of presentation had a significant effect on their recall. Thus, the improvement in the prosodic

Overall

Condition	T	DF	P
Lexical vs Prosody	6.38	11	0.00003
Lexical vs No-cues	1.4	11	0.094
Lexical vs Lexical	1.95	22	0.06
Prosody vs No-cues	8.72	22	0.00

Structure

Condition	T	DF	P
Lexical vs Prosody	7.97	11	0.000003
Lexical vs No-cues	4.53	11	0.0004

Content

Condition	T	DF	P
Lexical vs Prosody	6.04	11	0.00004
Lexical vs No-cues	7.09	11	0.00001
Prosody vs No-cues	-2.48	22	0.021

Table 3.5: Results from the Student T tests performed upon the results of the comparisons between conditions. T = value of T test; DF = degrees of freedom; P = probability.

condition was not simply due to the lack of lexical cues, but to the prosodic cues themselves.

As can be seen from Table 3.4, participants were able to recover more structure and content from expressions heard with prosodic cues than lexical cues (0.67 vs 0.88 T = 7.97 for structure; 0.52 vs 0.79, T = 6.04 for content), and thus performed better overall. The ability of prosodic cues to indicate structure exceeded expectations, being much better than the lexical cues. Thus the original hypothesis that prosodic cues would be at least as good as lexical cues was rejected in favour of a finding that prosodic cues were in general better than the lexical equivalent for conveying the structure of an expression.

The second hypothesis, that prosodic cues would enhance the recall of an expression's content, was also demonstrated. The participants performed significantly worse in the *no-cues* condition of the LN group (0.4 vs 0.35, T = 7.09). However, whilst recovering less structure, those in the no-cues condition recalled a larger amount of content (0.45 vs 0.76, T = -2.48). The only difference between the lexical and no-cues condition was the presence of lexical cues. This suggests that the lexical cues interfered with the retention of content by the listeners.

That the performance on recall of structure was worse in the no-cues condition was not surprising. Much of the information was simply not present. However, some of the residual lexical cues such as 'times', 'to the' and 'all' enabled some structure to be included and sometimes this was done correctly.

So prosodic cues can be included into a synthetically spoken presentation of algebra notation to enable listeners to recover syntactic structure and retain content. The recall was not wholly reliable. The following section describes the types of error made by listeners during the recall tasks.

	LP Lexical			LP Prosodic			LN Lexical			LN No Cues		
Question	**S**	**C**	**O**	**S**	**C**	**O**	**S**	**C**	**O**	**S**	**C**	**O**
1	12	12	12	12	12	12	5	11	11	12	11	5
2	12	12	12	11	11	11	1	12	11	11	12	1
3	9	2	2	12	12	12	12	1	1	6	12	12
4	12	12	12	12	12	12	10	9	9	12	12	10
5	6	2	2	8	7	5	1	1	1	7	3	0
6	5	0	0	12	10	10	2	1	0	3	9	2
7	0	0	0	12	12	12	6	1	0	0	11	6
8	7	5	4	6	4	3	0	3	3	8	0	0
9	11	11	10	11	8	8	2	12	12	12	3	2
10	5	1	1	10	8	7	1	0	0	4	6	1
11	9	9	9	8	6	6	3	7	3	5	7	2
12	8	9	7	12	12	12	10	7	7	9	11	10
Total	96	75	71	126	114	114	53	65	58	89	97	51

Table 3.6: Total numbers correct for each question in each condition. **S** = Structure; **C** = content and **O** = overall. LN = Lexical No-cues condition; LP = Lexical Prosody condition.

Types of Error in Recall

A more detailed examination of errors in each condition was informative about the process of listening to algebra. By observation, it can be seen from Table 3.6 that in all three conditions, errors were clustered on certain questions. A selection of those answers that failed in recall of structure were examined. A deeper, psychological investigation, whilst interesting, was not within the scope of this project.

It was important to know where the prosodic component did not perform satisfactorily, so that the rules for algebraic prosody could be improved or different design solutions proposed. Examination of the structural errors in the lexical condition gave insight into why that presentation style proved so inadequate for the task. These reasons could be generalised to all presentations, so aiding the design process. The control (no-cues) condition showed some of the problems of an ambiguous presentation. The descriptions also demonstrate the types of error made in recall of an expression's content.

Prosodic Condition from LP Group

There were relatively few structural errors in the prosodic condition of the LP group, but two important lessons can be learnt for the design of the user interface.

In Expression 5, $5+(x^2-9)^x+5$, which was spoken as:

‘ five . ↑ plus . ↳ *x squared minus nine* ⬏ all to the x . plus five ’,

four responses showed the mistaken grouping of the terminal $+5$ into the superscript. Two constants of 5 appearing at either end of the expression may have misled the listeners. However, users will have to be able to discriminate such ‘unlikely’ forms in the intermediate stages of problem solution.

A more likely reason for this mistake is the lack of redundancy in the juncture cues between the superscript and the following constant. Most terms are preceded by a pitch rise, as well as a pause. For final terms this pitch rise is missing. This may lead the listener to group the final term with the superscript.

Errors in recovery of structure also occur when nesting of structures causes a similar fall in redundancy in use of the prosodic cues. Within the denominator of expression 10, the flat pitch and pauseless speech means that only ‘times’ and speed remain to indicate the grouping. It is unlikely that prosody will ever prove entirely reliable in facilitating discrimination of structure. However, the cues remaining when redundancy is reduced will be exaggerated to aid parsing.

The other principal error found in the prosodic condition was to increase the scope of a superscript when the cue ‘all’ was used. For example in Expression 8, $4(x-9)+5(x+7)^2=0$, which was spoken as:

‘**four**. ↳ *x minus nine* ⬏ . plus **five** ↳ *x plus seven*. ↘ **all** squared . equals zero’;

Four of the listeners mistakenly included the whole of the left-hand-side prior to the superscript in parentheses. This error occurred frequently in all expressions that used the cue ’all to the’. It seems unlikely that this was due to the misleading prosodic cues, because most of the participants successfully inserted the correct parentheses. The mistake was probably due to the cue ‘all’ in the ‘all to the two’ in the utterance being strong and covering the widest possible scope. This was a consistent error in other conditions.

Alongside the structural errors, there were several types of content error. These can be put into the classes of omission, substitution and transposition errors. These are typical ‘slips of the ear’ as described by Garnham (1989). These types of error are probably unavoidable in a simple full utterance of anything but the shortest of expressions.

The data in this condition result from a single utterance and in ecologically valid situations, the reader will be able to take repeated views of an expression, just as a sighted reader will repeatedly sample the printed page. However, in general, whilst recovery of structure was good, it was not completely reliable. Some mistakes will always be made, even when experience increases. Other design solutions will be needed to enable complete discrimination of structure and recovery of content.

Lexical Condition from LP Group

Only those questions in the lexical condition of the LP group were examined in detail, as the lexical condition in the LN group was comparable. Error rates for all three factors were lower on shorter, less complex expressions. In this sense, the lexical cue presentation was adequate, but the mental workload analysis, presented below, indicated that this style was less easy to use. When the expressions became longer many structure and content errors occurred.

There was a common error of including an n at the end of some complex objects. It was assumed that this n came from the 'en' of the 'end *structure*' tags. The lexical cues, mixed intimately with the algebra content, seemed to have caused confusion. This sort of error might have decreased with practice, but was obviously a problem to be avoided.

The other content errors were essentially the same as those seen in the other conditions, but greatly exacerbated in the lexical condition. Many transpositions were seen and substitutions of one character for another of the same type were frequent. Overwhelmingly, the main content error was omission. The following example demonstrates this feature more thoroughly.

The recall of Expression six was typical of many in this condition, $9(y-6)+3(y+7)^7=3$ was spoken as: 'nine times the quantity y minus six end quantity plus three times the quantity y plus seven end quantity all to the seven equals three'. Most participants recovered the parentheses. However, the recovery of the two sets of parentheses was in contrast to the almost complete loss of content from within these structures. There was frequent loss of information from within complex structures surrounded by lexical cues. Material from either end of an expression seemed to be recalled better, especially if it was structurally simple.

Many of the errors in this condition were probably due to the overwhelming of the listeners' mental resources because of the large amount of speech. There were several examples of almost complete loss of information in the responses, something that did not happen in either of the other conditions.

As well as the large amount of information simply overwhelming listeners, it seemed likely that the suffix effect (see Section 2.3) was responsible for many of the errors. The uttering of an end tag, with no pause to afford processing time, may have overwritten the contents of such objects in listeners' short-term memory.

The nesting of structures caused considerable problems for most participants. When structure was recovered, it was often flattened out to reduce the complexity of the expression. Participants frequently completely failed to recall anything but the short, simple parts of complex expressions losing all structure and content from the complex parts. Again, the large amount of information was thought to overwhelm the listeners' memory resources.

It is recognised that in natural language processing, nested sentences are difficult to process and comprehend (Garnham 1989). Such expressions were probably an equivalent of such sentences. The response of the many responses seemed to suggest that some aspects of the presentation can be recalled, but not all. In making a choice to retain or rehearse one aspect of the expression others were lost.

The lexical condition again demonstrated the mistaken use of the cue ‘all’ to emphasise the scope of the superscript over complex objects. Many of the structural errors would have been removed if this cue had not been used. However, many other structural recall errors were made, so that the lexical condition would still have performed worse than the prosodic condition.

The results underline the need to avoid the cue ‘all’ from the interface. Also, the need to avoid any superfluous lexical cues from the speech was demonstrated. The prosodic cues add more usable information to the presentation than the lexical cues, but do not increase the amount of words spoken. This is the principle of minimum speech and maximum information.

No-Cues Condition from LN Group

This condition was less informative to the design process, but some interesting responses were seen. This description was included to complete the picture of what happened during the experiment.

Expression 2: $x+5(x-5)$ was spoken as ‘x plus five times x minus five’ and the imposition of structure is indicative of recall in this condition. Nine participants gave the following response:

$$(x+5)(x-5)$$

The cue ‘times’ would have informed the listeners that there was a parenthesised group at the end of the expression, but not at the start. The responses all seemed to have assumed the standard form of a ‘difference of two squares’. In the prosodic condition the same expression was recalled correctly by all but one participant. So the prosodic cues were strong enough to override a potential tendency to impose an ‘expected’ structure on an ambiguous expression.

The listeners could use the residual lexical cues as some indication of what structure was present. Responses varied considerably, but some features emerged. Superscripts were typically kept as simple as possible, whereas sub-expressions and fractions tended to encompass as much of the recalled content as possible. In other cases, structure was imposed to give ‘usual’ forms, as in the difference of two squares seen above.

Yet again, the no-cues condition demonstrated the danger of using the word ‘all’ in any of the cues. The lack of cues in the speech made most of the expressions’ grouping ambiguous, except where other features could allow the listener to infer structure. The participants varied in how they imposed structure on the ambiguous utterances. This was a useful demonstration of the types of errors that a presentation with ambiguous grouping can cause the listening reader.

Recovery of content from the expressions was generally good, but the same classes of errors occurred in this condition. Large scale loss of content from an expression was rare, except when a long undifferentiated stream of speech was heard. The lack of lexical cues probably meant listeners were less likely to lose information by overload and the suffix effect. However, the lack of pauses and other cues, which may afford processing time and allow anticipation of structure, may have precluded the level of recall seen in the prosodic condition.

Mistakes in the recall of content were made throughout the three conditions. The errors were worst in the lexical condition and amongst the longer expressions of all conditions. Such errors are inherent in such a listening task and demonstrate the need for the listening reader to be able to visit any part of an expression to examine smaller portions of content.

Task Load Index Results

The TLX evaluation bars were scored on the scale 0—20. Marks placed between bars were rounded up. Difference between evaluations for each index were tested, using independent T-tests, against the null hypothesis that there was no difference between the two values. An overall mental workload was calculated for each condition by taking the mean of the five factors used (inverting perceived performance level).

The overall mental workloads were 11.4 for the prosodic condition; 11.47 for the no-cues condition and 13.58 and 14.62 for the lexical conditions. The prosodic condition had a lower mental workload than the lexical condition (t=5.665; df=11; P= 0.000073). The prosodic condition did not have a lower overall mental workload than the no-cues condition (as the means were identical). The no-cues condition had a lower mental workload than that of the lexical (t=3.688; df=11; P = 0.0018). The overall subjective mental workload ratings had the following ordering:

- Prosodic $=$ No cues $<$ Lexical cues.

These results confirm the hypothesis that the use of prosody, instead of lexical cues, in the spoken display reduces the mental workload required for the task. Simply using the default prosody given to an unpunctuated algebraic utterance and inserting lexical cues to disambiguate that utterance severely increases the mental workload requirements. This ordering was reflected in the recall data shown above. The lack of disambiguating cues in the no-cues condition obviously led to many errors. However, the difference in recall of content between the lexical and no-cues conditions was marked, suggesting that participants in the lexical condition had to work much harder to retain and

Lexical-Prosodic Group			
Factor	T(11)	P	Percentage Change
mental demand	3.294	<0.01	14.17
time pressure	4.492	<0.01	16.67
effort expended	2.209	<0.05	6.25
performance level	-5.54	<0.01	-22.5
frustration	3.17;	<0.01	21.67

Table 3.7: Percentage changes from lexical to prosodic conditions.

recall this information. Similarly listeners in the prosodic and no-cues conditions worked equally hard, but the prosodic group had more information available and recovered more of that information.

Tables 3.7 and 3.8 summarise the results from the workload assessment. The raw scores for the TLX rating can be seen in Section A.2.3.

The workload assessment indicates that the expressions presented with prosodic cues were considerably easier to use than those presented with lexical cues. This result is borne out by the performance in the task. The perceived performance level was 22% higher and the actual performance level was 27% better in the prosodic condition. It is interesting to note that the frustration level was particularly reduced in the prosodic condition (21.67%), confirming that the prosodic voice was easier and more pleasant to use.

Whilst the mental demand was significantly lower in the prosodic condition, the actual level was still quite high, indicating that the listening task, in itself, is difficult. It should be noted that whilst performance was better in the prosodic condition, not all the ‘correct’ answers were 100% right. Any of the longer expressions typically had at least some content errors; confirming that the listening task was difficult. This is probably because a lot of effort has to be expended retaining the expression in memory. However, the reduction in mental workload should be a great boon to the listening reader. This load should be further reduced with the addition of control over information flow, which would generate speech at a pace comfortable for the listener.

In the LN group the TLX suggests that in the no-cues condition the task was easier. However, this time there was a disassociation between the perceived performance level and the actual performance level. Despite the often ambiguous nature of the presentation in the no-cues condition, subjects found the task easier and thought they had performed better. This is probably because the lexical cues were very intrusive and made the task of remembering the expression’s content much harder. Even in the no-cues condition, participants may have thought that the residual lexical cues were enough to infer the structure of an expression. It seemed that the participants knew they had

Lexical-No Cues Group			
Factor	T(11)	P	Percentage Change
mental demand	4.809;	¡0.01	18.75
time pressure	3.494;	¡0.01	10.42
effort expended	2.912;	¡0.02	12.08
performance level	-3.48;	¡0.01	-14.2
frustration	2.754;	¡0.02	19.17

Table 3.8: Percentage changes from lexical to no-cues condition.

lost a lot of information in the lexical condition and this may have been reflected in the higher perceived performance level and lower levels of frustration in the no-cues condition.

The overall preference for each condition had the same ordering as the overall correct answers. For the LP group the mean expressed preference was 16.17 (where 20 indicates a preference for the prosodic presentation). For the LN group the preference was 13.42 (where 20 indicates a preference for the no-cues presentation). This reinforces the view that the prosodic condition gave a most satisfactory presentation.

Summary of Evaluation of Prosodic Component

In summary, the performance on recall of structure, content and overall score for each condition was:

Structure: Prosody > Lexical > No-cues;

Content: Prosody > No-cues > Lexical;

Overall: Prosody > Lexical > No cues;

Mental workload Prosodic = No-cues < Lexical.

Preference Prosodic > No-cues > Lexical.

This experiment has shown that prosody was able to indicate much of the structure of an expression to the listener. Some of this effect was due to the lack of lexical cues, but the prosody itself added something. The study of Streeter (1978) showed that listeners could recover similar structure from human speech and a corresponding effect has been shown using a simple set of rules from algebraic prosody used in a synthetic speech presentation.

Two factors may have been responsible for the increase in recall of content from the utterance. The prosody chunks the utterance into meaningful subunits of information, rather than a single stream seen in the no-cues condition. The single utterance probably overwhelmed the working memory capacity of the listener. Breaking down the utterance into chunks, within a rhythmic structure, may have helped the listener retain the content. In addition, the lexical cues may have caused loss of information due to the suffix effect. The lack of extra words and the presence of pauses may have afforded the listener processing time to store the information.

Prosody may replace some of those printed features described by Kirshner (1989). In print, the least precedence operators are surrounded with white space to divide the expression into chunks that correspond to the initial parse points of an expression. The pauses between terms in spoken algebra may well serve the same purpose. Objects that are to be multiplied or divided are grouped together more closely in speech, as they are in print. The prosody groups objects together in an expression and this grouping is linked to how they should be parsed in an expression.

Prosody can be said to improve the role of spoken algebra as an external memory. Printed algebra presents an expression in a manner that shows the grouping and facilitates parsing. Prosody can be said to perform both these tasks, even if not as perfectly as the printed notation. More importantly the external memory relieves the reader of the burden of remembering large amounts of information. Prosody makes the task of remembering such information less demanding. Whilst the listener, at this point, still has to do the remembering, as the display cannot be reviewed, prosody indirectly introduces one aspect of external memory by making the expression easier to remember.

3.7 Conclusions

In this chapter two major questions have been investigated: What information should be displayed and how should this information be presented? The print informs the reader about the grouping of symbols within the expression and facilitates the parsing of the expression. The print acts as an information resource for the reader. It is the reader who brings his or her own mathematical knowledge to give meaning to the written expression. Two fundamental design principles can be formulated from this analysis:

- The auditory interface should present the notation not the mathematical meaning of the expression. Minimal syntactic interpretation should be used in the presentation, constrained by the usability of that presentation.

- The display should present the grouping of and the association between the symbols in a manner that the listening reader can recover the structure and retain the content of the expression. It is the user who does the reading, not the computer. The user should not be ‘read to’.

Two methods of presenting the structure of an expression were described and investigated. A subset of Chang's rules for inserting lexical cues were presented. These were chosen to reduce the amount of mathematical meaning in the presentation and also reduce the number of lexical insertions as far as possible. The concept of simple and complex notation was used as a guiding principle for when to insert information about structure. When structures are simple, the number of cues can be reduced, to avoid problems of information overload and invocation of the suffix effect. The need to reduce the amount of information leads to three more design principles:

- When more than one term is grouped together by explicit parsing marks or by spatial location, then cues should be inserted to delimit the grouping within the symbols.
- The display should be designed for minimum speech and maximum information output.
- The most common state of a particular object should become the unadorned, default presentation.

The second method for the display of the expression was the use of prosodic cues within the speech. A set of prosodic rules were derived from recordings of spoken algebraic expressions. These rules were consistent with, and extended, those rules proposed by Streeter (1978) and O'Malley et al. (1973).

It was found that the prosodic presentation enabled listeners to recall more structure and content than when using a lexical cue presentation. Prosody also reduced the associated mental workload, compared to the lexical presentation. The success of the prosodic presentation give more design principles:

- Prosodic rules should be derived that can be associated with the grouping within the information to be presented.
- These cues should be added to facilitate the disambiguation of complex objects and simple chunks within the information.
- The prosodic cues add some of the qualities of an external memory to an auditory display.

The addition of prosody makes the display more usable. All three measures of usability were enhanced: Effectiveness, as measured by recall scores; Efficiency as demonstrated by reduced mental workload and satisfaction by the participants' overall preference.

Performance on recovery of structure was comparable to the 75% reported by Streeter (1978). A similar figure was found for the recovery of content. A sighted reader would not be so prone to such errors (though there are 'slips of the eye' (Garnham 1989)). The eye can revisit and select portions of the expression with ease, so that no reliance has to be made upon potentially faulty internal representation. This active reading is one of the design goals for the Mathtalk program and is the subject of the following chapter.

Chapter 4

Controlling the Information Flow

4.1 The Need for Browsing

Simply improving the spoken presentation of algebra notation is not enough to allow visually disabled people to read by listening. The last chapter described how prosody was used to improve the apprehension of structure, the retention of content and the reduction of mental workload. Prosody did not solve all the problems of display. Some complex structures remained ambiguous and many expressions were too large to be reliably retained. No matter how good the presentation, the listening is still passive and error prone: What is needed is active reading. Control of information flow makes a passive listener an active reader. 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 that give the information only when it is needed.

Such access should relieve the reader of the burden of remembering all the material. Instead, the listener could use the display as a memory, thus freeing cognitive resources for mathematics. Such a feature is a vital mechanical aspect of reading.

This chapter describes the development of the control of information flow within the Mathtalk program. First the use of browsing to afford control is justified and the basic style of browsing developed. After the functionality of the browsing has been introduced the development of the manipulation style is described. The command language developed to mediate control and the browsing components went through several cycles of design before a final version was produced. The evaluation of the browsing functions and associated browsing language are then described.

4.2 The Nature of the Control

In a mechanical sense reading is a process of controlling information flow. The aspects of understanding 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. A simplistic view of reading is browsing or movement through the information space. It is the control of this movement that makes reading active and is the mechanical aspect of reading. Reading algebra may be viewed as a structure based process. So control over the information flow could be offered by giving the reader the ability to browse the structure of an algebra expression.

Such browsing gives a suitable task-based mechanism for reading. This reading of an algebra expression is a process of parsing the grouping given by spatial location and by explicit marks to derive some mathematical notion or accomplish a manipulation task. Such tasks are based on structure and Kirshner (1989) has demonstrated that the style of printed algebra facilitates parsing or reading by making the structure easier to access. As a general principle, the Mathtalk program aims to enable the reader to use his or her mathematical knowledge in combination with the information on the page. So the approach taken for reading in the Mathtalk program is to view it as a structure based activity.

Readers obviously vary in how they extract information and use of strategies for achieving mechanical 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, as described by Bates (1989).

Browsing functions give the potential of control, but the user needs to be able to manipulate those functions. In the Mathtalk program control is mediated with a command language, issued through the computer’s keyboard. Several factors led to the choice of a command style interface. A simple practical consideration was that a command line style needed no extra hardware for implementation. The aim of this component was to demonstrate that additional control, based on speed and accuracy, gave better, active reading. If successful, it would not suggest that a command language style was necessarily the best option. Further research would be needed to indicate whether a more direct manipulation approach such as Aron’s speechskimmer (1993) or some sort of pointing would be more appropriate. Using a command language with a set of browsing functions was thought to be the simplest manner to design for speed and accuracy believed to be necessary for active reading. Care was taken to design the best possible command language, being consistent in execution, feedback and with the user’s notion of object and move labels.

4.2.1 Hiding Complex Objects

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4.1}$$

The hiding of complex objects is an emergent property of the way expressions are represented in the system as a series of levels, the deeper levels representing the nesting of the complex objects within the whole expression. A complex object is as described in Chapter 3. So at a level higher, a complex object is seen as a single object, where within that object it is a series of objects. At the higher level the complex object is a single element referred to only as its type. So at the top-level (base-level) of the expression 4.1 there are only three elements: the character 'x', '$=$' and 'a fraction'. The last element contains the whole fraction or can be said to 'hide' the fraction.

This hiding of complex objects has two distinct advantages: First, it allows greater control over information flow than that given by browsing alone. In addition it should facilitate disambiguation of grouping. Hiding of complex objects encapsulates the notion of *levels*. The nesting of structures within an expression is made explicit to the listening reader, just as the printed form of an expression makes such structures explicit to the sighted reader by use of explicit parsing marks. For example, as the reader moves along the base-level, the whole expression can be seen in three objects and that all symbols, 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 or by displaying the object from the higher level. In addition, the user may also easily skip over that object in a single move, rather than having to move through all of its contents.

This division of an expression also adds another factor to the complexity of the design for a browsing language to mediate that control. Care will have to be taken to ensure the potential advantages of hiding objects are not dissipated by additional complexity in the browsing language.

An early design decision was made as to how these levels would be browsed. The listening reader would be made to explicitly move into and out of these structures. The alternative would be to automatically move into a complex object when it was met, and then to move automatically to the following object at the previous level when the end of the complex object was reached by individual browsing moves. Take the following expression

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

which would be presented at the base level as ‘3 times a quantity = 7’. When the reader moved onto the ‘quantity’ two choices would be available: To view the sub-expression from outside or to move into that sub-expression. Either alternative makes the structure explicit. If the reader moved into the sub-expression, that would be the scope of any commands issued. For example, speaking each object in turn would eventually bring the reader to the ‘4’ at the end of the sub-expression. Trying to have the next item spoken would simply give an ‘end of level’ message. Moving directly onto the ‘equals’ of the base-level could cause confusion. To avoid this, an explicit ‘exit sub-expression’ command had to be invoked, to reinforce the structure. Automatic movement into and out of hidden objects would also have contradicted the element of control whereby a user could easily skip over complex objects.

An extension of this situation was to make mandatory that commands only applied to each scope within nested structures. So within a deeply nested structure, the user would have to ‘climb’ out of each level from the bottom up, to return to the base-level. This was used as a device to reinforce orientation for the user. This had the potential of being frustrating or inconsistent with the reader’s notion of the task.

In algebra notation the structure is of such vital importance and each object within that structure equally so, that to be overwhelmed by a flood of symbols or to miss any part of the structure could be disastrous in comprehending that expression. For these reasons the notion of hiding information in lower levels and making these levels explicit and mandatory was adopted.

Speaking the contents of a complex object would utter the simple contents in full, but still hide complex objects at a lower level. Again, this strict hiding of objects may be frustrating, as readers would have to move into a complex object to speak other complex objects at a lower level. This strict approach was taken to be consistent with the notion of hiding complexity.

4.2.2 What Should be Spoken

A basic design question is what should be said during the browsing. The basic answer was to speak the object that has become the focus of attention. The following expression can be used to highlight some design questions:

$$x > y.$$

If the user was moving forward through the expression character by character we would have: ‘x’, ‘greater than’ and ‘y’. What should be spoken as the reader moves backwards through the list of characters? The same rendering as above, saying what the symbols actually represent, gives us: ‘y’, ‘greater than’ and ’x’, which effectively ‘means’ the opposite of the first rendering, if Mathtalk is giving the meaning of the expression. If this were the case $>$ would be spoken as $<$ when moving through the expression backwards. What would happen if a user moved forward, then backwards, halted on the $>$ and then asked for a repetition? The symbol would be rendered as 'greater than', then 'less than' and finally as 'greater than' again, which is potentially confusing for the listener. If the principle that the user does the reading, not the system, is invoked then it is left up to the user to do the interpretation . This is exactly the situation with sighted readers.

Another question was the association of operators with terms and the order of speaking operands within terms when moving backwards and forwards. An example is the expression

$$ax^2 + bx + c = 0$$

If the focus was on bx and a move made to the previous term, should the output be 'plus x super two a', 'plus a x super two', or should only the operator to the left of the term be spoken so only 'a x super two' would be spoken.

The general rule in the Mathtalk program was to speak the operator followed by the term. For a+b-c this is 'a' '+b' and '-c' when moving forwards. Speaking the operator to the left of the term, matches the association in spoken algebra (see Chapter 3). When moving backwards this would give 'c', '-b' and '+a'. This method was a choice of speaking a term either from the end or from the start. The operator to the right was spoken, to emphasise the movement backwards, but the operands of the term were spoken forwards to match output between directions. The other choice was to move backwards and speak forwards.

The finest grain of browsing is a character. Defining a character was not as simple a task as it may seem. Some symbols have more than one symbol in close relation; other symbols are groups of symbols or structures reduced to one labeled character by the hiding of complex objects. So labeled complex objects were regarded as single characters. For example, asking what the current character was may reveal 'x' as easily as it may reveal 'a fraction'. Similarly, with the expression x^2, asking for the current character may reveal 'x'. In this instance the reader would lose information if a strict notion of a character was taken. Superscripts and unary operators were taken to be part of a 'character'.

Moves in and out of complex objects were also announced by speaking the type of the object. This was used to reinforce the move to aid orientation.

4.2.3 Elements of the Control

Control over the information flow was to be offered by a series of browsing moves. These moves were to be independent of the means of mediating the browsing. The browsing functionality was designed first, with the reading tasks as the goal, and a mediating command language added afterwards. The aim was to allow the user to visit all parts of the expression with speed and accuracy: These are the transitions in browsing described by Kwasnik (1992).

The moves were to act directly on the target, without the reader having to move through all parts from the current focus to the target location. This was to avoid offering superfluous information. Again, the general principle of giving maximum information with minimum speech was used.

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 permanent fashion. The nature of the external memory and the visual system means that the current focus is usually available. The auditory display is transient, so the current selection or focus of attention also disappears. So one demand of auditory browsing or interaction is a request for the current 'thing' being viewed or displayed. So we have current, next and previous as basic moves or transitions.

These are often small scale, local moves. Control will also involve larger scale shifts in the focus of attention. These can be adequately captured by the moves beginning and end. As described earlier, the hidden objects necessitate the moves into and out-of to be incorporated into the set of actions. These common moves can form the basis of the browsing through the algebra expression.

The list of objects on which the moves can act are simply those that the Mathtalk program covers. To be consistent with the output forms, the definitions of the target objects can be found in Section 1.4. Parenthesised sub-expressions are referred to as 'quantity' and superscripts adopt the short form 'super'. The hidden object concept (described above) makes another object of 'level' useful as a target object.

4.3 The Command Language

The requirements for the browsing functions above give the means of controlling the information flow. This control needs to be mediated. Unlike visual reading this control must be mediated externally, as opposed to the sub-conscious, mental control, cued by the visual medium, of eye movement over a page. 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 may have a rich structure so any command language to manipulate the necessarily large number of browsing functions will itself be large and complex. Such a language will have to be carefully designed to ensure usability. The language must be simple in design to be reliable, learnable, quick to issue to give the speed component of control that capitalises on the accuracy given by the structure based browsing.

It was not the aim of this part of the research to claim that a command language was the best form for control of information flow. It was more that improvement of that control would improve reading. Browsing functions combined with a command language provide a simple method of implementing this control with no extra hardware and basic research. As discussed above, other methods may provide better means of control, but the use of browsing and a command language provide an interesting case for affording control and similar design questions must be answered whatever the method of control. The principal questions are how to:

1. cover the wide range of potential structures;
2. provide this coverage in a reasonably learnable, predictable manner;
3. enable fast and accurate control;
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.

The reading process is of primary importance. This means that the mediation of control via browsing should not interfere with the reading process. On a very simple level this means that the command language that mediates the control must be very easy to use, learn and adapt. The feedback from the control must contribute to the reading and not interfere with unneeded information. Each user of algebra notation is likely to use different strategies for reading and performing mathematical tasks. So the provision of a series of high-level reading strategies may not suit the widest range of readers. The approach taken with Mathtalk has been to implement a set of low-level browsing functions, which, if quick and easy to use, could be built up into higher-level reading strategies by each reader. The command language will provide the browsing moves described by (Bates 1989).

Higher-level strategy or stratagems will have to be built up by the reader him- or herself. This means the browsing system will be highly flexible. A danger with this approach is that many commands may have to be issued to achieve each sub-goal within a mathematical task. This may not be a problem when the task is reading alone, but whether this remains true for the more complex tasks of writing and manipulation will be the subject of future research.

Internal consistency in the language should make it easier to learn. For example, all commands are formed in a similar manner. As well as learnability, such consistent design should reduce the number of errors made by users when issuing commands.

The design can also be consistent with other features: Consistent with the task or consistent with some other browsing paradigm. It was thought that the structure based browsing would be consistent with usage of algebra notation. Algebraic manipulation tasks are expressed in structural terms. So allowing reading and, by extension manipulation, via the structure should be consistent with usage of the notation. The browsing functions were designed around the structure of an expression, so the command language should match this design. The style of the command language could be consistent with styles already known to potential users. This further level of consistency could build upon users' experience with both tools such as word-processors and interaction with human readers. Both these options were explored within Mathtalk.

4.3.1 Unconstrained Browsing

The basic set of browsing moves are described by the actions and the targets described above. 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. This language must cover this richness yet be simple enough such that it does not itself interfere with the reading process. The final 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 that any visually disabled person could use when interacting with a human reader. For example 'beginning of expression', 'next term' and 'previous character' emerge easily from the set of actions and targets as intuitive commands.

Table 4.1 shows the action and target words used in the browsing language. An action word was combined with a target word and mnemonically 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.

The action **speak** requires some explanation. There is a need to be able to speak the contents of complex objects 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 a fraction was nested within a fraction and the focus was upon that nested fraction, then the **current fraction** command could legitimately be applied to both. So another action, **speak** was used to utter the

Action	Target
Speak	Expression
Current	Term
Next	Item
Previous	Quantity
Beginning	Super
End	Fraction
Into	Numerator
Out-of	Denominator
	Level

Table 4.1: The set of action and target words used to generate commands for the final evaluation of the command language and browsing functions.

contents of a complex object while the focus was ‘on’ that object rather than within that object.

The meanings of the target objects are self-explanatory. If the labels are in accordance with the user’s knowledge of algebra structures then few problems should arise. However some of the labels need some explanation. The target **item** replaced the notion of a character. As explained above the smallest unit of speech could be indeed a character, e.g., x, but it could also be $-x^2$ or ‘a fraction’, to which the label ‘character’ does not fit. The target **level** refers to the current scope or level of nesting within an expression. The top or base-level encompasses the whole expression. Scope or level of nesting is a common concept in mathematics and should be explained to potential users in that context. The **item** and **level** are generic targets that may be thought of as ‘thing’.

From a small number of action and target words a very large number of commands can be generated. Table B.1 shows the valid commands for the command language. It is apparent that a relatively small set of actions and targets can be used to generate the large number of commands required to cover the complex needs of browsing algebra. The table shows that very few combinations did not generate valid commands. So even if a command was inappropriate some action would take place.

Some commands were context sensitive. The action **into** only works if the focus of attention is on an object representing a complex structure. So, **into fraction** is a valid command that is only appropriate in certain contexts.

This design gave consistent generation of a large set of commands. All browsing commands were two letter sequences, generated from a small list of actions and targets. 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 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 very learnable.

One element of inconsistency was present in the command set. The Mathtalk program presents a list of expressions, numbered from 1 to n. The list is circular, so a reader can move from one extreme of the list to the other in a single move, rather than issuing a series of commands. So the browsing task can be split into a small set of inter-expression moves and a much larger set of intra-expression moves.

To be consistent inter-expression the command **current expression** was used to give the expression's number in the list, as did **next expression** and **previous expression**. Being a complex object it might be expected that **current expression** spoke the whole expression, like **current fraction** would when the focus was inside a fraction. Since this command combination was used elsewhere, **speak expression** was used to utter the whole expression. **Speak** was otherwise used only to speak the contents of complex objects when outside that object, not within a complex object such as the whole expression. This inconsistency could be resolved by introducing another command, but the principle of parsimony was used and the inconsistency retained.

The command set shown in Table B.1 contains a large number of possible commands. Some of the combinations only exists for the sake of completeness; for example, **end term**. It is unlikely that a user would ever need to use such functions. It should also not be necessary for users to learn every single command. A core of commands would provide for most situations. Learning a core of basic commands, and as a consequence the actions and targets, would enable a reader to generate new commands spontaneously. So a small amount of learning of words and basic rules would enable a reader to deduce how to make new commands for new reading needs.

It was hoped that the users would be able to build up a series of small moves available from the command language into higher-level tactics or strategies. For instance with Expression 4.1 issuing the commands **nf, if, nq** and **sq** would move the user from the beginning of the expression to the radicand inside the numerator of the fraction. How quickly readers could develop such strategies would be an indication of how useful the command language was for reading.

The following scenario shows how the browsing language could be used to move through the expression

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

Current expression → '`expression one`'.

Speak expression → `'three times x plus four equals seven'`.

Current level → `'3 times a quantity equals seven'`.

Next quantity → `'a quantity'`.

Speak quantity → '`x plus 4`'.

Into quantity → `'the quantity x'`.

Next term → `‘plus four’`.

Current level → `‘the quantity x plus four’`.

End expression → `‘seven’`.

The command **current level** emerges as interesting. This command utters all simple objects in full, but reduces complex objects to their labels. This gives a precis of the level. It was hoped that users would be able to use this command to give an overview of an expression, shortening it if it contained complex objects. So Expression 4.1 would be uttered as ‘x equals a fraction’. Such a glance could be useful in planning how to use the browsing functions and affords another level of control for the user. The scenario also shows how the hiding of objects could be used to view the whole of a complex object without having to enter that object.

A simple extension was added to the actions **next** and **previous**. These actions could be ‘multiplied’ so that more than one command of the same type could be invoked at once. If users had to move to an expression remote in the list, several **next expression** or **previous** expressions had to be issued. The functionality was extended so that an integer could be prefixed to **next** and **previous** actions when applied to **expression**, **term** or **item**. These targets were those thought most likely to benefit from the application of multiple moves, probably being the most frequently used objects in a list of expressions.

4.3.2 The Default Browsing Style

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 controlled manner without having to think of or issue any other commands.

Two forms of this default browsing were originally designed: A term-by-term method and an unfolding style. The term-by-term method could be used to make the presentation move forward by one term, speak that term, leaving the pointer on the last object spoken, ready to move onto the first object of the next term. It was envisaged that this method would be very suitable for syntactically simple expressions with many terms. For example, the expression:

$$3x^4 + 7x^3 - 8x^2 + 9x + 1 = 0$$

would be presented in the following manner by this component (each item in the list represents the output from each invocation of the style):

1. three x super four;
2. plus seven x super three;
3. minus eight x super two;
4. plus nine x;
5. plus one;
6. equals zero.

Each time the user invokes the term-by-term function, a new term is spoken. The term was chosen as the basic unit of information or unit of control. The prosodic investigation suggested that the term was the basic unit in spoken algebra. This is also consistent with the term as the first product of parsing.

Some objects within a term may be complex. As described above, these complex objects were hidden and represented only by place-holders. A different presentation style was developed for these objects. Complex expressions could be unfolded. Invoking the unfolding style on a complex expression would: speak all simple objects, speak the label for a complex object, then stop. The next invocation would enter the object, announce its type and utter the contents of that object, leaving the pointer at the end of the complex object or on another complex object. An unfolding of the expression:

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

would be:

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

Each item in the list represents one invocation of the unfolding. Three different prefixes were used for the complex object labels. These were ‘the’, ‘a’ and ‘that’. The indefinite ‘a’ was used when focus moved to a new complex object with an unknown nature, that is, indefinite or information whose nature is as yet unknown. The definite label ‘the’ was used for information that is about to be given in detail. For example, that it is a fraction is known and the detail is to be specified. The label ‘that’ was used for contrast when a complex object had been unfolded and focus moved onto new information e.g., ‘that fraction plus 3x super five’.

This style was designed to emphasise the overall structure of an expression. By stopping on the fraction and simply announcing the next chunk to be ‘a fraction’ the whole structure is made explicit.

Control over the information flow was given to the user, so not too much information was ever spoken at any one time.

Informal evaluation of the two default browsing styles showed them to be useful, but the split created problems for some readers. The term-by-term and unfolding strategies were designed for different types of expression. The first was for long expressions with many simple terms. The second was for expressions containing complex objects. Few expressions are all of one structural type and having to swap between styles interfered with the reading process. Also, the reader had to judge which browsing style to use, rather than simply moving through the expression chunk by chunk.

The two styles were combined into a single default reading style. Instead of uttering all the contents of a complex object at once, they were unfolded term-by-term. The unfolding of the same expression as above now proceeds as follows:

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.

This was designed to make sure all objects were revealed in a consistent manner. It also avoided the need for two different default styles, that would necessitate a more complex browsing language or the use of a moded browsing interaction. Moded interactions are commonly thought to be a bad design feature (Tessler 1981). As a consequence Mathtalk was designed to be as modeless as possible. All browsing moves would be available at all points within and between an expression. As the reading process was of primary importance a more complex browsing interaction would increase the mental overheads felt by the reader and interfere with the reading process.

The space bar was used as the key-stroke to invoke the default browsing strategy. This was a large, easily accessed key with no other meanings attached. This should make the default style more attractive to the user.

4.3.3 Feedback during Browsing

Just as lexical cues inserted to disambiguate grouping can disrupt reading, so could superfluous lexical feedback during browsing. Thus, the command given will not be confirmed lexically, but by the spoken algebra itself. Any information that is necessary can be given via non-speech audio. Giving some signal by non-speech sound may avoid the suffix effect that can arise from using lexical feedback (Baddeley 1986).

Large amounts of speech were never automatically spoken. When moving between expressions only the expression's number was spoken, not any part of the expression. The aim was to maintain orientation within the list and give the user full control over speaking of the expression.

The term was taken as the default form of output. When moving between objects of a granularity larger than an item only a term or a hidden object label would be spoken. This was designed to reduce the amount of speech given at any one time.

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

1. First key-stroke error. A mnemonic for an action not appearing in the language was used.
2. Second key-stroke 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. Non-speech was used in order to give quick meaningful messages that would intrude into the reading process as little as possible. First and second key-stroke errors can be readily indicated with a single and double tone respectively. By extension a three tone message was used for the third error.

A system of non-speech messages was used to indicate the beginning and end of levels or complex objects within the expression. These will be referred to as *terminus* sounds. Descending and ascending C major chords were used to indicate the end and beginning of levels respectively.

A start was indicated by a rising sound and the end by a falling sound, to be consistent with the use of pitch within the algebraic utterance. A rising pitch was used to indicate the start of new information and descending pitch to indicate the end of the utterance. So, a rising pitch was used to indicate the onset of a new environment or structure and a falling sound to indicate the end of that environment.

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 or speech.

In a certain situation the inappropriate-target error message was not used. If the reader was at the end of a level in the expression and issued a ‘next’ command, (i.e., progress past the end of the level), an ‘end sound’ was issued rather than an error sound. In this case the orientation information was thought to be more important than the error information. The inappropriateness of the command used is encapsulated within the orientation message. Orientation sounds were given when **previous** and **next** actions were used with **item** and **term**. The same goes for the default browsing. When **next** and **previous** were used with complex objects inappropriate command messages were given, because the target does not exist, rather than the focus being at the end of the level.

4.4 Evaluation of the Browsing Component

The co-operative evaluation method was used to assess the usability of the browsing functions and command language. Co-operative evaluation (Monk, Wright, Haber, and Davenport 1993) was developed as a cut-price, informal method for evaluation. The rapid, informal nature of the evaluation allows several evaluations to take place without the time and cost overheads of lengthier forms of evaluation, such as those seen in Chapters 3 and 5. Co-operative evaluation relies on using a smaller number of participants to capture general or major usability problems of the user interface.

A set of tasks are designed for the system being evaluated and the user asked to perform these tasks. During this process the user is asked to ‘think aloud’, to say what he or she is doing and why. The participant is also encouraged to interact with the experimenter. This gives a rich source of information about the usability of the system, that cannot be captured by simple quantitative measures alone. This links back to the use of subjective mental workload assessment in the evaluation of user interface designs that was used in Chapter 3.

The method relies on the ability of the experimenter to judge and act upon the findings of the experiment, rather than perform objective statistical measures, though these are also useful on the data. The information about each round of evaluation then feeds back into the design of the next stage of the design, being either an incremental development of the existing design or a complete redesign.

One of the major aims of co-operative evaluation is to assist iterative development of the user interface. This chapter reports only one such cycle of evaluation in detail. However an iterative design methodology was used during the development of the browsing component of the Mathtalk program. Each major round of design involved informal evaluation and redesign.

The original browsing language used the cursor movement keys commonly found in PC DOS based word-processor packages as the basis for the language. The left and right cursor keys moved character by character. Holding the control key and pressing the left and right cursor moved from term to term, the counterpart of moving between words in a document. The `home` and `end` keys moved to the end of an expression. The up and down cursor keys moved between expressions and modifications of these keys with the alternate and control keys moved into and out of complex objects.

Co-operative evaluation rapidly demonstrated that such a language was not rich enough to cope with the requirements for Mathtalk's browsing component. Apart from the basic character and term movements, the commands had no real-world counterparts in word-processors and users consequently found them hard to remember. One major gap was the lack of a **current** command that could speak the current object selected. As a consequence of these early evaluations this command language was replaced with the one described above.

The new language went through several cycles of evaluation and design. The default browsing styles were combined into one, as described above. Browsing modes also existed for the unconstrained and default browsing styles and these were removed as a result of users' difficulties. The terminus sounds described above also replace lexical cues for the indication of termini of levels. The command words used also evolved. Details of some of the stages in the development of the browsing can be seen in Stevens and Edwards (1993), Edwards and Stevens (1993), and Stevens and Edwards (1994a).

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. The reading has to remain of highest priority. So the language has to be as easy to use as possible, in terms of delivery, learnability as well as enabling what the user wants to be spoken to be spoken. The tasks described below attempted to explore this basic usability of the browsing language.

Showing that increasing control was a good design decision was a task more relevant to the final evaluation of the integrated Mathtalk program. This follows the scheme of evaluating each component to show that it performed the task for which it was designed; that the sum of the components improved reading, while suggested by the separate evaluations, can only really be demonstrated with the whole system. However, this evaluation, as it essentially involves a series of reading tasks, was taken as a pilot for the final evaluation of the whole Mathtalk program.

The objectives of this evaluation can be summarised as follows:

1. Does the control component contain all the browsing functions to perform ecologically valid tasks?

2. Are all the words in the command language appropriate for the tasks and known by the users?

3. Does the language cover all the moves readers wish to undertake?

4. The accuracy component of control is inherent to the browsing, but can commands be issued to give the speed component of control without an undue level of errors?

5. Is the command language learnable; do users need to learn all the moves or can they generate new commands from knowledge of the command words?

6. Can users build higher level tactics from the low level moves available through the command language?

4.4.1 Design

An adapted co-operative methodology was used for this evaluation. This evaluation attempted to be ecologically valid. As well as simply giving navigation tasks, the participants were asked to complete some mathematical tasks. As Mathtalk only allows reading of the notation, simple substitution and evaluation tasks were used. As sighted participants were used, pencil and paper were allowed for writing down intermediate values during such tasks.

The minor adaptations to the method were to include some objective measures: error rate in issuing commands; speed of issuing commands and completion of the mathematical tasks. A larger number of participants were used to capture a larger number of errors. The previous evaluations had removed many of the more gross errors, and the increased number of participants would reveal the finer grained usability problems. Monk et al. suggest that the greater number of usability problems are revealed with 1–5 participants or evaluators, even when there is a low probability of a problem surfacing. For this reason five participants were chosen for this final iteration of evaluation of the browsing component.

Materials

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 the UK GCSE examinations and A-level mathematics courses (see, for example Bostock and Chandler 1981). Some of the expressions found to cause problems in other experiments were re-used in this

Number	Stimuli
1	$y = 7x + 3$
2	$2x^4 + 8x^3 + 7x^2 + 2x + 5 = y$
3	$3(x+6) - 4(x-3)^3 = y$
4	$y = \frac{9x}{(x+4)(x+9)}$
5	$x^n + 1 = y$
6	$x^{n+1} = y$
7	$4(x + 3(2x + 9)) = 7$
8	$y = \frac{1}{2}(x+9)^2 - 5$
9	$a^{2+n} + b^2 = c^2$
10	$4x + 3y = z$

Table 4.2: The stimuli expressions for the evaluation of the command language.

experiment. These were the nested sub-expression (Expression 7), the expression with a fraction followed by a quantity (Expression 8) and the partial fraction (Expression 4). The stimuli are shown in Table 4.2.

In the list which follows, questions were used to structure the evaluation. The tasks encapsulated both questions and training. The tasks started with high-level concepts of moving between expressions, finding the expression number and speaking the expression. Then default browsing and the **current level** glance were introduced, which naturally entailed moving between the extremes of expressions. Then finer level moves in structurally simple expressions were taught.

Having finished a core set of moves, the tasks moved onto dealing with complex objects in an expression. For example, moving **into** and **out-of** complex object; revealing the contents of such objects from outside and within the structure. The tasks were rounded off by some simple arithmetic tasks of substitution and evaluation. A final task of moving through the list and reviewing the expressions was used as a general review of use of simple browsing moves and how easy it was for users to extract information from the display.

The tasks were largely navigation and orientation based. This reflected the aim of giving control over information flow. The browsing was designed to replace the selection of information from the external memory. Kwasnik (1992) suggests transition and orientation are amongst the more important components of browsing and the tasks were designed to explore this aspect of the design. In general, a participant was asked to move to a certain part of an expression, and describe layout. These tasks were reasonably ecologically valid as such moves would be needed during the reading, writing and manipulation of algebra in a mathematical task.

The tasks were constructed so that the command was embedded in the utterance. As the natural spoken form was part of the design, avoiding its use so as not to be seen to be ‘helping’ the user, would have been to ignore a large part of the design.

The easy spoken form was used to facilitate training, in that the movement could be easily described. The command form was designed for exactly this purpose, to avoid such prompts would be counterproductive and yield contrived speech. Later questions necessitated using commands used earlier, as well as the one embedded in the question. This was used to probe the learnability of the language. Other questions directly asked participants to predict how they would invoke a previously unused command. The later, larger scale tasks were used to investigate if users could build up commands to larger tactics and strategies and spontaneously generate commands for the tasks from their knowledge of the action and target words. The tasks were set up so that maximum advantage could be made of the consistency in form of all the commands. Having once learnt the action and target words, a user should be able to generate all the possible commands. The tasks are shown below. The *emphasised* text describes the purposes of some of the tasks.

1. What is the current expression number?
2. Speak the whole expression.
3. Move to expression two.
4. Move to expression three and speak that expression.
5. Get an overview with **current level**. *This task introduces the use of hidden objects.*
6. Move back three expressions and check the number. *This task allowed the introduction of multiple commands and showed that the list of expressions was circular.*
7. Move to expression two and use the default browsing style to move through the expression. *Introduced space bar as a simple method for unfolding an expression.*
8. When the current term contains x^2, stop and state at which item the speech cursor is pointing. *This task examines whether the pointer was where the participant expected it to be placed.*
9. Continue to the end of the expression. *Do the users realise when the end of the expression has been found; also allows the* **beginning** *action to be introduced.*
10. Move to expression three and use the default strategy. *Introduces unfolding of complex objects.*
11. How many quantities were there in Expression three? *Tests overall knowledge of the expression's structure.*
12. Move to the beginning of Expression one, read the current term, then move to the next term.
13. How do you move to the previous term?

14. How do you read individual items? *These tasks test some fine grained moves and generation of commands.*

15. Read and describe expression four.

16. Move back to the beginning of expression four.

17. What are the next two items?

18. Speak the fraction. *Tests the revealing of a complex object's contents from outside that object.*

19. What is the current item? *Tests concept of a hidden object.*

20. Move to the end of the expression. Where are you? *Tests knowledge of expression's structure and users' orientation.*

21. Move out of the quantity, into the denominator.

22. Speak the current fraction. *Revealing the complex object's contents from inside the hidden object.*

23. Speak the current denominator.

24. What are the differences between expressions five and six? *Can the users compare two expressions? Shows the utility of hidden objects.*

25. Check what the first two items in these expressions are.

26. Speak the superscript in expression six.

27. Move to expression eight.

28. Explore question eight (without speaking it as a whole). Describe the structure of the expression. *Tests knowledge of browsing commands.*

29. Move to the end of the expression, move to the previous fraction and move into the denominator. *Builds up sequences of commands.*

30. Speak the current numerator.

31. Move out of the fraction and into the quantity.

32. Move back to expression two using multiple commands.

33. Where are you in the expression? *Use of terminus sounds.*

34. Move to the term with x^2 in expression two.

35. Move to expression seven. What is complex about this expression?

36. Move to the deepest part of this expression. What is the last item here? *Tests concepts of moving into and out-of complex objects.*

37. How many superscripts are there in expression nine? Speak each of them separately.

38. In expression ten, if x=2 and y =3 what does z equal?

39. Substitute x=2 into expression one.

40. Substitute x=2 into expression two.

41. Which are the most complex expressions in the list?

42. Come out of Mathtalk.

The expressions were presented to the participants using the Mathtalk program. This was a DOS program, implemented in the C language on an IBM compatible computer. The Mathtalk program takes expressions written in a sub-set of the LaTeX typesetting language (Lamport 1985) and transforms it into a data-structure suitable for speaking and browsing an expression. The Rules for speaking algebra with prosodic cues had been implemented so that they could be added dynamically to any expression of the range described in Chapter 3. All the rules for browsing algebra and hiding objects, all with appropriate feedback, were also implemented within the Mathtalk program. The speech was presented with a Multi-voice synthesiser. There was no visual output on the screen that would give any clue to the sighted participants as to what processes were taking place during the evaluation.

Procedure

Five sighted participants were used in this evaluation. All were familiar and confident, by their own judgement, with basic algebra notation. Four of the participants were experienced computer users. The fifth was a novice computer user. This was relevant with respect to keyboard skills. Whilst the novice user knew a typewriter keyboard he/she was unused to a computer keyboard. This participant was used as a severe test of the usability of a command language whose utility was so tightly bound to competence with the keyboard.

Sighted participants were used for the same reasons described in Chapter 3. A further practical reason was that the mathematical tasks necessitated some use of paper and pencil as an external memory to store intermediate values of the evaluation tasks. Use of tape, braille or a second computer by visually disabled users might have interfered too much with the tasks. There was one problem to be taken into account with this decision. During reading, sighted readers are used to having the focus of attention being what they are actually looking at in any given moment. With Mathtalk's reading style, the focus of attention is what the system is currently pointing towards. Presumably blind computer users or readers are more used to this situation. This factor had to be taken into account in the discussion of the results.

Initially the participants were given an explanation of the Mathtalk program, the command language and the style of the evaluation. It was emphasised that it was the software, not the participants' mathematical ability, that was under examination.

The Mathtalk program was described as 'presenting a list of expressions and allowing the reader to move around between and within expressions'. The command language was described as forming a set of spoken instructions and that the commands could be formed by extracting an action word and a target word from the spoken task. These two words were to be mnemonically mapped to the keyboard.

The first task of the set shown above was used to demonstrate the formation of commands. Initially the command words were stressed in the experimenter's speech. This emphasis was reduced as the experimenter judged the participant to be used to the style of commands. Otherwise help was only given when the participant asked for it. The participant was also asked if he or she wished for help or prompted for information when prolonged inactivity occurred. Given the novice state of the users, it would not be expected for all the instructions to be remembered by the users. Where participants ask for help may be informative about usability problems with the interface.

In accordance with the co-operative evaluation method, participants were encouraged to describe what they were doing in performing the tasks and why particular actions were chosen. The need to gather such data was balanced with the need not to interfere too much with the participant's performance. Participants were also encouraged to ask the experimenter questions about the system and advice on how to perform tasks. As the participants were novices with the Mathtalk program, expecting them to remember all the commands and facilities of such an obviously complex system in a relatively short session would be unrealistic. Questions asked would reveal that participants remembered that a certain move was possible or that a certain move was needed and the command supplied. That questions were asked does not diminish the learnability of the command language.

4.4.2 Results and Discussion

This evaluation was successful in demonstrating the general usability of the browsing system. 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 flaws in the design. First some general, positive, observations of the usability of the command language will be presented. Then some specific problems and design solutions will be discussed.

As the command language was based upon a natural spoken form, the tasks the participants were asked to perform often included the command name. Part of the training procedure was to emphasise this in the experimenter's speech. All participants successfully picked up the commands, usually uttering the two letter command derived from the experimenter's speech, though this habit decreased during the experiment. As the commands matched the tasks, simply extracting the command, taking the mnemonic mapping would complete the task successfully. Some exceptions to this general trend are discussed later.

The training program used in the evaluation delivered the concepts in a suitable order. No tasks required knowledge that was not given with the current task. An exception was the concept of the *item* which was, in fact, introduced too early. It was used to ask where the user thought the speech pointer was located; the introduction of the **item** should have waited until the **item** target was used, and the orientation question asked subsequent to that point. The amended training routine was adopted for the final evaluation of the Mathtalk program.

The form of the commands were remembered between tasks. The following example is typical of many of the later, more complex tasks.

Participant C4, Expression nine task 37:

1. Okay, I'm on eight, so next expression …
2. **n e** `expression eight.`
3. Oh, I'm on Expression seven,
4. **n e** `expression nine`
5. Current level.
6. **cl** `a with a superscript plus b super two equals c super two.`
7. Three.
8. *Speak each of the superscripts.*
9. Er, I'm still on the first item? So s s to read the first one. **ss** `two plus n.`
10. ns for the next one?
11. **ns** `b super two`,
12. ns again,

13. **ns** `c super two`.

The **speak superscript**, **next expression** and **current level** commands were retained. The **next superscript** command used by C4, was not taught, but generated by the participant from knowledge of the command words. For the simple tasks the command was always apparent in the speech, however, for more complex tasks, more commands had to be remembered.

Participant C2, Expression eight task 28:

1. **cl** `y = a fraction times a quantity super two minus five`.
2. **sf** `(inappropriate command sound)`.
3. *That was an error message that meant*
4. I wasn't on the fraction.
5. **nf** `a fraction`.
6. **sf** `one over two`.
7. **ee** `minus five`.
8. Oh **pt** `a fraction times a quantity super two`.
9. *What are you trying to do?*
10. Find what the quantity is.
11. **sq** `(inappropriate command sound)`.
12. …*What were you saying there?*
13. Speak quantity.
14. *You've got to be on the quantity. If you go to the next…*
15. **ni** `a quantity`.
16. **sq** `x plus nine`.

This example shows a use of a wide range of commands without prompting. The natural spoken form of the command language probably makes it easier to remember the command set. Simply remembering the words enables commands to be generated. Again commands like **next quantity** and **next fraction** were often generated by participants without them being taught explicitly.

The underlying concepts behind the language were also readily adopted. For example current, next and previous in particular seemed to be intuitive and readily adopted. Whilst mistakes were made with **speak quantity** by participant C2 above, C2 had knowledge of having to be on the complex object and chose the correct commands to achieve her goal. The use of complex objects is discussed more below.

Participant	Total	CL	SE	Default
C1	179	0.20	0.03	0.17
C2	168	0.12	0.02	0.25
C3	192	0.16	0.05	0.24
C4	202	0.14	0.02	0.29
C5	248	0.10	0.07	0.26
Overall	989	0.14	0.04	0.24

Table 4.3: Command usage as a proportion 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 proportion for each command.

The discussion of error rates below reveals how few commands were mis-extracted from the experimenter's questions. This suggests that the command language and associated browsing functions matched the tasks. Given that the tasks were reasonably ecologically valid, i.e., they were tasks that would actually be undertaken during reading or 'doing' algebra, the structure based nature of the control component was a correct design choice.

The labels used for actions and targets were appropriate. Users readily extended the language from what they were taught to new combinations within the command words. This was particularly true of using *next* and *previous* to move rapidly to complex items. Also once *into* was introduced, for instance, it was readily applied to all complex items. The use of **next superscript** above was a typical example of this generation of commands. Similarly, when browsing Expression eight above, C2 used **next fraction** without being explicitly taught the command. The **speak**, **into** and **out-of** actions were similarly applied to many targets.

The default browsing strategy was widely used. Table 4.3 shows that a fifth of the total commands issued by all participants were for the default browsing style. The range varies from 0.16 up to 0.3.

When a new expression was to be read, the default browsing strategy was one of the main methods used to examine the expression. An example of the use of the default browsing strategy can be seen below.

Despite the short period of the evaluation users developed strategies for reading expressions. Three commands were prominent in such cases: **current level**; **beginning expression** and the default strategy. Using **current level** to speak the base-level of an expression, utilising the effect of hidden objects reducing the amount of speech and giving an overview of the expression, was taken advantage of by the participants (see the examples above). The **current level** command acted like a glance at the overall structure of the expression. This glance was an emergent property of the hidden objects.

After using **current level** to obtain a first view of an expression participants often used the default strategy to explore the expression. During the reading of Expression 4, C3, found the whole expression too much to comprehend. He then used the **current level** followed by the unfolding provided by the default browsing to build up a full representation of the expression. This sequence, without the use of speaking the whole expression, was adopted for the first approach to exploring most expressions.

Beginning expression became part of initial reading strategy because the Mathtalk program adopts the previous position when returning to an expression. If this return was for an unconnected task, the reader may have become disorientated. To avoid confusion users started to issue the **beginning expression** command before any others. For example,

Participant C2, Expression 3 task 41:

1. **ne** `expression four`.
2. **cl** `denominator a quantity times a quantity`.
3. okay?
4. *Remember it remembers where you were in the expression.*
5. **be** `y (beginning sound)`.
6. **cl** `y equals a fraction`.

More local strategies could also be seen. For example when asked to read each superscript in Expression nine, participant C2 issued a series of **next superscript** followed by **speak superscript** commands (see the earlier example). Other examples show the low level moves put together to achieve sub-goals of the overall task. The development of such strategies or tactics is an important part of the goals of the control system. Only low-level moves are provided and the user is left to make his or her own tactics by combining these fine grained moves. 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**. Table 4.3 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 emphasises 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. For example, simply by moving between Expressions five and six, and using the **current level** participant C2 was able to immediately see that one contained a complex superscript and the other a simple one.

The substitution and evaluation tasks were successfully accomplished by all users. Unfortunately the tasks were not well designed in that Expressions one and ten were too short to force use of browsing functions to accomplish the tasks. All users simply spoke the whole expression and calculated internally. Even when the expression was not remembered in one step, a full utterance was used instead of any browsing functions. This may indicate there is a threshold under which expressions may be held internally and the overheads of thinking about which browsing functions to use, as well as performing arithmetic are not worth the investment. However, with Expression two all users browsed the expression, calculating an answer term-by-term. Four participants used the default browsing strategy and one used **next term**. Two of the participants moved backwards and forwards through the expression to check answers.

General Feedback

The non-echoing of commands caused no problems for the participants. At no point did users request confirmation of an action just executed. The design principle that only information pertinent to the reading task should be presented was successful. For example, the command **next fraction** either gave the output ‘a fraction’ or the inappropriate command warning. For moves that did not speak labels, (e.g., **next term**) the spoken algebra seemed to provide sufficient feedback. Perhaps, the memorable, unambiguous form of the commands may have helped by making the user confident of actions executed.

One aspect of the feedback during use of the default reading strategy was noted as tiresome by four of the five participants. If the default strategy was used, then interrupted by use of some other moves, then re-adopted then the current term was repeated before the strategy took the reader onto the next chunk of information. This was due to the technical difficulty of the system being able to record what was last spoken. The users expected the strategy to take them onto the next term whenever it was pressed. This usability problem should be removed from the system.

Task 13 revealed some problems with how operators were associated with terms. When moving forward through a term the operator to the left of the term was always spoken (for the reasons given in Section 4.2.2 above). When moving backwards, with the **previous** action, the operator to the right was spoken. This confused three of the the participants, who expected the left-hand operator to be spoken. For example:

Participant C3, Expression 1 task 12

1. *How do you think you move to the previous term?*
2. p t.
3. **pt** `equals y (beginning sound)`.
4. *notice you’re at the beginning. I want you to …*

Participant	**Commands**	**Errors**	**% errors**
C1	179	7	3.91
C2	168	4	2.38
C3	191	7	3.66
C4	202	4	1.98
C5	248	8	3.23
Total	978	32	3.28

Table 4.4: Table of number of commands issued, the number of errors and the percentage of errors for each participant and totals.

5. When I go back to the beginning why does it say equals?
6. *speak the whole expression*
7. **se** `y equals seven x plus four`.
8. *you were going backwards.*
9. It sounded like the expression was equals y something something something.

The output was redesigned so that a simple rule of only speaking the operator to the left of the term will be used. The fact that the operator to the left of a term was not spoken when using **current term** was not commented upon, so the rule was not implemented for this command.

No users explicitly requested the need for a mute function to terminate spoken output. On one occasion participant C3 spoke the whole of Expression two and showed some frustration with the long 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. This shows the benefit of designing for control of information flow from the beginning of development.

Most users adopted the command **current level** as the first attempt to display an expression. As described above, this command potentially reduces the amount of speech produced. The reduction of speech to a minimum, and not automatically speaking any of the expression, seemed to reduce irritation and frustration in the participants.

Command Errors

Very few errors were made by participants when issuing commands. The overall error rate was 3.28%. For the large number of tasks and associated commands, such a low error rate for novice users was very encouraging. These errors were those commands that generated error sounds. That is, first or second key-stroke errors, or inappropriate commands. Those commands that did not accomplish the task do not appear in the Table 4.4.

Participant C5 had slightly more complex results than the other participants. C5 was a novice computer user and made the frequent mistake of holding down the keys of the keyboard such that they repeated. Observation of the log file indicated that the intended command was correct. These errors were not counted in the individual or overall error rate. These repeats meant 28 extra commands and 28 errors were generated extra to those shown above.

The command errors fall into distinct categories that demonstrate some of the main areas of usability problems:

- The participants so readily extracted commands from the experimenter's speech, that some incorrect commands were derived. Three action words account for these difficulties: speak, move and first. Task 25, 'move to the first item in these expressions' led to participants using the command **fi** to move to the **first item** instead of using **beginning expression**. This demonstrates a problem with having the commands so readily fall into a natural form. These mis-extractions could also be due to many of the tasks being driven by the experimenter. Such errors will decrease with growing familiarity with the command set.

  In a similar manner, participant C1 mis-extracted the action **move** from the task 'move out of the quantity', typed **mo…** for **move out-of**, the **move** caused an error, **out-of** was accepted as an action, **move** was re-entered following the mistake, resulting in the command **out-of Mathtalk**, which then terminated the session.

- Many errors occurred because the participant offered a command that was suitable to accomplish the task, but invoked it in the wrong context. This was primarily true of entering or speaking the contents of complex objects. The system focus had to be on the hidden object to perform these actions. This was especially true of task 31on Expression eight. The participant had to leave the fraction and move into the denominator. All but two of the participants issued the command sequence **out-of fraction** and **into quantity**, without moving the focus to the quantity using **next item**. This may be unfamiliarity with the system or that the participant's focus had moved to the sub-expression, but not the system's. This type of error may also be a consequence of using sighted participants. Visually disabled computer users, especially those using speech screenreaders, will be more familiar with the concept of moving the screenreader's focus to the portion of the display to be read. Despite the difficulties with the hidden object, they generated relatively few errors and the advantages of controlling speech output outweigh the problems with adding extra commands. Removing the hidden objects would not decrease the error described above.

- Many errors were a result of not distinguishing between commands used inside a complex object and the same target being acted upon when on the hidden object representing that target. For example, in Expression four, ‘speak fraction’ would be used inside the fraction when ‘current fraction’ was the correct command.

One further possibility for these errors would be the use of the action word **speak** in a non-action context. The phrase ‘speak current numerator’ could be mis-extracted as **sn** instead of **cn**. Removing the word **speak** from the action list would remove this cause and leave only the design of the hidden objects themselves. This is discussed further below.

- Sequence errors were generated by reinterpretation of a second key-stroke as an action, after an erroneous action was issued. An example was given above with the ‘first item’ command. **F** was rejected as a first key-stroke, the **i** intended as a second key-stroke was then interpreted as a first key-stroke, i.e., **into**. The user would be unaware of this status and continue to issue command pairs, which would all be out of step and give a sequence of errors. The sequence of errors meant a series of error sounds. This ‘cascade’ of sounds meant each individual error sound was indistinguishable from another, so rendering these sounds unusable.

  A mute function was added to help avoid superfluous speech and recover from command errors. Pressing the escape key would mute the speech and consume any command key-strokes awaiting processing. Forcing the user to recover from an error state would also prevent entry of any further key-strokes which could be mis-interpreted. This would ensure only single key-stroke errors would be given, thus making the error sounds more useful.

Gaps in the Browsing Language

Detailed use of the browsing language during the tasks revealed some gaps in the coverage by browsing functions. A good example was the need for the command *speak item* to be used as a generic command to speak complex objects.

In the evaluated design, only specific targets could be used with the **speak** action. Similarly *into item* and *out-of level* could be used as generic commands to move into and out of complex items. The use of **item** and **level** in this manner allows many of the frequently used moves to be accomplished with fewer targets. These two generic targets could be used to reduce the amount of learning a user has to undertake. A complete matrix of valid commands can be seen in Table B.2 in Appendix B.

Timing Information

The timing information was not informative. The data recorded did not reveal how quickly users generated commands from the task utterance or during self-motivated browsing. The time-stamped

Participant	Key-strokes	Mean	Variance
C1	142	0.29	0.20
C2	124	0.44	0.08
C3	144	0.46	2.63
C4	141	0.43	0.31
C5	189	0.77	5.20

Table 4.5: The mean and variance of inter-key-stroke time for two letter commands.

key-strokes do indicate the length of time between key-strokes. The mean and variance for the inter-key-stroke time for the double key-strokes are given in Table 4.5.

These times reveal that the commands themselves were quickly issued. The vast majority of execution times were well below one second. For each participant a few execution times were of multiple seconds, for C5 the maximum was 23 seconds. This indicates, that while most commands were given very quickly, a few required much thought or even prompting. This quick time suggests the command language could be used to give the speed characteristic necessary for active control of information flow.

Hidden Objects

Despite the utility of hidden objects in controlling information flow demonstrated by the emergent glance and the reduction in speech flow, their use did cause some problems. The concept of being ‘on’ or ‘inside’ a complex object caused some problems. The hiding of objects necessitated an extra action within the language. This was the distinction between **speaking** an action from outside that object and speaking the **current** contents of that object.

Speak was part of many of the task phrases as well as an action word. Choosing another action such as **show** to cause the contents of a complex item to be spoken without having to move into that item may remove a source of confusion between revealing contents of complex objects from without and within. Other possible action words such as ‘utter’ and ‘reveal’ are obscure. **Show** maybe thought of as a ‘visual’ word, but this should not cause problems. Landau (1988) suggests that blind children are aware of the meanings and distinctions between such words as ‘show’ and ‘look’ that are common in our visually based everyday language. The use of **show** may reduce some of these errors by avoiding confusing re-use of the word **speak** and having a tighter semantic connection with the results of the command.

The inconsistency between the use of the actions **show** and **current** may have made the ‘without’ and ‘within’ moves more difficult to teach and confusing to use. **Speak expression** was used to utter the whole expression, i.e., the action **speak** was used to utter the whole expression whilst ‘within’ the expression. Consistency would dictate that **current** be used instead. **Current expression** was used to utter the expression’s number in the list to avoid introducing another action word, i.e., the principle of parsimony. As intra-expression browsing was the focus of the tasks, rather than inter-expression movement, this inconsistency may have made the teaching of the system more difficult than necessary. The action **current** was made consistent intra-expression by introducing a new action word **which** that could be combined with **expression** to utter the expression’s number in the list.

Despite appreciating the hiding of complex items, there were some complaints. For example, when **speak fraction** is used, complex things within the fraction are still hidden and they cannot be revealed without going inside that object.

Small complex objects could be spoken in full. A configuration could be used to set the numbers of items in a complex object for it to be hidden. Another option would be a *keep simple* command that would cause the whole item to be regarded as simple. This would give the reader greater flexibility as what would be hidden during a **current** or **show** command.

The scope of the **into** and **out-of** command within complex objects caused some frustration. For example in Expression four, when inside the quantity in the denominator **out-ofi** fraction would be inappropriate, despite obviously still being inside the fraction. This was designed to avoid ambiguity and force maintenance of orientation. For example, if a reader is inside a nested sub-expression (e.g., expression seven) wanting to come out of the quantity may mean the inner sub-expression or the outer one. The function of the **out-of** action was altered so that it took the narrowest possible scope for the given target. The use of **current** with a complex target caused similar problems. The functionality was changed so that a complex target would be accepted if the current location was anywhere within that object.

Most of the few command errors were accounted for by difficulties with the hidden objects. The utility of this design, as demonstrated by reduction in amount of speech and being able to treat them as single items to facilitate quick movement beyond those items out-weighs the problems discussed above. The use of hidden objects when presenting algebra notation is novel and obviously care needs to be taken in training for maximum use by listening readers.

Orientation

One of the tasks specifically probed the participants’ knowledge of the position of the speech pointer. This was only ever focussed upon a single item (as defined above). In all but default browsing, the speech cursor pointed to the first item of a term spoken after a move. The **current** action did not move the cursor and the speech pointer remained at its original position. Pointing the focus to the first item of a new term or object was on the assumption that reading would usually proceedes forwards, left-to-right, through the expression. So, placing the cursor on the first operand would be suitable if the term needed to be explored in further detail. Also, the first operand, rather than the operator was thought to be of most interest. The default browsing differed in that the pointer was located on the last item spoken, given that it always spoke the next term to be unfolded.

The participants gave widely differing accounts of where they thought the pointer would be located. Task 8 in Expression two asked for the pointer location during default browsing. When the unfolding reached $+7x^2$ they were asked at which item the pointer was located. At this stage the concept of an **item** had not been explained and this complicated the task. C3 thought the cursor would be on the superscript two, but when told that x^2 was an item, adjusted his decision to that object. C1 assumed the pointer was upon the 7 of the term. C4 thought the cursor was either at the $+$ sign or on $7x^2$, when reminded that the pointer was on a single item he changed his mind to $+$. C2 assumed the pointer was on the 7. Participant C5 thought the pointer would be on the 7. A similar question was asked of C3 for the *next term* command. In Expression one the participant moved from the y to the $= 7x$, with the pointer then located on the 7. C3 thought the cursor was on $7x$, as a result of the cursor being on the x^2 (two objects) in the previous task. Again, after being reminded of the singularity of the pointer he changed his mind to x.

There were not enough specific tasks within the evaluation to form a clear idea of whether the movement commands placed the user at a suitable location. The examples above may tend towards the beginning of the new object as being the appropriate location. Writing or manipulation tasks would give a better area for discovering appropriate location as such tasks would involve finer grained action than the simple reading tasks undertaken in this evaluation. The default locations of movements will be kept as they were originally designed, but note should be taken that the behaviour of the system needs to be explicitly and clearly taught to the user and that there may be a need for subsequent redesign.

Users maintained location within the list fairly consistently, but would sometimes become lost or double check location with the **current expression** command.

Participants were often lost within complex objects. For example, in Expression four, after moving to the end of an expression the participant was asked to explain where they were in the structure. Only two answered correctly and with any confidence. The question was asked after the expression had been read when participants may have been expected to have formed an idea of the overall structure.

Participant C4, Expression 4 task 20:

1. …*do you want to move to the end of the expression.*

2. `nine (end sound)`.
3. *Where are you in the expression?*
4. um, I'm at the end of the second term in the denominator.

In contrast,

Participant C3, Expression 4 task 20:

1. *can you move to the end of the expression.*
2. **ee** `nine`.
3. *can you tell me where you are in the general structure of the expression?*
4. besides at the end? …I don't know. I'd have to look at the whole thing, or browse it to find out.
5. *remember you can do things like…*
6. current level.

Typical of the other participants, C3 did not know where the current location was within the overall structure. This was obviously a complex task, there were a large number of tasks between the original overview and this orientation question. The system lacks a quick global overview and orientation device.

Expression eight also caused similar orientation problems. Some participants seemed to work very hard before realising that there was a nested sub-expression within the first sub-expression.

Participant C5, Expression 7 task 35:

1. **cl** `four times a quantity equals seven`.
2. So the quantity, four times a quantity equals seven, Does that mean the quantity equals seven?
3. …*that means* …
4. four times something equals seven.
5. *you've got a four times something big in brackets and then equals seven. So the current item is the quantity, do you just want to check that?*
6. **ci** `a quantity`.
7. *right, go into the quantity.*
8. Will it work, i q?
9. **iq** `the quantity x`.

10. So its four x equals seven? Oh hang on.
11. *tell me what the current quantity is now.*
12. **cl** `the quantity x plus three times a quantity`.
13. Oh.
14. *So this quantity itself is x plus three times a quantity.*

This example shows C5 working very hard to find out the structure. The interaction breaks down and C5 has to be coaxed through the expression until the nesting of the sub-expression was revealed. The participant's model of the hidden objects had broken down from an earlier confidence and C5 seemed to be overwhelmed by the complexity of the task and the complexity of the browsing needed to reveal the structure.

Getting lost does not necessarily indicate poor usability of the browsing functions or browsing language, but inherent difficulties of maintaining orientation within such complex environments. This is very much the same situation as described by sighted hypertext users in Chapter 2. However not being able to retain the structure may be more than simply becoming lost. The main problem was not having a sufficiently permanent representations of the structure of complex expressions. The listening reader in this case does not have the same context of position on the page and obvious boundaries in which to orient him- or herself. There is a need for extra feedback or browsing functionality to allow the listening reader to orientate him- or herself. The ability to glance at the whole expression, from anywhere within the expression, could help solve some of the problems of maintaining an overall view of the expression's structure.

On returning to a previously read expression, the resumption of the previous position causes many problems. Users eventually adopted the strategy of always returning to the beginning of expression. With only reading (and not manipulation) tasks, the need for holding positions in equations is reduced and the facility should be made optional.

Non-speech Audio Feedback

The terminus sounds (described in Section 4.3.3) were appreciated by all the users, each of whom commented on their usefulness. Both beginning and end sounds were used to confirm location. The end sound was particularly useful when using the default browsing. As each key-press moved the focus of attention forward, it would be possible to miss the end of a complex object if it were not announced in some manner.

One problem was noticed in the use of the sounds, or in fact, use of a lack of sound. On several occasions participants did not take the lack of a end sound to indicate there was more material to read. This happened in two situations. On entry to a complex object, using the default style, the first term was uttered. Even when no end sound was heard, participants assumed there was no more to hear, assuming the whole contents were spoken automatically.

Participant C3, Expression 3 task 10:

1. **space** `three times a quantity`.
2. …It will read the next chunk in more detail.
3. **space** `the quantity x`.
4. …The quantity x?
5. *Carry on.*
6. err, I was expecting it to say a bit more there, it was reading that quantity.
7. *But it reads things a term at a time.*
8. So this is the first bit, the x plus blah blah blah whatever.
9. **space** `plus six`.
10. That's the end of that complex bit.

Participants recovered and no longer made the mistake after being reminded of the term-by-term nature of the default style. The second situation was to assume the end of the whole expression without hearing an end sound. This was very similar in nature to the first, but appeared to happen after the participant had read a complex object and left that object. The participants seemed to assume that they had read enough to establish a complete expression. There are two solutions to this problem: A reminder of more information to read and to give the reader an expectation of what is available to read. A sound could be designed that would indicate there was more to read. When leaving a complex object and the term on the previous level spoken, this sound would follow prompting the reader to continue. The alternative would be to see if practice would train listening readers to use the lack of an end sound to fulfill the same purpose. This could be combined with the second solution: The provision of an expectation of what is to come. This could be provided by a preview or glance at the expression before it was read to apprehend its overall structure and rapidly review the structure while browsing. The use of **current level**, as described above, provides the beginning of such a glance. Participants had already started to use this glance as a strategy and practice may lead to reduction in premature termination.

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. The audio glance described in the next chapter provides a solution to some of these problems by associating different timbres with each structural environment.

A minor observation was made by three participants about the positioning of the beginning sound. This was played after the spoken object to be consistent with the position of the end sound and to form ‘audio brackets’. These participants noted that it should be placed before the spoken object to appear like parentheses around the expression or complex object. The participants’ model of brackets was more consistent with printed brackets than the designer’s!

The other comments pertained to the lack of terminus sounds in all expected positions. They had only been placed to appear when a user actually moved to an object adjacent to the terminus of a level. Participants wanted them to be played whenever an object was spoken which was adjacent to a terminus. This principally meant adding terminus sound to **current** actions and on entry to a complex object with **into** and the default style.

It was also decided to have both terminus sounds played when the spoken object was at the beginning and end of a level, e.g., the denominator ‘two a’ in Expression 4.1. When whole expressions or levels were spoken the sounds were not played as the listener should realise the boundaries are there.

4.4.3 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.
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 action **speak** 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.
4. 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.
5. 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 ‘climbing’ out of nested complex objects.

6. When speaking a term the operator to the left is spoken, to avoid confusion when moving backwards.
7. The terminus sounds were made consistent with the timbres used in the audio glance (see Chapter 5).
8. The terminus sound at the end of the whole expression was repeated to add this information to the general end sound.
9. The start sounds were played before the spoken object to 'bracket' the objects with terminus sounds.
10. An attempt was made to ensure the default reading style no longer repeated the current object when another command had been used during use of the default reading. Unfortunately this could not be implemented reliably and some objects were missed.
11. The key-stroke error sound was made more meaningful by replaying the PC speaker beep with a typewriter sound, linking it to the key-stroke.
12. Users could recover from a key-stroke error by using the `backspace` key. A mute function was added that not only terminated all speech, but cleared any remaining key-strokes in the buffer.
13. The **keep simple** action was designed, but not implemented. This would enable a reader to label nested complex objects to be spoken in full during a **show** command.

4.4.4 Problems with Experimental Design

Taken as a whole the evaluation method used here worked very well for the specific needs of this component of the Mathtalk program. It allowed quick evaluations which immediately probed the usability of the chosen design. The rapidity of the method allowed an iterative design method to be used, finally producing robust, effective set of browsing functions and command language. However there were some general observations to be made about the final evaluation of the browsing component.

A short evaluation of this type does not reveal how readers would use the control facilities when fully conversant with the style of interaction. The beginnings of strategy development were observed, but a longer term evaluation would be needed to fully investigate this aspect of the control component. Nevertheless, the basic usability of the browsing was demonstrated and the potential of strategy development observed.

A strong feature of the command language is its basis in a natural spoken form. This meant most of the tasks were worded in a manner that contained the command to be used. This makes the language easy to use and learn, but also means that the participant did not have to work at remembering commands or deciding on which one to use, except in those tasks that did not contain command words. Using different words to those appearing in the commands was not a realistic choice. Using different words would have made artificial, contrived wording, and missed the central purpose of designing the language based on naturally occurring phrases. The use of some tasks without command words also tested participants' recall of commands, and the ability of users to use commands not explicitly taught demonstrated the generality learnability of the language.

All the tasks were structure based and the command language was designed to browse structure, so the results might have been expected to have been good. The tasks seemed to be representative of those a reader would need to make during the reading of algebra notation. A full task analysis of people reading and using algebra notation would be needed to justify the decision to make a structure based language and to justify the design of the tasks. This was not within the scope of this project.

The mathematical tasks used at the end of the evaluation were not complex enough to fully test the control. Two of the expressions were too easily internalised so avoiding use of the control features. The evaluation of Expression two, did however, show the utility of the default browsing strategy for such an evaluation. This excursion into mathematical tasks was very useful for the final evaluation of the Mathtalk program (see Chapter 6).

Whilst resuming the previous position when revisiting an expression proved unsatisfactory, some tasks and expressions could have been included that demonstrated its utility when comparing expressions. Expressions five and six could have been rewritten as:

5 $y = x^n + 1$

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

When comparing the objects with superscripts, retaining the previously held position would have been useful, whilst always returning to the beginning may have been irritating.

4.5 Discussion and Conclusions

This chapter has described the development and evaluation of a browsing component for the Mathtalk program that aims to give the reader control over the flow of information. A set of moves for browsing around an expression's structure were implemented and a browsing language for controlling their use was designed. The design concentrated upon the browsing components of transition (the moves) and orientation within the expression. The browsing language was based on a set of *action* and *target* words that could be combined to generate moves to any part of the expression with a granularity based on the structural elements of the expression. The browsing language also allowed review without movement.

The following set of design principles arise from this investigation:

- The addition of browsing can give control over information flow.
- Basing the browsing on the structure of an expression can generate the moves or transitions necessary for reading an expression.
- The language should be designed to give the listening reader fast and accurate control over information flow. A structure based language gave the accuracy component of selection and the command language gave rapid access to this structure.
- The reading, not the control must remain as the top priority of interaction. This means that superfluous feedback must be reduced. An example of this is not echoing the commands given and using the results of the moves to give feedback.
- Hidden objects help to reduce speech overload and help remove grouping ambiguity. The hidden objects also gave rise to a spoken overview of expressions containing complex objects.
- Simple non-speech audio sounds that indicate boundaries of complex objects assist in maintaining context and expectations during browsing.

The browsing language, based on a natural spoken form, was easy to teach and proved easy to learn by the users. Commands could be extracted easily from the spoken form. The labels used for both actions and targets in the language were readily adopted by the users.

The browsing language and functionality cover a wide range of moves and enabled the users to perform most of the actions they required. the evaluation enabled some gaps in the browsing to be found and the language and functionality were extended to resolve these problems.

The principle of designing for simple and complex structure re-appeared as hidden complex objects in the display. These allowed the amount of speech given to be reduced. This should reduce the mental overheads encountered by listeners and increase the speed of interaction. This lead to widespread use of the **current level** command to give an overview of structurally complex expression. By naming structures and giving their scope the hidden objects reduce grouping ambiguity in the output. During browsing the hidden objects should also make it easier to locate and move beyond complex objects, rather than having to move through that objects contents.

The main usability problem during browsing was orientation within the expression and retention of overall structure during browsing when away from the top level. The browsing language offers solutions to gaining local orientation with the terminus sounds and the **current level** command, but global orientation within the expression is difficult. The audio glance described in Chapter 5 provides a potential solution to the global orientation problem.

Chapter 5

The Audio Glance

5.1 Introduction

The two major interface design principles for an auditory reading of algebra notation are now in place. First, the spoken presentation has been improved by the addition of prosody. Secondly, passive listening has been replaced by active reading by the addition of browsing functions that give fast and accurate control over the information flow. This chapter describes the last component in the interface, namely an audio glance called *algebra earcons*.

A scan or glance is proposed as the first stage in the reading of an expression (Ernest 1987). A glance is usually not possible for a listening 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.

This chapter starts by describing the need for a glance in detail, defines a glance and then of what this glance should consist. The audio glance described here uses non-speech sounds. The reasons for this choice are presented, before describing the development of an audio glance called algebra earcons from the prosodic component of the speech described in Chapter 3 and standard earcons as described by Blattner, Sumikawa, and Greenberg (1989). The rules for constructing algebra earcons are described in detail, together with examples.

Two experiments were performed to assess the utility of the audio glance. The first examined whether algebra earcons presented enough information for a listener to recognise a printed expression. This experiment also assessed the efficacy of the presentation of individual components of an expression. The second experiment re-examined a modified audio glance, then probed the recall of an expression to further gauge the utility of the glance.

The audio glance provided by algebra earcons was found to work as specified. Listeners were able to recover a large amount of useful information, from a simple idea of the amount of material present in an expression up to a complete and correct representation.

5.2 What is a Glance?

A glance is best described as an overview or general impression of the nature of an object or environment. A glance is an operation familiar to most people, but is something ill-defined. Many people are familiar with taking a glance at a room to check its contents, either human or inanimate, to confirm the presence of a particular person or object. The important features of this glance are that it is rapid; detail is omitted and only the salient information appears to be made available to the viewer. It is also useful to note that the glance is unreliable; it is possible to miss the person or object being sought. The ability to glance comes from the fine control over what is being viewed by the visual system. In the following paragraphs some descriptions of what can be thought of as glances will be given. From these descriptions a working definition of a glance will be made that can serve as the basis for the design of an audio glance at algebra notation.

An abstract of an academic paper is a glance at the contents of that paper. The abstract should encapsulate, in a simple form, each of the principle features and arguments in the document. The reader uses this glance to decide on whether to read the paper and to create expectations of the contents of that paper. Thus the abstract gives a rapid overview of both the structure and content of the paper, without the detail.

A glance can be used on a document at a higher level than the abstract. The manner in which the print is laid out on the page can give the reader information about the contents (Southall 1988). The black print on a white page is divided into paragraphs; section headings are prominent; tables, diagrams and figures have formats that enable them to be located with a ‘glance’. This is a glance at the physical structure of a document, whereas an abstract is a glance at the argument structure, and indirectly the physical structure. To plan the reading of an algebra expression the reader needs to apprehend the physical structure.

Document previewers provide a glance at the overall format of a document. The view of the document can be progressively shrunk until many pages appear on the screen and the elements of the document appear as black shapes. All the detail of the text is hidden, but the writer may see the arrangement of paragraphs, tables and figures on the pages to check for typographic appearance.

From these descriptions of ‘glances’ the following definition can be made: 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. In the highest-level view, or glance, this is merely the size or length of the expression. The representation gained from a glance could also extend to a full framework for the expression that lacks only the lexical detail.

Thus this glance is not simply the length of the expression, but also the type of objects within that expression that define its complexity. To do this the glance should contain information about the presence of 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 extract information that is useful at several levels. While doing this, the glance at an expression must be quicker than the expression spoken in full.

5.3 The Need for and Nature of the Glance

Ernest (1987) proposes that a scan is the first stage of reading an expression. At this stage the reader can gauge the complexity of the expression to see if it is manageable and create expectations. This idea is supported by 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. For example, a glance shows where all the variables x lie within the expression and prompting the user to move them all to one side. In addition Ernest suggests that the reader may use a glance to review the expression for any unknown or difficult symbols. This gives the reader the opportunity to abort the task or select particular strategies for reading and solving an expression.

Such a scan or glance is facilitated by the spatial nature of printed algebra, as described by Kirshner (1989). The spacing of the expression into terms, the elevation of superscripts, vertical juxtaposition of fractions with a fraction-line and obtrusive nature of other symbols all help to give an expression ‘shape’ as well as simply length that can give an impression of type and complexity of the expression. This overview is all part of the utility of paper as an external memory together with the control afforded by the visual system.

This type of overview or first impression is not easily available for a blind reader. The only way to ascertain the nature of the expression is to read the expression in full. Each object has to be read in full detail, retained to be integrated at the end of the reading process or as that process continues or is tediously repeated. All this adds to the mental workload experienced by the listening reader. With no idea of whether the expression is even long or short the listener may be easily overwhelmed by a long expression or surprised by a shorter one. Either repeating an expression or having to search for a particular expression is potentially slow, tedious and frustrating.

The ability to scan an expression for certain structural features, such as a fraction or superscript or pattern of elements, could facilitate searching and create appropriate expectations on which to base the reading of an expression. A glance could give the top-down component of reading an algebra expression described by both Ernest (1987) and Ranney (1987). Without the creation of expectations for a framework, a listener may be locked into a reading strategy of integrating an unknown number of items into a structure of which there is no foreknowledge.

Another aspect of the need for a glance can be gained by examining how active reading is achieved in the Mathtalk program. Mathtalk offers browsing based on the structure of an expression. For the most efficient and effective reading strategies to be developed from the browsing moves, the reader needs to know the overall structure or nature of the expression before detailed reading begins. A short and simple expression could easily be read with a full utterance. A structurally simple expression, but one that is long with repeating subunits would probably best be read term-by-term. An expression that is long and complex would best be read by unfolding complex structures from a named type to reveal contents in a controlled manner (see Chapter 4).

Without reading the expressions in full the listening reader has no idea of whether the expression is short, long or complex. Even a simple glance at structure could have great benefits for the listening reader by judging these types.

So the glance to be designed within Mathtalk will be a glance at the structure of an expression. This is consistent with the notion that the main purpose of the display was to present the structure or grouping within an expression and that the browsing was movement around that expression.

A glance could make the listening reading interaction more efficient and effective. An idea of the structural nature of an expression could avoid ambiguity of grouping and allow the reader to generate appropriate strategies for the reading of that expression. The ability to combine a top-down view with a bottom-up approach to reading, as proposed for the visual reading of algebra expression could provide the most efficient and effective way to accomplish an auditorily based reading.

5.4 Choice of Medium

There was a choice of medium in which to present an audio glance: Synthetic speech or non-speech audio. Whichever medium was chosen had to be able to fulfill the criteria described above: Rapidity; presence of type, but not instance, of object; location of objects and relative size of object.

There are several ways in which synthetic speech could be used to present an audio glance to a listening reader. A full utterance could be used and the listener left to extract the information salient for the glance. This is potentially very long and the detail that must be presented could well intrude into the reading process. This integrating of a large amount of detail into an overall structure is what the audio glance should attempt to avoid.

A text-based structural description could be given. For example, ‘a single operand equals a fraction with a large numerator and short denominator’ would be a structural description of the expression:

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

If such a description were short enough to be useful, too much information would be lost. For example 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 the expression:

$$5x^5 + 4x^4 + 2x^2 + 9x + 7 = 0$$

could be described as ‘a quintic in x’ There are several problems with this approach: the description ‘quintic’ accurately describes $x^5 = y$ as well as the expression above. The phrases and descriptions used may not be meaningful to the level of user or a specific mathematical description may not exist; finally, an interpretive mathematical description contravenes a basic design principle of Mathtalk, that the reader performs any mathematical interpretation.

Compressed synthetic speech could also serve as a glance. SpeechSkimmer, developed by Arons (1993), uses speech compression algorithms that speed up recordings of natural speech, but retain many of the prosodic cues that indicate document structure. The use of prosodic cues to indicate structure is central to the development of the non-speech audio glance used in Mathtalk. Speeded up speech, that retained structural cues such as division into terms and the grouping of objects into complex items would have many features of a glance as described above. The only information lacking would be on the type of the object being represented. For example a fraction may appear as two adjacent sub-expressions (see Chapter 3). Similarly, a superscript would not be differentiated from a simple term to which it was attached. 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, compared to those used by Arons in his SpeechSkimmer.

The **current level** command described in the prior chapter gives a glance at the structure of complex expressions. Such objects are rendered by simply referring to their type. This method, whilst useful during browsing, is not a general solution. Simple objects are rendered in full. Complex objects are rendered as type with location, but no size.

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, e.g., that a particular object is the letter ‘a’. Instead the type of an object can be readily expressed in sound: Different musical timbres could be associated with different classes of object in an expression. The occurrence of a particular object, for example, letter or fraction etc. can be given without giving any detail. 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 fulfilled. The delivery of non-speech audio can be rapid. It should be possible to play a short sound in order to recognise the associated object type, where the spoken form may be much longer. The length of notes can also be proportional to the size of the object, giving the size of complex objects.

The last reason for choosing non-speech audio for the glance was for longer-term design opportunities. In Section 2.3 the suffix effect was reviewed. This effect operates across modalities: vision and audition. Within the auditory mode speech and non-speech sounds operate in separate channels (Baddeley 1986). This means that non-speech sounds played after a spoken phrase will not seriously affect the retention of that speech. Similarly, incoming speech does not seriously disrupt non-speech sounds already present in short term memory. This means that information can be presented in two separate channels without undue interference. Thus the amount of information can be increased giving greater effectiveness and efficiency in the reading interaction.

Initially the audio glance and the speech based reading would both be intimately mixed. A reader may glance, process the information, then start browsing with speech. However, as will be seen later, the audio glance works by associating different musical timbres with the structural types within an expression. This association can be exploited within the browsing process to aid navigation and orientation. This ability to re-use the association between musical timbre without one source of information interfering with the other has important design implications. This exploitation was introduced in Chapter 4 and will be described in detail in Section 5.10.

5.5 Development of the Audio Glance

One method used to add non-speech audio sounds to the computer-user interface is the earcon (Blattner, Sumikawa, and Greenberg 1989; Brewster, Wright, and Edwards 1994a). Earcons are abstract structured sequences of non-speech audio used to give messages in the computer interface. The audio glance developed in this chapter 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.

The purpose of the audio glance is to give a quick summary of the structure or grouping within an expression without giving all the detail. Prosody can indicate the structure of an utterance, but the speech signal also carries the lexical detail of the expression; the exact detail of each object. The requirements for the audio glance would be fulfilled by presenting the listener with prosody without the lexical detail or prosody without the speech. The size and location of objects can be delivered using prosody. The main task of the audio glance is to replace the lexical detail while retaining the structure presenting properties of prosody. The following section describes earcons, the method chosen for presenting the structure of an expression and subsequent sections develop the idea of the audio glance and present rules for its construction.

5.5.1 Review of Earcons

Earcons were developed by Blattner and colleagues (Blattner, Sumikawa, and Greenberg 1989; Sumikawa, Blattner, Joy, and Greenberg 1986; Sumikawa, Blattner, and Greenberg 1986; Sumikawa 1985). They use abstract, synthetic tones in structured combinations to create auditory messages. Blattner et al. (1989, p13) define earcons as ‘non-verbal audio messages that are used in the computer/user interface to provide information to the user about some computer object, operation or interaction’.

The basic building block of an earcon is the motive. These are short, rhythmic sequences of pitches that can be combined in different ways. Sumikawa et al. (1986, p5) describe them as: ‘A motive is a brief succession of pitches arranged in such a way as to produce a tonal pattern sufficiently distinct to allow it to function as an individual recognisable entity’. In addition: ‘The eloquence of motives lies in their ability to be combined to create larger recognisable structures. The repetition of motives, either exact or varied, or the linking of several different motives produces larger, more self sufficient patterns. We use these larger structures for earcons’.

The most important features of motives are: Rhythm, pitch, timbre, register and dynamics. These can be varied to create motives that are sufficiently different to be discriminated and therefore useful as bearers of messages. Repetition, variation and contrast in use of the parameters described above are used to design earcons.

Blattner et al. (1989) describe two types of earcons: Compound earcons and family, or hierarchical earcons. Compound earcons are simply made from concatenated motives for elements of a message. Brewster (1994) gives an example from a series of file actions. These individual motives for file actions could then be combined in different ways to provide information about any interaction on the file. Such single element motives could be, for example, ‘create’, ‘destroy’, ‘file’ and ‘string’ these could then be concatenated to form earcons. For the ‘create file’ earcon the

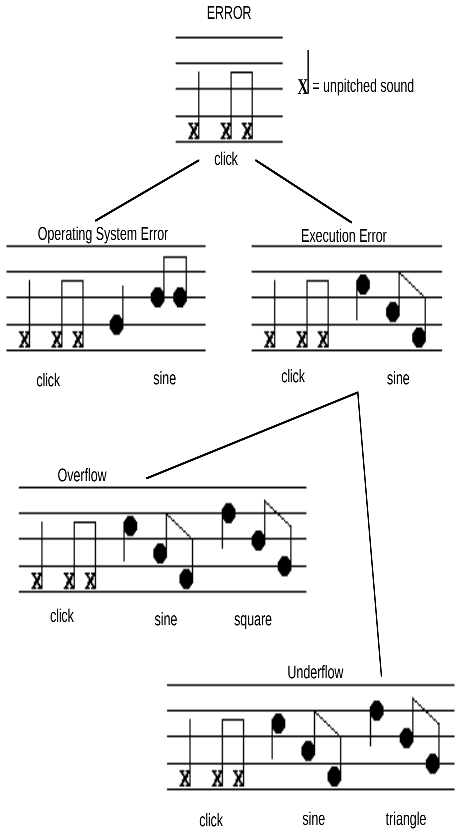


Figure 5.1: A hierarchy of family earcons representing errors (adapted from Blattner et al.).

‘create’ motive is simply followed by the ’file’ motive. This provides a simple and effective method for building up complex messages in sound.

The second type of audio message is called family or hierarchical earcons. Such an earcon is shown in Figure 5.1. Each earcon is a node on a tree and inherits all the properties of the earcons above. For example, the top level of the tree is the family rhythm, in this case it is a sound representing error. This sound just has a rhythm and no pitch, the sounds used are clicks. The rhythmic structure of level one is inherited by level two but this time a second motive is added where pitches are put to the rhythm. At this level, Sumikawa suggests the timbre should be a sine wave, which produces a ‘colourless’ sound. This is done so that at level three the timbre can be varied. At level three the pitch is also raised by a semitone to make it easier to differentiate from the pitches inherited from level two. Other levels can be created where register and dynamics are varied. Each level in the tree adds more parameters to the sound gradually making the information more specific. In the error family earcon, error can be split into file or command error by rhythm. At the next level the error timbre and command rhythm can be further varied by pitch to give specific command errors.

5.5.2 Linking Prosody and Earcons

There are several interesting parallels between the prosodic component of speech and earcons. At a basic level they are essentially described by the same parameters of rhythm, pitch, amplitude, tempo and timbre. Earcons are musical sounds and the inconsistent, individual differences inherent in speech make it fit uncomfortably within the musical paradigm (Crystal 1987). In addition, earcons convey their message via the structure of the sound and for the required glance the message is structure. These fundamental similarities offer a compelling basis to link spoken and non-speech messages. If the required information, that is, structure, is contained within the prosodic component of speech, but the lexical part of speech stops the presentation becoming the required glance, then the possibility of presenting prosody in musical terms should be explored.

Another parallel exists between the prosody of algebra and the construction of earcons. This is at the level of the term. The spoken expression is divided into a series of terms. Compound earcons are constructed from one or more motives linked together to give a message. The term can be regarded as an equivalent of the motive. An algebra earcon could be constructed from a series of term based motives, each representing the structure of that one term, to give the whole structure of the expression. Brewster et al. (1994a) suggest separating motives by pauses of at least 100 ms, a pause similar to that seen between spoken terms. Distinct pitch and tempo changes are recommended to make the structure of each motive easy to comprehend. Similarly, prosody uses pitch and tempo changes to make the structure of an expression obvious.

Brewster *et al* suggest that rhythm and timbre are the most important cues that aid discrimination of earconic messages. At present these two parameters do not have a high priority in the prosodic presentation of algebra. Musical timbres are proposed below as a replacement for the spoken objects in an expression to hide detail, but still present the type of an object. In the rules for algebra earcon construction, the term is used to form the rhythmic structure of the earcon. These proposals bring algebra earcons directly into line with standard earcons. To make the earcons ‘musical’ and therefore easier to use, a stylised form of prosody is used, but the basis of algebra earcons are still firmly rooted in the prosodic component of speech and form a strong basis to design non-speech audio in the interface.

As described above, an appropriate glance could be provided by using prosody without the speech. The similarity between the guidelines for earcon construction and the rules for algebraic prosody mean that earcons provide a mechanism for displaying prosodic information without the lexical or verbal detail. By using musical tones instead of spoken items in the expression, but retaining the prosodic form, an earcon can be constructed that maps to the structure of an expression via the prosodic form.

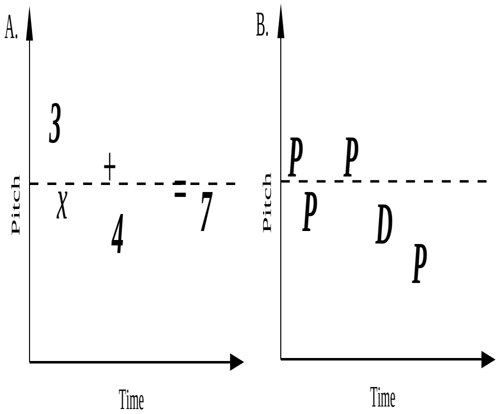


Figure 5.2: Prosodic form of $3x + 4 = 7$ and derived earcon representation based on spoken form. The timbres are **P**iano for letters and numbers, **D**rum for relational operator and white space represents silence.

Figure 5.2 demonstrates how the algebra earcon form of an expression maps to the prosodic form and thus the structure. Algebra earcons work by representing only the structural type and not the instance of an item in an expression. Different musical timbres were used to represent the basic syntactic types within an expression. A piano note is used to represent ordinary letters and numbers at the base level; a drum sound represents a relational operator and silence for printed binary operators, such as $+$ and $-$. The arrangement of the notes in time and pitch has a similar pattern to that seen in the more variable prosodic form. Silence is used, as the important feature for the glance is the division into terms, not the form of that division. Such a representation is in the nature of the glance and silence assists the closure of one term, before the onset of another. The presence of a relational operator is shown explicitly. This is because the relational operator is the first parsing point in an expression, its presence discriminates between expressions and equations and its location indicates the balance of the expression and so is a vital structural cue.

The use of musical tones to replace spoken items serves two purposes in the development of the glance. Firstly, it removes the detail from the expression, but still retains the types of the objects being represented. This gives a more abstract view of the expression. Secondly it preserves the location of the object, another necessary component of the glance. Another requirement is that the glance gives the size of objects. This is only necessary for complex objects such as sub-expressions; that letters and numbers are indicated as present indicates their size. The principle of hiding information in complex objects can be carried forward into the design of the glance. For a glance to work, only the presence and relative size of a sub-expression need be shown. What such a sub-expression contains is not important at first glance. In algebra earcons a cello sound indicates the presence of a sub-expression and the length of that sound, relative to the piano sound for a

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

Table 5.1: Table of musical timbres used in algebra earcons.

letter, indicates the relative length of that object. Simply by replacing the spoken items within an expression the requirements for an audio glance can be fulfilled: the presence and type of objects; the location of these objects; and the size of these objects. Only representing the type of object immediately hides the detail of an expression and the musical sounds should be able to be played much quicker than the default rate of speech. Thus the glance can be rapid, show the presence, location and types of object within an expression.

5.6 Constructing Algebra Earcons

Algebra earcons are constructed by blending the visual representation of algebra syntax with the prosodic cues used when it is spoken. Different objects within an algebra expression were 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 Table 5.1. The timing, pitch and amplitude of these sounds were then manipulated according to the rules below.

A priority was to establish a rhythm by which a listener could group items together, discriminate elements of structure, aid retention, enabling algebraic structure to be presented. An overall rhythm was important so that each term-based motive could fit together into a ‘musical’ whole.

The first stage in the construction of an algebra earcon was the establishment of this rhythm. In spoken algebra the term formed the foot or basic unit of rhythm in the utterances. The foot is the equivalent of a bar in music (Halliday 1970). First a bar length was defined for the earcon. This was based on the length of the longest term in the expression. For simple terms, each object contributed one beat to the bar length. The last operand in a term contributed two beats. This lengthening mimicked the final syllable lengthening in speech. An extra silent beat was added for a printed binary operator. In length calculations, a relational operator was included in the following term, being counted as one beat, plus a separator of a silent beat.

All complex objects (including superscripts) were represented by a continuous tone with a constant pitch, as were non-terminal parenthesised sub-expressions in speech. This simply indicated that such an item was present, but revealed nothing of its contents, except its length and location. This was consistent with the idea that an algebra earcon is a glance. The lengths of complex objects were calculated as above, but binary operators did not make a contribution. This reflected the faster, pauseless uttering of these objects in speech.

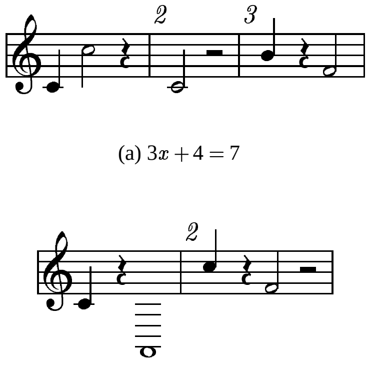
After the maximum term length had been calculated, each term in the expression was fitted into this bar length. Shorter terms were padded at the right with silent beats to preserve the rhythm of the algebra earcon. For the first term or motive, a maximum of two silent beats was allowed. This avoided long pauses at the start of the expression that could disrupt or prevent the establishment of a rhythm within the earcon.

Algebra earcons were played in the C major scale. The pitch of each new term started at middle C (C_3). Subsequent objects were played at one note below the previous. The last term's pitch started at A_4. This mimicked 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 were played at a pitch two notes higher than their base, in the octave above. At this point there was a dissociation between the earconic form for a superscript and the spoken form. In the spoken form, superscripts followed the pitch fall of the term to which they were attached. The higher pitch used to represent the superscripts in an earcon were chosen as a correlate of the higher position in print. The pitch change was introduced to add redundancy to the indication of a pitch. Simply using the musical timbre to find the superscripts may not have been sufficient if the pitch trend simply followed that of the rest of the term.

Sub-expressions were played two octaves and two notes below the preceding object or initial pitch for a term if the quantity was the first item. This form of presenting sub-expressions mimicked that of those spoken in the middle of the utterance. The linear pitch falls seen for complex objects and the termini of utterances were not used in order to make the form of the earcon as simple as possible. This reasoning was also used to exclude the declination effect in the earcon. The sharp pitch fall of the hat effect was used to signal the termination of the earcon. The rules for algebraic prosody were not directly mapped to the earcon. A more stylised, restricted form was used to make the earcons as simple and as predictable for the listeners as possible.

Simple and complex fractions were both represented by pan pipes, but with a different pitch profile. Simple fractions had the same pitch fall throughout as simple terms, but there was a one octave drop at the start of the denominator. The last note in a simple fraction was lengthened as if it was the last note in a term. Again, this representation was very similar to that of such fractions in the spoken form. Complex fractions were represented by two long notes of constant pitch, separated by two silent beats. This change in representations for complex fractions mimics the similarity



(a) $3x+4=7$

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

Figure 5.3: 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.

between complex fractions and parenthesised sub-expressions, which was also seen in the spoken form. The silence between the two terms of the fraction represented the fraction line or ‘over’. The second note, the denominator, was played two notes lower than the first. This attempted to indicate that the denominator was ‘lower’ and separate from the numerator.

For all complex objects any objects appearing as a prefix or suffix were separated from the complex object by a silent beat. This mimicked the separation seen in the spoken forms. Such separations were thought to aid discrimination between objects in the earcon. This rule means that two sub-expressions $(a+b)(a-b)$ would be separated by a silent beat. Similarly, $(x+1)^2$would have a silent beat between the sub-expression and the superscript.

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

For example the expression $3x+4=7$ has three terms making a three bar algebra earcon. The musical form for this earcon can be seen in Figure 5.3(a). 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 silent beat for the ‘$+$’ which separates it from the following term. The second 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 was added for the minimal separation of one silent beat from the next term or motive. The final term ‘$=7$’ has a length of four beats. One for the equals symbol, one silent beat separating this from the ‘7’ and two for the ‘7’ itself. A separator beat was not added to this term as it was 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 was 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 emphasise 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 $3(x+4)=7$.

The example $3(x+4)=7$ (see Figure 5.3(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 silent beat was 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 silence after the final term, represents any rest that is needed to complete the bar length.

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.

In a similar manner the equation $x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ can be reduced to an algebra earcon giving an audio glance of only four tones. A piano for the 'x', a marimba for the '$=$', a long pan pipe note of constant pitch for the long numerator of the fraction (the root was regarded as a quantity) and a shorter pan pipe note for the much smaller denominator. Just as the **current level** command greatly reduced the amount of speech, the hidden objects also work in the algebra earcon to reduce the number of sounds presented for complex expressions. This simplicity is an integral part of a glance. However there is a potential problem of a dissociation between the number of objects in a glance and the complexity. The complex expression above was reduced to four tones, where the simple expression $3x+4=7$ is played as 5 tones in its earcon.

A relatively small set of rules can give a general mapping from an algebra expression to an algebra earcon glance. Care was taken to give the earcons a strong rhythmic component, to aid retention and discrimination of syntactic structure (Deutsch 1982). Distinctive timbres must be used to aid this discrimination (Brewster, Wright, and Edwards 1994a). Also following Brewster's guidelines, pitch changes used to contrast different classes of object were made distinctive by making the differences at least one octave.

5.7 Evaluating the Audio Glance: Experiment One

Two experiments were performed to assess the ability of algebra earcons to act as an audio glance. The aims of the experiments were narrow. They sought only 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. That is, the usefulness and usability of the earcons was not investigated. This sort of investigation was part of the final evaluation of the integrated Mathtalk program described in Chapter 6.

A simple multiple choice design was used to find whether listeners could recover enough information to pick an expression and could they recall enough information that could be used as a glance. This method also allowed flaws in the construction of algebra earcons to be probed. The multiple choice design did not reveal anything about the internal representation gained from the audio glance. In the second experiment a recall section was added before the same multiple choice design used in the first experiment. The recall section allowed the representations held by the listeners to be probed and the repeat of the multiple choice design allowed a further investigation of improvement to rules changed after the first experiment.

5.7.1 Design

The aim of the first experiment was to assess the basic ability of algebra earcons to work as a glance. First, could listeners recover enough information about the objects within an expression such that they could determine its type? Secondly, do algebra earcons present all types of object within an expression to equal effect? To this end, a multiple choice paradigm was used to fulfill the limited aims of the experiment. An advantage of this design is that the distractors presented along side the stimulus can be so designed such that all aspects of the rules for constructing algebra earcons can be probed.

A two-condition, within-participants design was used. With four choices for each stimulus in the multiple choice design, a significant bias in answers towards the correct answer amongst the four choices would suggest the earcons were successful in presenting 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 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.

Two sets of expressions were used. First the rules for the simple expressions were investigated (simple condition). Then a second set (complex condition) investigated the presentation of more complex structures.

5.7.2 Participants

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. The participants were also mixed as to their level of musical training. A participant was musically trained if he or she had learnt or currently played a musical instrument or sang. The group was not balanced for any of these factors. All participants were familiar with the form of algebra expressions and could name parts of expressions.

5.7.3 Materials

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.

Training Expressions for Simple Condition

1. $a+b$
2. $ab+c$
3. a^b
4. $a=b$
5. a/b
6. $ab^c=d$
7. $\frac{a}{bc}=d+e$

Stimuli Expressions for Simple Condition

1. $ab = c$
2. ab^c
3. $ab + cd$
4. $a^b + c^d = e^f$
5. $ax^2 + bx + c = d$
6. $\frac{ab}{c} + \frac{d}{e} = f$
7. $\frac{a}{b} + \frac{c}{d} = e$
8. $a = bc + d$
9. $ax^4 + bx^3 + cx^2 + dx + e = 0$
10. $ab + cd = e$
11. $ab^c = d + e$
12. $a + b = cd$
13. $\frac{a}{b} + \frac{cd}{ef} = g + h$
14. $abc^d + ef^g = h$
15. $ax^3 + bx^2 + cx + d$

Training Expressions for Complex Condition

1. $(a + b)(cd + ef)$
2. $a(b + c) = d$
3. $a - (b + c) = d$
4. a^{c+d}
5. $(a + b)^{c+d}$
6. $\frac{a+b}{c-d}$
7. $(\frac{ab}{(c+d)(e+f)})^g$

Stimuli Expressions for Complex Condition

1. $(a+b)(c-d)$
2. $a=b(c^d-ef+g)$
3. $\frac{ab}{(c+d)(e+f)}$
4. $a+(b^c-d)^e+f$
5. $\frac{a}{b}(cd+e)=f+g$
6. $(ab+c)^{de+f}$
7. $a=\frac{b+(c^d-efg)}{hi}$
8. $a=(b+c)^d-e$
9. $a(b+c)-d(e-f)^g=h$
10. $(a+b)^{c+d}$
11. $a(b+c(d+ef))=g$
12. $a+b(c+d)+\frac{ef}{g+h}$
13. $(a+b)^c(d+e)^f=g$
14. $\frac{a}{b+c}(\frac{de}{f})^g$
15. $(a+b)+\frac{c}{d}(e+f)^{g+h}=i^j$

The stimuli were ‘partially shuffled’. The short and long expressions were mixed, but reordered to ensure that longer, more complex earcons did not appear at the start of the experiment.

Appendix C.1 shows the multiple choice questions for each of the stimuli expressions. Three distractors were constructed for each of the stimuli. In each distractor part of the original stimulus expression was transposed, transformed or removed. Figures 5.4(a) and 5.4(b) show two sample multiple choice questions. The responses within each question were randomly ordered.

The algebra earcons were hand-crafted using the rules described above in an Ez Vision sequencer on an Apple Macintosh. The algebra earcon sounds were produced on a Yamaha DP110 synthesiser controlled by an Apple Macintosh and played to the listener via external loudspeakers.

A $\boxed{ax^3 + bx^2 + cx + d}$

B $x^3 + ax^2 + bx + c$

C $ax^2 + bx + cx + d$

D $ax^3 + bx^2 + cx + d = e$

(a) Question 15 simple condition

A $\boxed{\frac{a}{b}(cd + e) = f + g}$

B $ab(cd + e) = f + g$

C $\frac{a}{b} + (cd + e) = f + g$

D $\frac{a}{b}(cd + e) + fg$

(b) Question 5 complex condition

Figure 5.4: Questions 15 from the simple condition and 5 from the complex condition. Correct responses appear in boxes.

5.7.4 Procedure

The aims of the experiment and nature of the algebra earcons were carefully explained to the participants using a prepared script. The link to the prosodic component of spoken algebra was used as the basis of the training. The training expression was spoken by the experimenter, with the key features of the prosody indicated. Then the algebra earcon was played and the link to the spoken expression explained. The expression was spoken again and the algebra earcon played once more. The participant was then asked if he or she required further explanation or that the algebra earcon be played again.

The ordering of the training expressions in the simple condition was particularly important as this formed the basic training for all algebra earcons. The first expression $a + b$ introduced the piano timbre and the silent gap to indicate a $+$ or $-$ operator. The pitch fall for a final term could also be introduced. The second expression $ab + c$ reinforced these points, but also introduced the pitch fall within a term. The expression ab^c introduced the superscript timbre and its relationship to the rest of the term. Then the marimba timbre for the relational operator was given. The last element to be introduced was the simple fraction. Subsequent training expressions put these components together in different orders to allow more practice.

The training for the complex condition followed the same principles. The only new timbre to be introduced was that for the sub-expressions. The pan-pipe timbre for the fraction appeared in the complex form as well as the simple form already encountered. The similarity of complex fractions to sub-expressions was used in the training. Similarly, the superscript timbre was also encountered in a new long form that representing complex superscripts.

During the experiment, the algebra earcon was played by the experimenter at the instigation of the participant. When the algebra earcon had stopped playing, the participant was handed a card with the stimulus and three distractor expressions. The experimenter remained silent, except to answer

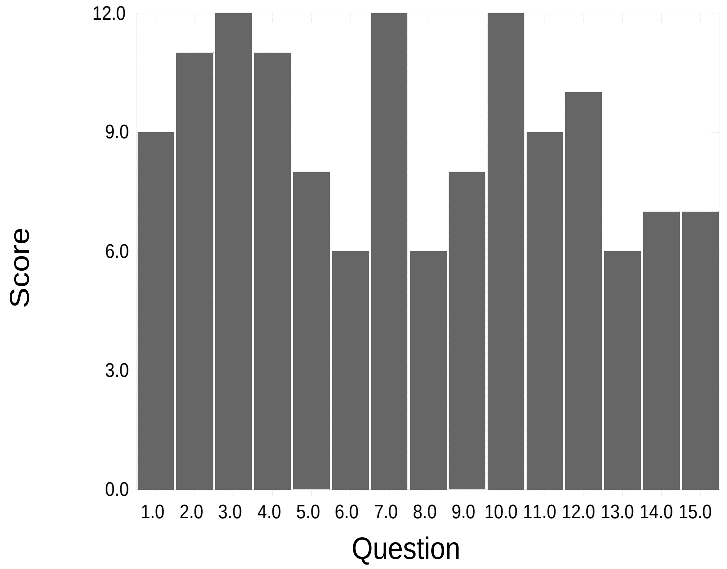


Figure 5.5: Frequency of correct answers for each question for the simple stimuli expressions.

technical questions. This avoided disrupting the participant's concentration or retention of information. Questions about association of timbre to algebraic object were only answered after the participant had finished his or her response. This allowed ease of learning to be probed, but did not place an artificial burden upon the user and the short training program.

The participants were asked to answer all questions, even if that answer were a guess. No time limit was placed upon the participant answering a question.

5.7.5 Results and Discussion

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 Table 5.2 for individual and combined scores for each condition). The raw scores may be seen in Section C.2. 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. Listeners were able to recover enough syntactic information from the algebra earcon to choose an appropriate expression from a list of similar alternatives.

All participants fell into the two upper quartiles. A slight bunching of subjects in the upper quartile suggested there may have been a ceiling effect. Those appearing in the third quartile tended to score worse in only one of the two conditions. Those participants who had musical training performed significantly better in this task (T(22)=3.94, p=0.0007). This training may have enabled

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

Table 5.2: Separate and combined scores for each subject. M = musical training, N = no musical training.

them to extract more information, or recover the information more easily from the audio glance. Earlier work on earcons (Brewster, Wright, and Edwards 1994a) did not find this difference in performance. The difference found here may be due to the more complex and variable stimuli and the very different task. Whether this finding has important implications for the use of algebra earcons will only be shown by the type of mental representations listeners derive from the glance and more longitudinal studies.

Analysis of Errors

An examination of the results across questions revealed which presented the most problems. The choice of incorrect answer should highlight problems with the glance. Figures 5.5 and 5.6 show the frequency of correct answers for each question. 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.

Questions 15 from the simple condition and five from the complex condition (see Figure 5.4) were chosen as representative of the earcons that were prone to errors. For Question 15 (Figure 5.4(a)), only two participants gave answer **B**, which differed from **A** by omitting the coefficient from the first term. This was a trivial mistake and may not affect how the user would plan a reading. If either **A** or **B** were held as mental representations by the user, they would be a good framework by which to guide the reading. Three participants did not recover a superscript from the earcon and chose answer **C**, a more severe error. Such a mental representation, if it were held, would provide a good guide to syntactic complexity, but not an exact guide to the nature of the expression. These were

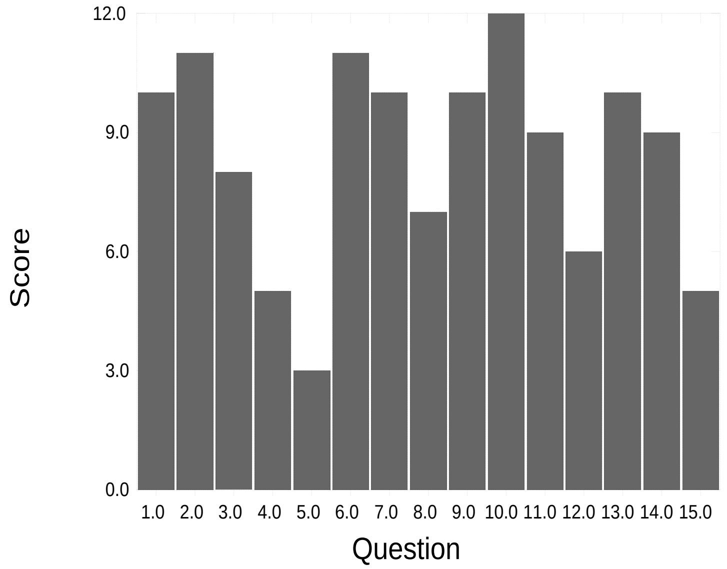


Figure 5.6: Frequency of correct answers for each question for the complex stimuli expressions.

examples of the common error of missing an object or group of objects, but to pick an expression of the right form. Alternative **D** was never chosen, probably because of the equals symbol. Participants commented that the relational operator was a very useful discriminant in the glance.

Question five, of the complex condition, (Figure 5.4(b)) is the only Question in which the correct answer was not the most frequent response. In choosing answer **B**, the error was not to perceive $\frac{a}{b}$, but ab. Participants may have remembered the two sounds, but not their form. It may have been perhaps, that the fraction timbre was not distinctive enough. Many participants complained that these sounds were too fast, and even if recognised as a simple fraction, the internal structure was not noted.

The choice of **C** is a timing error. A gap is perceived between the fraction and the sub-expression. Mistaking products as sums, and vice-versa, was a common timing fault in other earcons. Not choosing **D** is an example of how strong a cue the relational operator is for choosing an expression. This was reflected throughout the experiment.

The distractors were put into categories as to how they differed from the stimulus expression. Large numbers of errors falling into any one category would indicate a problem in the design of the earcon. Then the distractors were separated into those that were *unpicked* and those that were *picked*. The unpicked distractors have some interest in that they will show strong features of the design.

In the simple condition 46 errors were made and these were distributed amongst 21 of the 45 distractors. This clustering of incorrect answers on some of the distractors indicates the validity of the multiple choice design in highlighting design flaws. For these *picked* expressions five broad categories emerged:

1. Timing Errors. There were 11 timing errors where a sum was chosen in preference to a product, or *vice versa*. Seven of these errors involved the combination of two simple fractions into one (Questions six and thirteen).
2. Superscript errors. Six errors were made with regards to superscripts appearing in the expressions. Five of these were the omission of a superscript from the end of an earcon.
3. Omission errors. A total of 20 errors were made where objects other than superscripts were omitted. Ten of these were the omission of terminal objects.
4. Timbre errors. These errors cause distractors with different types of object from the original to be picked. These could be caused by participants making an incorrect mapping between musical sound and object type, or by not being able to discriminate between timbres.
5. Relational operator errors. In Question eight, $a = bc + d$, five participants translocated the equals sign and the plus symbol to give $a + bc = d$. In one other case (Question 9) the equals sign was not remembered (see omission errors above).

The *unpicked* expressions offer another view on the important features of the algebra earcon or which parts of the algebra earcon did work. The distractors that were never picked by a listener must have differed from the true expression, as represented by the algebra earcon, in some respect that made it certain to the listener that it was not the correct answer. Even contradictory expressions may indicate some interesting features of earcon design.

In the simple condition, 23 of the 45 distractor expressions were never picked by any of the twelve participants. That approximately equals 50% of the distractors were never picked should be examined for any revelations that could be gained on the utility or design of algebra earcons. These expressions were extracted into the following categories:

1. Relational operator distractors. Eight of the 23 *unpicked* expressions involved some alteration to the representation of the equality operator. The prevalence of relational operator distractors in the unpicked class is in contrast to the few found amongst the picked. This probably reflects the importance of the cue and its distinct nature.
2. Reflection of expression distractors. The two expressions that involved a simple reflection of the expression (Q1 and Q9) were never picked. This indicates the power of the relational operator to give a clue to the overall structure or balance of an expression.

3. Omission distractors. Nine unpicked distractors involved the omission of objects. Short expressions are the most common in this class, with longer expressions causing more omission errors. Thus it seems likely that only short expressions (or earcons) will be retained in their entirity.

4. Timbre changes. Four questions, (Q7,a; Q7,b; Q7,c; Q13, b), involved turning fractions to ordinary operands (pianos). However, in other expressions these timbres were confused. That no other types appear in this category suggests that not being able to discriminate timbres was a frequent problem.

Omission errors formed the largest category of errors. There are memory limits to how many objects or groups of objects that listeners can maintain after hearing them. Algebra earcons with more sounds or groups of sounds are likely to be remembered less well, for example Question nine. Also, more complex questions, for example those with fractions (Question 13), may cause listeners to devote more resources to maintain or decode the complex parts, causing other objects not to be remembered. Expressions with fewer objects did not have distractors with omission errors chosen. This would support the suggestion that the majority of omission errors occur when the number of sounds were 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, but the glance needs to be rapid. In addition, the glance need not be fully correct. That one term is missed from the start of Question 9 would not seriously impair the use of the representation held by the user as a glance. Similarly, reducing the first term of Question 14 from abc^d to ab^c still gives the listener an impression of the expression as a whole.

Other errors may be easier to resolve from a technical point. Two types of object, fractions and superscripts, cause 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. When a superscript appears on the final object in an expression the violin sound used in the algebra earcon has its pitch decreased. This may have made it more difficult to discriminate from other sounds. 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.

Another factor causing problems was the timing or length of pauses between objects in the earcon. Errors due to the representation of fractions account for most of these errors. The fact no distractors with timing errors were *unpicked* supports the finding that timing was a problem. It is difficult to know whether it is only the timing structure within or between fractions that cause the problems.

Participants complained that the sounds were too fast. Increasing the pause between terms may make recovery of structure easier, but also increases the length of the algebra earcon. Listeners may learn to cope with faster algebra earcons, just as listeners learn to comprehend fast synthetic speech.

Errors due to mistakes with the relational operator were rare and a large number of the distractors with altered relational operators remained unpicked. However, most of the participants complained that the marimba timbre used to represent the relational operators was difficult to pick out from the other sounds. Such an important cue should be as easy as possible for listeners to apprehend and the representation should be changed.

In the complex condition 54 errors were distributed amongst 26 of the 45 distractors. For the complex condition the picked distractors fall into the following categories:

1. Error in scope of complex objects. A complex object is represented by a single, long note of continuous pitch. This representation gives only the relative length, its type and none of the contents are described. 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 superscripts.

2. Timing errors. Fourteen timing errors were made, 11 of which were in turning sums to products. This may suggest that the gap between objects was too short.

   Five more timing errors were made in interpreting the length of complex items in the algebra earcons. In Question three, three of the participants chose option c, which has a denominator shortened to the length of the numerator. This suggests some listeners have trouble in perceiving the relative size of complex objects. That the object is long is apprehended, but not exactly how long.

3. Superscript errors. There were eight errors due to the omission of a superscript. Seven of these were terminal superscripts. Five extra errors with superscripts were made in Question eight and are included in the scope errors above.

4. Relational operator errors. Only three errors, on separate distractors, were made. Two were in the translocation of binary and relational operators, the third was the transformation of an equality term into a superscript.

5. Timbre errors. There were twelve timbre errors which involved transformation to or from fractions.

6. Omission errors. There were two other omission errors. The complex expression earcons often contain fewer objects and this may account for the reduced number of omission errors.

In the complex condition 19 of the 45 (42%) distractors were never chosen by the 12 participants. They fall into the following categories:

1. Relational operator distractors.
2. Scope change distractors. Eight of these were never picked. Five of these options have a complex object transformed into a number of simple objects. Four questions are of the form $(a+b)^{c+d}$. In the distractors, two sounds are changed to at least three sounds, with different timing and timbres. Such changes should be obvious and if the audio glance is to work at all, such options should not be chosen. That three of these four options are in one question makes such a conclusion less valid.
3. Four timbre change distractors were not picked.
4. Only one of the timing change distractors was never picked. In Q3,d, the longer denominator and shorter numerator are inverted. This suggests the great difference in length is a strong cue.
5. Omission distractors. Three options with other omissions were not picked. All these options involved omissions at the start or mid-portion of the expression.

Many of the errors shown in the complex condition are of the same nature as those found in the simple condition. Timing errors are again prominent. That so many were in 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. However, increasing the overall length of the earcon by making the inter-term pauses longer would be undesirable, as the glance is supposed to be rapid. Instead it should be investigated if practice affects the number of timing errors.

In the complex condition the representation of superscripts, and both simple and complex fractions seem to cause problems. There may be a tendency for superscripts at the end of an algebra earcon to be missed more readily than those earlier in the expression. Errors with a terminal superscript were also found in the prosody experiment of Chapter 3. That such errors cannot be reliably resolved with the algebra earcon was disappointing.

It may be that the high-pitched violin sound was ‘too weak’ or short to be readily recognised and retained. All timbres were played at the same volume setting on the mixer for this experiment. It may be better to increase the volume of the violin timbre.

A similar solution may also be used for the pan-pipe timbre used to indicate fractions. Most of the timbre errors involved fractions. Addressing the problems of timing and timbre perception in fractions could resolve a large number of errors in perception of algebra earcons.

Few other omission errors occurred in this condition, compared to the simple. This may be because while many of the expressions represented were complex, the algebra earcons might only have a few sounds, making it easier for the participants to retain. There may also have been a practice effect. There were also few errors involving relational operators, but many participants complained that the timbre used was not easy to discriminate from the others.

5.7.6 Conclusions

This first experiment on algebra earcons demonstrated that a remarkably high proportion (approximately 73%) of questions were answered correctly. 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. A new timbre for relational operators had to be found. The representation of simple fractions had to be improved. These two changes would give the greatest enhancement to the earcons. Some changes were also needed for terminal superscripts to make them easier to recognise.

The two other major sources of error were the recovery of grouping information and loss of algebraic items. The loss of objects from the earcons is likely to remain a problem, unless the design is radically altered to reduce the number of simple object sounds. Perception of timing structure may improve with practice.

5.8 Evaluation of Algebra Earcons: Experiment Two

5.8.1 Design

The first experiment only tested recognition of a printed expression after presentation of the audio glance. This experiment indicated that algebra earcons could present enough information for this recognition task to be performed successfully. For the audio glance to be useful, the listener must be able to recover information from the algebra earcon, form an internal representation and recall that representation during the reading process. This second experiment was designed to investigate the recall of expression structure from the audio glance and to retest the amended rules for generating algebra earcons.

The same experimental design was used in this experiment, but with one difference. Before the participant was handed the printed sheet with the multiple choice expressions, he or she was asked to describe an expression that would be represented by the algebra earcon just heard. This recall task should give a good indication of the level and nature of the internal representation gained by the listener from the audio glance. The repetition of the recognition task after the recall stage should still be useful in revealing faults in algebra earcon design. The recall stage should also be able to feed into this design process, as listeners' descriptions may reveal faults in the algebra earcon design not covered by the distractors in the multiple choice.

Participants

Six of the twelve participants used in the previous experiment were retested. These participants' familiarity with the concept of algebra earcons allowed practice and learning to be taken into account. The three best and the three poorest performers on the first experiment were chosen.

Materials

The same materials were used from the first experiment. The algebra earcons were remade according to the recommendations arising from the first experiment. A three month delay between the first and second experiment was deemed long enough that participants would not remember details of individual expressions, but would remember the concepts behind the audio glance.

The following changes were made to the algebra earcons:

- The marimba timbre used for relational operators was replaced by the more prominent 'rim-shot' percussion timbre.
- The representation of simple fractions was significantly changed. A one beat pause was added between numerator and denominator to 'spread out' the presentation and make it slower. Only the first operand of the numerator was stressed and the last note of the numerator was no longer lengthened. These changes were designed to make the fraction sound more cohesive and similar to that of the single term rather than two separate terms. The pan-pipe timbre was played at a higher volume than other timbres to prevent its being masked.
- All superscripts were played at the same pitch, rather than altering the pitch depending on that of the base. The relative loudness of the violin timbre was increased to make it more prominent.
- All sub-expressions were played at the same pitch to help relieve any confusion with the representation of complex fractions.

Despite the large number of timing errors in the first experiment the timing structure of the earcons was not altered. It was hoped that familiarity with the concept and form of the sounds would enable participants to deal more easily with this part of the representation. If algebra earcons could be kept as short as possible, the rapidity of the glance could be maintained. If, however, the timing errors persisted the design would be altered.

Procedure

A similar procedure was used in this experiment as for the first. The only difference was that after the presentation of the algebra earcon the participant was asked to describe an expression that could be represented by the algebra earcon just heard. The experimenter asked any supplementary questions needed to elucidate the description given. The questions and descriptions were recorded on tape and later transcribed. Otherwise the training and experimental procedure were identical.

5.8.2 Results and Discussion

Table 5.3 shows the scores for the participants in the second experiment. The raw scores for this experiment may be seen in Appendix C.2. The results showed participants still achieved a good score, despite the interference of the recall task. Five of the six participants showed an improvement over their previous score, the overall score of the sixth decreasing by three, causing the improvement to become non-significant. The two top participants again exhibited a ceiling effect achieving 88% correct answers. The participants at the bottom of the range showed the greatest improvement.

A repeated measures T-test (T(5)=1.45 p=0.21) indicated a non-significant improvement in score between the two experiments. A long gap was left between the experiments, so any improvement between the two experiments should only have been due to alterations in the rules for earcon construction and knowledge of the concepts of the algebra earcons. The top scores were already very high, perhaps indicating a ceiling beyond which no improvement could be expected. A larger improvement was seen for the three low scoring participants from the first experiment, though this was not tested for statistical significance.

A detailed examination of errors show that the changes carried forward from the first experiment reduced the number of errors, but several categories of errors still remained relatively frequent. That not all the error categories were addressed between the two experiments perhaps accounts for the non-significant improvement.

Participant	Simple	Complex	Total
E1	10	10	20
E11	10	11	21
E5	11	10	21
E4	12	10	22
E7	14	12	26
E8	13	14	27
Participant mean	11.67	11.17	
Across Participant Variance	2.67	2.56	
Question Mean	4.67	4.47	
Across Question variance	2.38	2.55	

Table 5.3: Separate and combined scores for the second experiment.

Experiment One	**Experiment Two**
Simple fractions	Timing patterns not apprehended
Indistinct relational operator timbre	Objects lost from earcons with many notes
Timing patterns not apprehended	Loss of terminal superscripts
Objects lost from earcons with many notes	
Loss of terminal superscripts	

Table 5.4: Summary of major sources of errors in the multiple choice parts of experiment one and two in evaluation of algebra earcons. The sources of error are listed in order of decreasing impact on performance.

Analysis of Errors

Figures 5.7 and 5.8 show the frequency of correct answer for each question. Many of the same stimuli still caused problems, but the proportion of errors decreased. On others less severe errors were made. For example, choosing an alternative that differed from the correct answer by omitting one object. Such mistakes would not affect the quality of the glance. This was supported by the recall data discussed below. Table 5.4 shows the major sources of error in the two experiments. The high-impact errors found in experiment one were eliminated by the changes implemented before experiment two.

Performance for the two example questions (Figure 5.4) showed a marked improvement. For Question 5 half answered correctly, as opposed to only one quarter in Experiment one. Only one participant did not distinguish the simple fraction from a product, indicating that the representation of simple fractions worked much better. Two participants made timing errors, making the left hand side a sum, rather than a product.

For Question 15, a majority (4/6) answered correctly. The error of missing an initial term did not occur, perhaps indicating better retention of the information present. One error is a trivial missing of a coefficient from the first term. The inclusion of the equals symbol in one answer is unusual and

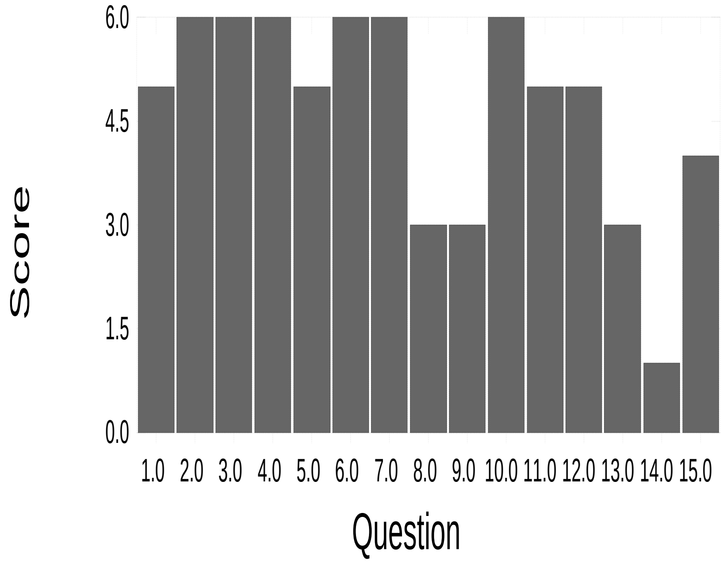


Figure 5.7: Frequency of correct answers for each question for the simple stimuli expressions.

has no apparent explanation as relational operators were sometimes missed, but hardly ever erroneously included.

The distractors were divided into *picked* and *unpicked* options in the same way as in Experiment one. Only half the participants took part in this experiment, so the numbers of errors must be viewed in this light. A total of 20 errors were made in the simple condition. The categories for the simple condition picked distractors are as follows:

1. Timing errors. Only one timing error occurred. This was choosing the option $a + b = c$ instead of $ab = c$ in Question one. There were eleven errors of this type in experiment one simple condition. It is reasonable to assume that in the simple condition the number of timing errors had decreased. However this improvement was entirely due to fewer mistakes being made on simple fractions.

2. Superscript errors. Three superscript errors, each of a different type, occurred in this condition. There were six superscript errors in experiment one simple condition, indicating that these errors were as prevalent in the second experiment.

3. Relational operators. Three errors were made involving relational operators.

4. Omission error distractors were chosen thirteen times on six expressions. Terminal objects were more commonly lost and omission errors occurred more frequently in earcons with many sounds.

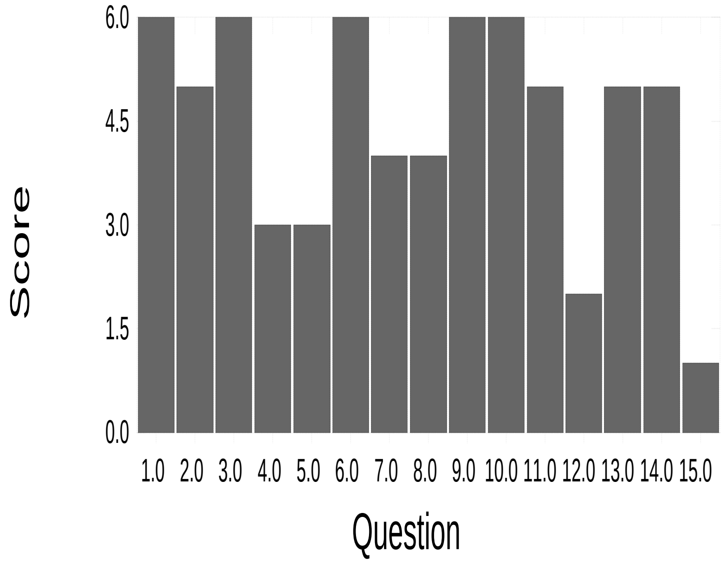


Figure 5.8: Frequency of correct answers for each question for the simple stimuli expressions.

Thirty three options in 45 distractors were never picked in the simple condition of the second experiment. This is 10 more than the corresponding condition in the first experiment. This may be due to the smaller number of participants, but may also reflect improvement in the algebra earcons, making it less likely that some distractors were 'attractive' options.

1. The two options in which the balance of the expression around the relational operator was reflected were not picked. This was consistent with experiment one.

2. Ten omission options were not picked. Eight of these options had initial or mid-expression objects missing. Only two options had final objects removed. This supports the finding that objects from the end of the earcons were more readily lost. Broadly, these were the same unpicked questions as found in experiment one.

3. Superscript options. One option, with a missing terminal superscript was not chosen. This was a very short expression ab^c. Three options with missing initial or middle superscript were not chosen. The number of superscript options *unpicked* had increased by one from the first experiment. Loss of the terminal superscript still remained an error despite the alterations to the earcon design.

4. Four timing options were never chosen. Two, had a product transformed to sum and vice versa. Both of these short expressions were chosen with these faults once each in experiment one. Two options where two fractions are combined into one were not chosen, in contrast to Experiment one. Timing errors did not appear in the *unpicked* category in Experiment one.

This may support the finding that timing perception improved in the simple condition and that the construction of simple fractions was better.

5. Relational operator options. Nine options with an altered relational operator were not picked. This situation is equivalent to Experiment one.

6. A similar pattern of options involving timbre change remained *unpicked* as found in Experiment one.

Omissions still cause the largest number of errors. Two questions (nine and fourteen) cause the two largest contributions to this number. As argued before, missing one object from a large expression should not matter in the context of a glance.

There were three superscript errors in this condition and six in the corresponding condition of the earlier experiment. With half the number of participants in this experiment this does not indicate a decrease. However, not all the errors are classed as being terminal superscripts. There was only an increase in one of the number of unpicked superscript errors. This slight improvement suggests that the representation of superscripts still remains to be improved.

Only one timbre error is reported in this condition, against two in the first experiment. No conclusions may be drawn from timbre errors in this condition.

The number of timing errors in the simple condition reduced from eleven to one, and four distractors appeared amongst the unpicked options, where none appeared before. This suggests that the timing information, though unchanged directly, was being perceived or interpreted better in the second experiment. In the first experiment seven of the timing errors were accounted for by the combination of two simple fractions. The change to the fractions, rather than a direct effect on pauses, probably accounts for the change in the number of timing errors.

The low number of errors involving relational operators continues in this experiment. However, all participants commented that the new rim-shot percussion timbre, which replaced the marimba, was much easier to discriminate from the other sounds. There was some comment that the rim-shot sound was sustained too long and caused some confusion. The sustaining of the sound over the next notes may have caused it to be misplaced in the expression.

In the complex condition there were 23 errors and these fall into the following categories:

1. There were six timing errors. Four were of sums to products. Four errors, two in Question four and two in Question five, were mistakes with the object prior to a complex object: Either making a coefficient to the sub-expression into sum plus sub-expression or *vice versa*. The number of timing errors increases to seven if the inversion of long numerator and short denominator in Expression seven is included. Fourteen timing errors were found in the

complex condition of experiment one, suggesting there has been no change. This may be due to simple fractions, which improved performance in the simple condition, not forming such a large factor in this condition.

2. Scope errors occurred seven times, given the numbers of participants, proportionally similar to experiment one. The nature of the scope errors seems to be different from experiment one, where most were accounted for by superscript errors. These seem to have reduced leaving timbre errors and misapprehension of number of objects.

3. Three other timbre errors were recorded. All involved fractions. In Question two a single sub-expression was transformed to a fraction and in Question 15 (a) a simple fraction was transformed to a sub-expression.

4. Superscript errors. Five errors were made with the omission of a terminal superscript. This happened once in Question 13 and 14 respectively. In Question 15, three participants removed the terminal superscript. All these expressions were long. Three superscript errors were also included in the scope error category.

5. Only one other omission error occurred, with the removal of a sub-expression from the start of Expression 15.

6. Unusually option c from Question seven was chosen. This was the only time a reflection of the true expression was chosen.

In the complex condition of experiment two, 29 options from 45 distractors were never chosen. These unpicked options fall into the following categories:

1. Timbre errors. Five options were never chosen. Most involved short earcons with few notes. This makes it difficult to draw any conclusions, but mistakes in discriminating timbres seem to be rare, suggesting all the musical sounds used are suitable.

2. Ten scope options were not picked. All involve a change in timbre and the number of objects represented.

3. Three options involving timing were not picked. Two involve changing the length of complex objects in the algebra earcon. One such expression occurred as an error.

4. Only three other options with omission errors were unpicked. All three of these options involve the removal of the initial object.

5. Five options with changed representation of relational operators were never picked.

6. Two options with the representation of a superscript near the beginning were never picked.

The improvement in the number of timing errors in the second condition is not as marked as in the first. The number is slightly reduced in the errors and slightly increased in the unpicked options. In this condition the situation of two adjacent simple fractions does not arise and this accounted for the large decrease in the first condition. Most of the timing errors may be accounted for by the placement of coefficients and a complex object. Terms are separated by a minimum of a single silent beat and coefficient and complex object are also separated by a single silent beat. This small separation of terms may be confused with the juxtaposition of coefficient and complex object. Overall it would seem that the number of errors due to the interpretation or perception of gaps between object has not decreased over the two experiments. Thus the number of silent beats between terms will be increased from one to two. In addition, no gap will be used between any object and subsequent complex object (except superscripts following complex objects). This should make the difference between juxtaposition of simple object and complex object compared to the separation between terms more obvious.

There is little insight to be gained from the timbre errors. The unpicked options suggest that the cello for sub-expressions and violin for superscripts were easily discriminated, especially when occurring together. The errors suggest that some confusion is caused by the pan-pipe timbre for the fractions, but these are infrequent. The choice of timbres is limited by the technology. The Yamaha DP110 is a relatively old synthesiser and a more modern one may enable the use of timbres that are more easily discriminated.

Excluding the superscript errors, omission errors were rare in the complex condition. Initial object omissions were amongst the unpicked and many omissions seem to occur towards the end of an earcon. Despite the increase in loudness of the superscript sound and the consistent pitch throughout the expression, many superscript errors still occur compared to other types of error.

It is difficult to judge the reason for these errors. The high-pitched violin sound may be difficult to discriminate from amongst the other sounds. Like the general omission errors many seem to occur towards the end of the expression, suggesting rehearsal of earlier material may preclude apprehension of later violin sounds. The timbre itself may be at fault, making it more difficult to perceive. A new timbre should be sought, perhaps played at a lower pitch and with increased length to make superscripts more prominent.

The scope errors were similar to those in Experiment one. The same superscript errors occurred in this experiment. Options that increase scope seem unlikely to be picked, but some scope increase errors were picked. However, after removing the superscript errors, scope errors are relatively unlikely to occur. This indicates that algebra earcons are able to present the gross structure of complex expressions with ease.

Relational operator errors remain consistently rare, once an equals symbol is added. Once the whole expression was reflected, indicating a gross error in the interpretation of the algebra earcon. Such a misapprehension might cause problems for a reader as he or she started reading, but such errors were rare in this experiment. The participants' comments that the new timbre for the relational operator was much improved were reiterated in this condition.

Recall Reports

The recall part of the experiment was recorded in an attempt to probe the types of representation the listener recovered from the audio glance. By investigating what type of representation of an expression a listener can recall some idea of the quality or usefulness of the audio glance can be obtained. Whilst the distractors can reveal weaknesses in earcon design, some gaps in the participant's representation of an expression may not be shown. Common faults in a listener's representation may reveal further weaknesses in the design of the audio glance.

From the recordings of expressions recalled the following categories of representation were made:

1. A full account of the expression: All the objects described have their classes, locations and relative sizes in place. This representation, however, need not be correct.
2. There is a knowledge of presence and location of most objects and an idea of their grouping. The general shape of the expression was given by the participant.
3. An idea of the general structure of the expression was given by the participant. The location of some objects were given, whilst others may have been missing. Some participants simply gave a list of objects.
4. A simple classification of stimuli into an expression or equation. the balance of left hand right hand sides of the equation around the relational operator may have been included. A few descriptions contained some object categories.

These four categories give a spectrum of representations from an exact framework for the expression down to a simple idea of the length of the expression. These were not discrete categories, the representations given form a continuum.

It was not easy to determine which expressions or algebra earcons fall into which category. As will be seen from the examples below, a combination of factors seemed to be at work. Short simple expressions had a good representation, as did expressions that gave short simple earcons. That is, an algebra earcon with fewer sounds was more likely to yield a good representation than one with a large number of sounds. Another factor could be speed. An earcon with predominantly short sounds will appear fast and overwhelm the listener perhaps leading to loss of information from the internal representation.

In the quotations from the recall data dialogue appears as conventional dialogue in text. Where the participant speaks an expression, variables appear in *emphasised* typeface. Other events within the dialogue appear within ⟨angle brackets⟩.

Level One

Some of the stimuli were usually remembered correctly, these were either simple expressions or complex expressions that produced simple algebra earcons. For others in this class a full account of the expression was given, but some feature was incorrect. Often the timing information was not correctly extracted, or some small feature was omitted. In this level algebra earcons are first analysed across participants where the majority of the protocols reveal a full account of the expression. Exceptions to this class are also discussed at this point.

Taking the simple condition first: For Expression 1, $ab = c$ five of the six participants recalled the correct expression, participant E11 misinterpreted the timing and recalled $a + b = c$ without pause. Similarly all six participants recalled Expression two ab^c and Expression 10 $ab + cd = e$ correctly.

Expression three, $ab + cd$ was recalled correctly by five of the six participants. E4 simply stated '*ab* plus *cd*' with no hesitation. E11 recalled it as two fractions. This is a mistake in recognition of timbres, the two adjacent notes for ab have a similar form to a simple fraction. This mistake was resolved when none of the responses contained fractions. This highlights a problem with the multiple choice paradigm and the usefulness of the recall data.

In expression four of the simple condition, $a^b + c^d = e^f$, all recalled expressions were basically correct, with two mistakes occurring. One left-hand side was recalled as a product. Five out of the six participants did not recall the terminal superscript, but gave full accounts of the expression. However, the response sheet did not have an option containing an expression with terminal superscript missing. The correct expression could be chosen with only the left and side and these participants were prompted to recall the terminal superscript by the printed expressions.

Expression eight, $a = bc + d$ is remembered correctly, but with one interesting mistake occurring. All items are remembered, but the binary operator (represented by a pause) and the relational operator are transposed to give $a + bc = d$. This also happened with Expression two of the complex condition, and in a few other cases (e.g. Expressions 10 and 11 of the simple condition). The rim-shot timbre representing the relational operator is sustained after the note is turned off, so overlaps with the following note. This may cause confusion or hide the next note. A new timbre will be selected in which this does not occur.

Some of the expressions in the complex condition also show very good recall when the nature of the expression means the algebra earcon only contains a few notes. All six participants recall that

Expression one, $(a+b)(a-b)$, contained two sub-expressions. However, two (E7and E4) had then, as a sum, not a product. A pause is inserted between the sub-expression notes, to ensure they are distinguishable, but this has been interpreted as a printed operator.

Expressions two and three were also well remembered. When dealing with Expression two, $a = b(c^d - ef + g)$, participants twice transposed the equals symbol (as described above), but it was recalled well. Expression three, $\frac{ab}{(c+d)(e+f)}$, was recalled correctly by five of the six participants and the difference in lengths of the numerator and denominator commented upon. E8 recalled: '…a fraction, with a longer denominator than numerator.'

Expressions six and ten in the complex condition were of the same form: $(a+b)^{c+d}$. All six participants recalled the expressions correctly. E5 gave a full description for Expression six: 'Okay this is something in a bracket to the power of something and they're both I think approximately the same length. So they have the same number of expressions within them.'

This firm and often correct recall was not restricted to simple expression or short algebra earcons, but was less consistent across participants for other expressions.

In the simple condition for Expression six, $\frac{ab}{c} + \frac{d}{e} = f$, E4 remembered 'Pan pipes, *ab* over *c*, …*d* over *e* equals *f*.' This is correct, down to the two items on the numerator of the first fraction. This suggests that the simple fraction presentation has improved over the first experiment.

For the long polynomial, Expression nine, E7 recalled: 'It's a quadratic and it's got, … think its highest coefficient is four. That could be an equation that matches it.' E7 later stated that 'coefficient' was intended to be a reference to superscript. The participant related not what the expression looked like, but simply gave its type. If this type of information could be regularly recovered from the audio glance, then such a tool would be very powerful in facilitating the reading of algebra notation.

In the complex condition some good representations were also gained from the longer expressions. In Expression five, $\frac{a}{b}(cd+e) = f+g$, E5 recalled: 'Okay, that's something divided by something, two pan pipe noises, definitely an equals sign in there, and there's also a bracket before that, so it's like *a* over *b* could possibly be a space, long bracket, then there's …okay then I'd say it's *a* over *b* plus a bracket equals another letter.' This participant built up a good representation, only missing the terminal operand and misinterpreting the timing to produce a sum and not a product.

For Expression nine, $a(b+c) - d(e-f)^g = h$ E4 recalled: '*a* times expression, plus *b* times expression to the power of something equals something else.' This expression was one of those on which many errors occurred in Chapter 3 due to the mis-interpretation of the cue 'all to the'. In an algebra earcon the marked difference of rhythm and timbres would mean such errors were rare.

E8 also recalled structure well from Expression thirteen: $(a+b)^c(d+e)^f = g$. 'Okay, sub-expression to a power, I think it's times another sub-expression to a power. Equals something. I think it was an equals.' This participant had a perfect representation of the expression.

It was gratifying to note that so many complete and correct representations were recalled by participants. This level of representation was more prevalent with smaller expressions or algebra earcons that have fewer notes. It is in the nature of a glance that the greater the amount of information contained in such a glance then the less full or correct the glance will be. Not all the representations in this level are correct. Many mistakes in grouping are seen. This confirms that the timing structure needs to be made more prominent. However the listeners all gave full accounts of the expressions. These would be the representations used to choose the alternative from the response sheet. The internal representations would be good enough to choose the correct answer in most cases. A firm, but incorrect representation could mislead a listener during reading, but if the user accepts that the algebra earcon is a glance and that it is in the nature of a glance that it cannot be fully relied upon, then this should not be a problem.

There seemed to be a prevalence of missed superscripts, particularly those at the end of an expression. Other single notes were also missed. The other problem revealed here are mistakes in the recognition of timbres.

Level Two

In this level of representation there was a knowledge of presence and location of most objects. The account of the expression is almost complete. There may be parts of the expression, remembered, but unidentified. However, the listener has a general idea of the shape of the expression.

Taking the simple condition first, Expression five, $ax^2 + bx + c = 0$, E8 recalled: 'Okay, two things multiplied together to a power, add two things multiplied together …equals something.' The salient features of the expression are remembered: The first term in full, the balance of the equation, together with the knowledge of extra terms before the equals. For the long polynomial, Expression nine, $ax^4 + bx^3 + cx^2 + dx + e = 0$, E8 said 'It's, two things multiplied together to a power, plus another two things multiplied together to a power plus a few more things with an equals at the end.' This representation is of the same form as for Expression five above. The beginning and end of the expression are recalled, giving its overall shape and length is given by infilling with 'something else'. This is still a good glance; the listener has the general shape of the expression, has an idea of length, balance of the equation.

For Expression thirteen, $\frac{a}{b} + \frac{de}{f} = g + h$, E7 recalled: '…a fraction added to some fancy fraction, equals something plus something.'
'What was the something, what sound?'

'Pianos.' The general form of the expression was recalled. The first term was labeled only as a fraction. It was noted that the second had several items, but no order was given to the objects within the fraction. For Expression nine in the complex condition, E8 said : 'Okay, something times a sub-expression plus something times a sub-expression …, I think there's something else, an equals and something else.' E8 has the basic structure of the expression, knows there are more objects, but had lost them.

This level of representation was relatively sparsely populated compared to the others. The first level captured the bulk of the representations recalled, but some of these may have been better placed in the second, less complete level.

Level Three

This level continued the trend from level two: less structure was given to the objects retained. Sometimes simply a list of significant features was given. In the simple condition For Expression 5, E1 gave: '…All I got out of that was something to a power on the left hand side. There might have been two other terms after that I don't know, because it just seemed to rush through so quickly. And then there was an equals sign and something on the right.'
'how much was on the right?'
'I guess one term, but I was still trying to process the stuff that I'd just got before the equals sign.'
E1 has knowledge of the balance of the expression, the number of terms and the presence of a superscript. Little other detail is present.

E7 gave a typical recall for this level for Expression 7: 'I can't remember, there might have been a drum. It's something equals, fraction. Don't think there was a complicated expression in that one.'

For the long polynomial in the simple condition E11 recalled: 'a b to the c plus blah blah to the whatever plus blah blah to the whatever and then I lost it after that. Oh there was an equals there. Oh things being raised to some power, I couldn't count how many, and there was an equals sign over towards the right hand side. That's all that I've got.'
'What about the nature of the things being raised to a power?'
'I couldn't, I was just trying to grasp a hold of anything that came past me.'
This last example demonstrates the problems with long simple expressions having a large number of simple objects giving a large number of sounds. However the listener has still rapidly gained a potentially useful representation of the expression, even if it is only that it is long.

One of the problems with the experimental design was that the listener was asked to apprehend a full and correct account of the expression. For the final evaluation of the Mathtalk program, the algebra earcon training will have to teach the user how to use the algebra earcons as a glance.

In Expression 4 of the complex condition, E8 recalled: ‘Okay, something …plus a sub-expression, to a power and I can’t remember the rest.’ E8 recalls the significant features of the left hand side, but missed the final term, but suspected something was there.

In Expression 5, E7 recalled: ‘o ’eck. …, equals something. err, and what that something is I can’t really picture. Umm’,
‘can you tell me about the left hand side?’
‘something plus something or something minus something. I think it had a pan pipes in as well and a complicated expression.’ E7 listed all the features of the expression but put little ordering or structuring on the objects.

For Expressions 12 and 15 two participants simply gave lists of the objects recalled, putting little or no structure around these items: E1 recalled ‘Two pianos, one cello, maybe an equals sign, …’

Level Four

The last level of representation indicated by the recall data was minimal information about size and complexity of the expression. At this level information about the presence of a relational operator was recovered and the balance between amount of information on either side of that operator.

For Expression five of the simple condition, $ax^2 + bx + c = 0$, E11 recalled: ‘Dada dada da du …err.’
‘What sounds did you hear?’
‘Damn I’m down on instruments as well dada duda ⟨tapping⟩ da da.’
‘You don’t necessarily have to say the instruments, you can say what they mean, just use any old labels.’
‘*a b c d* ⟨tapping in time to earcon⟩ …is it *ab* plus *cd* equals …. ⟨tapping⟩ ab plus cd something ⟨irritable tapping⟩ this is a hellish experiment. That’s as far as I can go I think. Don’t know what the da da means at the end.’

In this recall E11 had an idea of there being several groups of objects, but could put little form upon them. The superscript and relational operator were not recovered. At several points E11 hummed the algebra earcon (a common feature of this part of the experiment), but failed to interpret or recall the sounds’ labels, an example of timbre error.

This earcon had many short notes and seemed to have overwhelmed the listener. However some knowledge of the number of items seems to have been recovered. Such information should be enough to help the listener choose how to read an expression.

E11, like other participants, mentioned that the experiment was hard work mentally. Whilst the performance was good during the experiment, that the algebra earcons require a large amount of mental resources could severely decrease their usability. This may be partly due to the task being complete recovery of the expression's representation, rather than simply a glance. Usage of the glance in the final evaluation of the Mathtalk program will reveal if this impinges on usability.

For Expression 14 of the simple condition, E4 recalled: '…*abc* plus I spent so long thinking about the first bit that I've forgotten about the rest.'
'Have you got any idea of quantity or type?'
'I think there were about three or four items, I think there was an equals towards the end.'
Again, this participant could recall enough information to get an impression of the amount of information that had to be read. This expression caused problems for all participants. The first term has three operands followed by a superscript. The large number of sounds seem to overwhelm the listeners.

Level four representations were also found among the complex stimuli: For Expression five, E11 recalled '…I've no idea.'
'Can you tell me anything about the …'
'errr no not really. Err I know there's an equals in there near the end. And an *a* plus *b* …maybe *a* plus *b* multiplied by something in brackets ….'

For Expression 8, E1 recalled only the highest level information about the expression's form: 'There was an equals sign at the beginning. Which says there's more on the right hand side, than there has been previously. But I fancy there was a pan pipe as well. Again, I can't give much more.'

Many of the expressions in this lowest level are the long and complex expressions. The representations tend to be at a high-level sometimes containing only knowledge of amount of material, but often some type of objects and that the expression is an equation and the balance of that expression. Such representations can still be classed as glances. Such knowledge should be able to guide the listener in selecting a strategy with which to read an expression, even if only at the level of choosing between a full utterance and an unfolding style. Such a representation would not allow the listener to search for an expression, unless they were only looking for one particular feature to pick out of the earcon.

All of these representations could be useful as a glance because they would indicate the syntactic complexity of an equation. However, a strong, but inaccurate framework has the potential to mislead a reader. As algebra earcons were only designed to provide a glance, such inaccuracies would not be too great a problem because any glance is not supposed to be entirely accurate. A good representation of the equation would be a bonus for the reader. The task forced subjects to recover as much information as possible from the earcon and meant that participants were probably not using the earcons simply as a glance.

Recovering information from the glance may be a difficult task, as described by some participants, but this may be exacerbated by the novelty of the audio glance and the artificial nature of the experiment. In addition, the difficulty of using the audio glance has to be balanced against having to use a full utterance to guide the reading process.

These data, taken with the results of the multiple choice part, indicate that algebra earcons can work as a glance. The presence, type, location and size of objects can be conveyed to the listener. Some of the changes resulting from the first experiment improved the presentation, especially that of the simple fractions and relational operator timbre. However, recovery of timing structure remained a problem and design amendments were proposed to help resolve this problem.

Omissions remained a large class of errors. Numerous amongst these were omission of superscripts. Again a design solution was proposed. Omission errors are always likely to remain, especially from earcons with large numbers of sounds. Such errors are well catered for in the concept of a glance. The ability to recognise complex groups from the use of hidden objects in the earcon, indicates that the glance may help in reducing grouping ambiguity.

5.9 Rapidity of the Glance

These two experiments have shown that algebra earcons have fulfilled most of the requirements for the glance at the structure of an algebra expression. These were the type, location and size of objects in an expression. It was argued that these were the salient features of an expression that needed to be conveyed to the reader to enhance active reading. The last criterion to be fulfilled was that of rapidity. The glance at structure needs to be much quicker than simply listening to the expression in full and retrieving the structure. To show how much faster the algebra earcon glance was than the spoken equivalent, the following expressions were timed for both the earconic and spoken presentation. The results are shown in Table 5.5. The default speech speed of 180 words per minute was used in this comparison; the same rate used in the evaluations in Chapter 4.

Expression	Speech	Glance	Proportion
1	2.42	0.82	0.34
2	4.07	1.32	0.32
3	7.03	2.03	0.29
4	8.13	2.69	0.33
5	9.56	1.98	0.21
6	10.38	2.31	0.22
7	7.31	1.65	0.22
Total	48.9	12.8	1.9
Mean	6.98	1.83	0.28

Table 5.5: Times in seconds for the spoken and algebra earcon presentations of expressions. Also shows glance time as a proportion of spoken time.

1. $x = y$
2. $ab + c = d$
3. $ax^2 + bx + c = 0$
4. $\frac{1}{2}(ax + b) = c + d$
5. $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
6. $3(a + b) - 9(c - d)^2 = 0$
7. y=3(x-8(x-4))
8. $y = 3(x + 7)^{x+4}$

This comparison of spoken and earconic presentations show, that on average, the earcons take 27% of the time of the spoken equivalent. Thus, in relative terms, the algebra earcon provides a rapid glance. Compared to a visual glance, a glance of several seconds is not fast. An interesting avenue of research would be to find how fast both speech and algebra earcons can be played and still recover a useful amount of information.

5.10 Exploiting the Utility of Algebra Earcons

There were opportunities to exploit the association of musical sound with objects within the expression to enhance the usability of the interface. As discussed in Section 4.3.3 the terminus earcons that indicate the start and end of a level in an expression can easily indicate the type of level simply by using the musical timbre associated with that level.

At present, when a reader reaches the beginning or end of a level or the whole expression, two general *terminus* sounds are used to aid orientation and navigation for the reader. The evaluation of the command language revealed that these sounds were very useful during browsing, preventing some basic orientation errors. However all users found the overloading of the 'end of level ' sound to mean the end of any level or end of expression to be confusing. A sound indicating the end of a sub-expression could be mistaken for the end of the expression or a user could not remember the terminus of which particular type of environment had been reached.

A family of earcons has been developed to solve these problems. Instead of having a different sound or earcon for the start and end of each syntactic type, one basic sound is used and the parameter of timbre used to indicate which syntactic type is being browsed. For example, using the Sub-expression timbre for the end of level earcon only when the reader is browsing a sub-expression should facilitate orientation within the expression and thus aid navigation. When the end of any particular level is the end of the expression, the terminus sound is repeated so that the reader knows when there is no more information to read.

Ambient sound (Gerth 1992) could also be used to aid navigation and further capitalise on the consistency of sound within the interface. (Brewster, Wright, and Edwards) (1994b) use ambient sound to indicate the relative position of the current page during scrolling through a simple text editor. These sounds were successful in helping users to maintain a sense of position while performing tasks within the editor.

This use of sound can be extended to the browsing interface in Mathtalk. For example, as the reader enters a sub-expression, the cello timbre is switched on as an ambient sound (Gerth 1992). The onset of such background sounds is noticed by the listener, but fade into the background (Buxton, Gaver, and Bly 1991). The listener can then sample such sounds to determine the current environment, and again notice the switching off of the sound as he or she leaves the current environment.

The onset of the cello sound on entry to a sub-expression reinforces the move the user has made. As the reader browses through the sub-expression the cello sound is quiet enough to fade into the background of consciousness, unless the reader consciously pays attention to confirm current orientation. As the reader leaves the environment, the offset of the sound can be noted, further reinforcing the browsing move.

The use of ambient sound could be further elaborated. When one complex structure is nested within another multiple ambient sounds could be used to indicate the depth and nature of the nesting. The current environment could be made prominent and the higher levels faded further into the background. Care would have to be taken to ensure the sounds do not become unpleasant, intrusive or overwhelming. At present these sounds have been designed for the Mathtalk program, but not implemented and most importantly not evaluated.

5.11 Summary and Conclusions

This chapter has described the development of an audio glance at algebra notation. The audio glance called algebra earcons gives the listening reader the opportunity to gain such a glance. The principles guiding the formation of the glance were:

- The salient features of a glance for structure based reading are the presence, location and size of the objects within the expression.
- The glance must be more rapid than a simple spoken alternative.
- The prosodic content of speech can indicate structure. This capability can be re-used in the design of an audio glance.
- The non-speech audio form called earcons were used to provide prosody without the speech. Earcons and prosody are described by the same parameters and this fact can be used to guide the design of non-speech audio messages in the computer interface.
- Rules for algebraic prosody became the rules for algebra earcons: The replacement of spoken objects by musical timbres can give prosody without the speech.
- Hidden objects were realised in the algebra earcons to hide complexity, a general property of a glance.
- The association of musical sounds with structural type can be exploited throughout the interface. Extra information can be added to the terminus sounds to show which structural environment has been terminated, rather simply that something has been terminated.

Two experiments were performed to explore the ability of algebra earcons to give a glance. The results supported the ability of the audio glance to convey high-level structural information about an expression. The experiments also provided useful information on flaws in the design of algebra earcons.

These experiments did not show that such an audio glance was useful in a Mathtalk style interface. They only indicated that they could convey the intended messages. The final, full evaluation of the integrated Mathtalk program explored the usefulness of such an audio glance.

Chapter 6

Confirming the Design Principles

6.1 Introduction

In the previous three chapters, the major components of the Mathtalk program have been designed and evaluated. Each component has been shown to make its contribution to addressing the problems presented by control and external memory for a listening reader. Each of these features was then integrated into the Mathtalk program. The object of this final stage in the development of the Mathtalk program was to test if the integrated system does in fact transform the passive listener to the active reader by addressing the problems of external memory and control of information flow. In this chapter, the evaluation of the full Mathtalk program is described.

In the second part of this chapter, a paper design will be presented for the Treetalk program. This design uses the principles developed during this research to build a user interface for reading an auditorily presented phrase structured syntax tree (Lyons 1979). Trees, such as that shown in Figure 6.1, are a common method of presenting linguistic information. Section 6.5 describes the problem that a blind reader would have using a tree and discusses what information such a display contains and therefore should be presented in an equivalent auditory display. A design is presented for the use of prosody, browsing and glancing to enable active reading of grammar trees. This paper design shows how the principles used in the design of the Mathtalk program can be applied to an unrelated information format that requires a blind reader to access complex information auditorily.

6.2 The Nature of the Mathtalk Evaluation

A comparative evaluation has been chosen to demonstrate the usability of the Mathtalk program. It would have been possible to assess the usability of the Mathtalk program in isolation. A similar style of evaluation, to that used for the browsing component, on the Mathtalk program alone, could have demonstrated that integrating the components gave a suitable reading of algebra notation. Each component has been shown to improve the presentation; give the reader control over information flow or allow a glance at algebra notation. Simply showing that these components worked together to allow a suitable reading could have been sufficient to validate the design decisions.

A comparative evaluation increases the value of the exercise. Demonstrating that the Mathtalk program improves the reading of algebra notation over and above that afforded by currently used methods will give a stronger indication of the value of designing for control and external memory.

What the Mathtalk program should be compared to was a difficult decision. Performing algebraic tasks non-visually is accepted to be difficult, so it could be thought that any comparison would succeed in showing an improvement. The current methods used to perform algebraic tasks, using speech only, not tactile aids, are:

Amanuensis A sighted reader speaks an expression and writes down any changes the listener makes as a result of the information given.

Tape recorded speech The user listens to algebra spoken onto tape by a human reader. The basic play, pause, forwards and backwards actions are used to control the information flow.

Word-processor Algebra notation is written in some linear notation such as a programming language style and the user controls the information flow with the cursor controls of the word-processor. The user can also write down any changes made as a result of the information read.

It can be reasoned that the word-processor option gives the best opportunities of the three options above. The word-processor gives some degree of control and an unambiguous presentation of information. The design of the Mathtalk program focuses on these issues and as well as showing if there is an improvement over ‘best practice’, such a comparison would further indicate that designing for control of information flow and external memory is a sound basis for design.

The Mathtalk program was compared to the use of expressions, presented in $\LaTeX$ format in a word-processor accessed using a screenreader and synthetic speech. The survey of secondary level mathematics undertaken as part of the EU Tide project Maths (Cahill and Boormans 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 (Edwards 1993; Stöger 1992). 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 thought 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 Chapter 3. The IBM Screenreader (Thatcher 1994) speaks the expression

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

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, but the control afforded is not wholly appropriate to the reading of algebra. 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 poor control and display compared to the Mathtalk program should highlight the differences between access and usability and show that designing for control and external memory improve the usability of the reading process in the auditory mode.

A modified co-operative style of evaluation was used. 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 was used to assess the participant’s mental workload, preferences and comments on the two systems. Again, the stance of Wright and Monk (Wright and Monk 1989) was adopted. Quantitative data on number of commands, error rates, speed and accuracy of task completion do not tell the whole story of the usability of the interface. The participants’ comments on what they were doing and why were equally as effective at demonstrating the contrasting usability of the systems.

The qualitative data was important as the evaluation did not seek to test the participants' mathematical success. The evaluation sought to judge whether the participants could accomplish the task in the manner they wished, to their own satisfaction.

6.3 The Evaluation of Mathtalk

6.3.1 Design

Two conditions were used in a within participants design: The word-processor condition and the Mathtalk condition. A similar design was used for this final evaluation as was used in Chapter 4. In the previous evaluation, the balance of tasks was towards the navigation and orientation within and without expressions. This time the tasks were skewed towards 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:

- 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.

6.3.2 Changes to the Mathtalk Program

The following changes had been made to the Mathtalk program from that used in the evaluation of the browsing language described in Chapter 4:

- The action **glance** had been added to the list of actions. This action worked on all the structural targets available. The complete list of commands may be seen in Appendix B.2.
- The command changes detailed in Chapter 4 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 synthesiser. This synthesiser 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 Chapter 4 were implemented.
- A mute function was implemented, but found to be unstable and so removed from the interface. Instead, if errors occurred, an error state was held from which the user must recover.

6.3.3 Materials

One set of training expressions and two sets of matched expressions and questions were set for each condition. The training expressions can be seen in the list below. The same expressions were used in both training sessions.

1. $ab + c$
2. $ab + cd = e$
3. $y = ax^2$
4. $y = \frac{1}{2a} + bc$
5. $y = 3(x + 7) + 9$
6. $y = \frac{ax+4}{2a}$
7. $y = x^n + 1$
8. $y = x^{n+1}$
9. $y = (a - b)(a + b)$
10. $y = \frac{1}{2}(x + 4)^2$

Mathtalk Training

The Mathtalk program is obviously extensive and complex. In such a short evaluation it would have been impossible to expect the participants to learn and use all the features and commands.

The training followed the pattern of tasks used in Chapter 4. As the training proceeded the features of the prosodic component were taught as they arose.

The appearance of the terminus sounds during browsing were used to introduce the associations between musical sounds and the objects within an expression. After an initial pass through the list of training expressions, a second pass was used to train on the algebra earcons. The training took about 30 minutes, depending on how the participant reacted and made enquiries. The training was still superficial, given the complexity of the system, and this reinforced the need for the use of the co-operative style of evaluation. The detailed stages of the evaluation are given below (the numbers in parentheses refer to the expression concerned):

1. The concept of a list of expressions was introduced, with each expression being numbered.
2. The command style was taught: Actions and targets with a mnemonic mapping. The first command taught was **current expression** to speak the whole expression (1).
3. Moving between expressions and the circularity of the list were taught. From (1) to (2), back to (1), (10) and back to (2).
4. Introduction of **current next** and **previous** as principal actions. These had already been used with **expression**. The principal targets were completed with **term** and **item**. **Current term**, with **next** and **previous** gave the opportunity to teach the terminus sounds (2).
5. After using the **next** command the participant was at the end of an expression, so the **beginning expression** command was taught (2), followed by **end**.
6. The default browsing style was taught on (2).
7. Superscripts were introduced with expression (3).
8. The concept of an **item** being more than one character was taught using x^2 in (3).
9. Unfolding of fractions taught with expression (4). Also used to introduce fraction timbre. Introduction of the **hidden objects** and the concept of a **level**.
10. Use of expression (5) to introduce **current level** as a glance at the overall structure. A contrast with **current expression** was made. The default browsing was then reinforced by browsing through this complex expression.
11. Prosodic features introduced with contrast between simple and complex superscripts in (7) and (8). Utility of hidden objects in disambiguating grouping pointed out.
12. **Which expression** taught to find number of expressions in the list (7).

13. Moving to expression (9) was used to teach multiple commands.

14. Default browsing through (9) used to teach building up an expression; moving into and out-of complex objects and use of terminus sounds.

15. After moving by default through expression (10) more intricate moves were taught: **out-of quantity**, **show item**, **into item**, **into denominator** and **out-of fraction**.

16. Move back to expression (1) to start training on algebra earcons. Recap of associations of timbres with objects.

17. **Glance expression** used on the simple expression (1). Parallels with the prosodic component were emphasised. The use as a glance rather than a mechanism to extract full structure was also emphasised.

18. Expression two introduces a drum sound as a relational operator.

19. Expression (3) used to introduce the *beep* sound for the superscript.

20. Expression (4) introduces the fraction sound (trombone sound).

21. Expression (5) gives the string sound as the sub-expression or quantity.

22. Other expression used to reinforce associations of sounds and train the participant in use of the glance. Glancing at objects smaller than an expression were not taught in this training.

During the training the simpler moves were reinforced and the participant told that he or she did not need to remember all the commands and could ask the the experimenter at any point for any information. Part of the training was to emphasise that the best strategy was to remember the command words, rather than the commands themselves. Having done this, the participant was told to make up commands from the words.

Word-processor Training

The general features of the LaTeX representation of the algebra were explained in the following order:

1. Expressions appear one per line.

2. Each expression is preceded by a number and full-stop.

3. Most of the expressions are formed by normal keyboard characters.

4. Parentheses are used to group items into sub-expressions.

5. The circumflex character (`^`) is used to denote superscripts.

6. *Simple* superscripts contained only one character and that *complex* superscripts, with more than one character, needed braces to indicate what was in the superscript.

7. Fractions are preceded by `\frac` and numerator and denominator are separately grouped by braces.

8. The use of special symbols such as

   ```
   \pi
   ```

   to represent π was taught as the experiment proceeded, as it was thought the participant would not remember such detailed information.

The word-processor WordPerfect was used in the experiments. All the users were familiar with WordPerfect, but the basic browsing moves were explained. The movement centred around the cursor star 6.4:

- The `up` and `down` keys to move between lines and therefore expressions;
- the `left` and `right` keys to move character by character;
- modification of the `left` and `right` cursor keys with the `control` (ctrl) key to move between words or terms in an expression.
- the `home` and `end` keys to move to the extremes of an individual expression;
- the `page-up` and `page-down` keys to move the cursor to the extremes of a file and therefore the list of expressions.

These controls would not allow the user to read the current line, word (term) and character without moving to and from that object. The screen reader's keys for performing these tasks were taught, along with the mute button. The screen reader's browsing keys were on a separate keypad, placed on the side of the keyboard corresponding to the participant's dominant hand. The participants, all of whom were not familiar with this system, were allowed to practise these moves.

The training proceeded along the following lines:

1. Reading a line with the screenreader keypad. The numbering of expressions was introduced(1).

2. An alternative technique for reading a line by moving to and from that line with the up and down cursor keys was taught (2).

3. Moving in between expressions with up and down cursor keys.
4. Expression 2 introduces the equals sign; Expression 3 introduces ‘circumflex’ that designates a superscript.
5. The participant was told that he or she would have to use the expression number to know a different expression had been encountered.
6. On moving to a new line, the screenreader started to read the whole of that line, so the expression’s number was guaranteed to be spoken.
7. Introduce word-processor commands to move around Expression 2: left and right cursor, then control left and right cursor. The word character was used instead of **item**.
8. Introduce home and end to move to the extremes of lines.
9. Introduce keypad commands for current word and character.
10. Move to Expression 4 and explain format of LaTeX fractions.
11. Move to the beginning of Expression 4; examine each element of the fraction, especially the word ‘`\frac`’. An attempt was made to try and teach the participant to read word-by-word so that ‘backslash frac’ was spoken as one word rather than individual characters.
12. Moving through (5) character by character to examine each element: ‘Left brace’ starts the numerator and ‘right brace’ terminates the numerator.
13. Immediately after the end of the numerator, another ‘left brace’ ends the denominator and then a ‘right brace’ terminates the fraction.
14. Expression 5 introduces the parentheses as groupers. Training here was easier as the hidden objects concept did not have to be taught.
15. Expression 6 introduces complex fractions and reinforces the use of braces to group the terms of the fraction together.
16. Expression 7 iterates the use of ‘circumflex’ to indicate a superscript. The lack of braces was taught to mean that only the single object after the circumflex was the superscript character.
17. Expression 8 was used as contrast with (7) to introduce braces to extend the scope of the circumflex character.
18. Expression 9 iterates the use of parentheses.

19. Expression 10, being complex, enabled a discussion of how to break down the expression to occur. It was explained that either a whole line had to be read or by chunking into words and characters.

20. Movement to and from the end of the file was taught using either the page-up and page-down keys, or simple use of the up and down cursor keys.

21. An overview of the moves was given, reinforcing the layout of the expression into terms by using spaces before operators. The grouping of terms was reintroduced.

The set of commands and possible moves in the word-processor were only a fraction of those available in the Mathtalk program. This made training much simpler and shorter in this condition. Equal emphasis was made in each condition as to how objects were grouped together, despite the contrast in styles of presentation. Given the simpler set of commands it was difficult to balance the two training schemes in terms of time. The same set of expressions were used for both training sessions and this helped to balance the training. However, the aim was to teach the same level of sophistication in the use of each style of presentation. This was made more difficult by the participants being familiar with the word-processor and having used a similar method of working with algebra in their education.

Experimental Condition Materials

Two sets of expressions, matched for complexity, were created for each condition and shuffled to a random order. The two sets of expressions may be seen in Table 6.1. The two sets were matched for structural complexity. Matching was achieved by independent assessment.

The LaTeX code was altered slightly to make sure the word-processor condition was not artificially difficult. Each expression was placed on one line, prefixed with a number followed by a full-stop. LaTeX for a mathematical code is surrounded by dollar signs ($), these were removed, given that the human reader could recognise algebra notation in a manner that a computer cannot. The screenreader spoke words as the default unit of speech. To make the LaTeX code more usable the code was divided into ‘words’ that would reduce the amount of speech given at any one point. The word-processor presentation can be seen below:

```
1. y =\frac {7 -x} {x +7}
2. (x +3)(x -3) =y
3. y =3((x +7) +9x) -5
4. y =19 -3x
5. y =2xˆ2 +3
```

	Condition	
Number	**Mathtalk**	**Word-processor**
1	$y = 7x + 3$	$y = \frac{7-x}{x+7}$
2	$y = (x+3)(x-2)$	$(x+3)(x-3) = y$
3	$y = x^2 + 8$	$y = 3((x+7)+9x) - 5$
4	$y = 3^{x+1} - 7$	$y = 19 - 3x$
5	$y = \frac{x+4}{x+8}$	$y = 2x^2 + 3$
6	$y = \frac{1}{2}(x+5)^2$	$y = 2^{x+1} - 5$
7	$y = x^2 + 4x + 2$	$y = 3x^4 + 5x^3 + 2x^2 + 8x + 4$
8	$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$	$y = \frac{1}{3}(x+5)^2 - 7$
9	$7x^4 + 3x^3 + 7x^2 + 5x + 3$	$y = \frac{x+2}{x+5}(x+1)$
10	$y = \frac{x+3}{(x+5)(x+1)}$	$v = \frac{1}{3}\pi r^2 h$
11	$y = 2(x + 5(x+2)) - 3$	$3x + 2y + 1 = z$
12	$v = \frac{4}{3}\pi r^3$	$p = \pm \frac{lx_1 + my_1 + n}{\sqrt{l^2+m^2}}$

Table 6.1: The expressions used in the Word-processor and Mathtalk conditions of the final evaluation.

```
6. y =2^{x+1} -5
7. y =3x^4 +5x^3 +2x^2 +8x +4
8.y =\frac {1} {3} (x +5)^2 -7
9. y =\frac {x +2} {x +5} (x +1)
10. v =\frac {1 } {3}\pi r^2 h
11. 3x +2y +1 =0
12. p =\pm \frac {lx_1 +my_1 +n} {\sqrt {l^2 +m^2}}
```

A set of questions was devised for the first set of expressions. Once this was finalised, it was re-ordered and adapted to match the expressions in the second set. The questions fell into two parts. The first was a series of navigation and orientation questions devised to assess the user's ability to move around an expression, apprehend structure and maintain orientation. This part also continued the training section of the experiment. The second part of the questions were substitution and evaluation tasks.

Questions for the Mathtalk Condition

The navigation and orientation questions were:

1. Move through Expression one until the end is reached, then move back to the beginning and read the current term.

2. What is the significant feature of Expression four?

3. Read and describe expression Eight.
4. Move to Expression two, explore and describe this expression.
5. Move to Expression six; explore and describe.
6. Find the longest expression in the list.
7. Find the most complex expression in the list.
8. Find the quartic expression necessary and move to the term with x squared.
9. Move to Expression 11 and find the deepest part.
10. Find the denominator in Expression five.

The substitution and evaluation questions are shown below. The question number is the number of the expression into which the given value should be substituted.

11 x = 2

6 x = 3

12 Move to expression 12 and use radius = 3 to find the volume of a sphere with that radius.

7 x=6

10 x = 5 and simplify

1 x = 3

2 x=4

5 x = 4 and simplify

4 x=2

9 x=2

3 x=3

Questions for the word-processor Condition

1. Move through Expression four until the end is reached, then move back to the beginning and read the current term.
2. What is the significant feature of Expression six?

3. Read and describe Expression 12.
4. Move to Expression two, explore and describe this Expression.
5. Move to Expression eight, explore and describe.
6. Find the longest Expression in the list.
7. Find the most complex Expression in the list.
8. Find the quartic and move to the term with x squared.
9. Move to Expression three and find the deepest part.
10. Find the denominator in Expression nine.

The substitution and evaluation questions were:

4 x = 7

1 x = 4 and simplify

11 Find z when x=7 and y = 3

7 x=2

6 x=2

5 x=4

10 This Expression finds the volume of a cone. Find the volume of a cone with radius r=5 and height h=4

8 x = 4

9 x = 5 and simplify

3 x = 2

2 x=5

After each condition a set of questions were used to elicit participants' comments about the style of presentation, ability to move to objects and discriminate one object from another. Questions were also asked about how the participants used each style to perform the mathematical tasks. Finally, the subjective mental workload associated with each condition was assessed with the NASA-TLX described in Chapter 3. A similar scale was used to assess overall preference for the conditions. The questions used are shown below.

1. How good at computing would someone have to be to use the presentation to do the tasks?
2. General presentation of expressions
   - How easily could you tell different parts of the expression apart?
   - Could you tell when fractions, sub-expression and superscripts began and ended?
   - What cues in the presentation began and ended these structures?
   - Could you get a general impression or overview of the expression?
   - How can you gain a general overview?
   - What made finding the shape of the expression easy, if anything?
   - What made finding the shape of the expression difficult, if anything?
3. Navigation and orientation
   - What techniques did you use to move to a new term?
   - How would you move to and speak the numerator of a fraction?
   - How did you move to the start of a sub-expression?
   - How easy was it to notice the end of such a structure?
   - How did you tell what structure it was?
   - How did you note the end of a denominator?
   - Could you use browsing to help disambiguate the structure of an expression?
   - How did this disambiguation work?
   - Did you feel that you became lost in any of the expressions?
   - If so, in what sort of expression did you get lost?
   - How easy was it to choose the portion of an expression to be spoken?
   - Did the browsing on offer allow movement to any part of an expression you wanted?
   - Are there any movements that were particularly difficult or missing?
   - How did you choose which browsing commands to use?
4. Doing the tasks
   - In what ways were some questions more difficult than others?
   - How did you deal with more complex expressions?
   - How did you use browsing to help in evaluating the expressions?
   - What strategies did you use?

- How did you plan your evaluation of an expression?
- How did the presentation help you in the tasks, if at all?
- How did the presentation hinder you in the tasks, if at all?

Equipment

The Mathtalk condition used the Mathtalk program used in Chapter 4. The browsing functions and command language had been amended as described above and in Chapter 4. The algebra earcons had been implemented so that an audio glance could be generated for any expression that could be presented by the Mathtalk program. Again, no visual display was available.

The IBM ScreenReader was used to access the WordPerfect word-processor used to access the LaTeX form of the expressions. This enabled the participants to use the Multi-voice speech synthesiser in both conditions. None of the participants were familiar with this synthesiser, but the quality was such that no training was needed. None of the participants were familiar with either the Mathtalk program or the IBM ScreenReader, but all were familiar with WordPerfect.

Participants

Each of the three components of the Mathtalk program had been evaluated by sighted participants–because the features being assessed would work equally well for sighted as for blind users. However, for this evaluation blind participants were used. The integrated features of the Mathtalk program can only really be tested by the end users themselves.

Four blind participants were used in this evaluation. The participants needed to be not only visually disabled, but computer users and already at a reasonably advanced level of mathematics education. These criteria made finding such participants difficult. However, given the nature of the evaluation, this small number of participants need not present too much of a disadvantage.

Short biographies of the participants used are:

F1 was in the second year of an ‘A-Level’ mathematics course. His preferred method for using algebra was to write a linear notation of his own devising into a word processor. These lines of notation could then be edited and printed out. F1 did most of his work on a portable computer and was thus unused to a standard computer keyboard. F1 said that his method of working was adequate, but naturally was not the most efficient way of performing the tasks. F1 was blind from early childhood, had spent time in special education, but his present course was in a mainstream college.

F2 was on the same ‘A-Level’ mathematics course as F1. His preferred method for performing mathematical tasks was by amanuensis. A sighted person read problems to F2, who then directed the amanuensis to write, and read the equation. F2 said that this was hard work and that it was frustrating. He had used mathematics written in some form of linear notation in a word-processor, but finding this difficult had been trying amanuensis. F2 had been blind from early childhood, had spent time in special education, but the current course was in a mainstream college.

F3 was a first year undergraduate. He had a GCSE mathematics qualification. F3 did not currently use algebra or mathematics in any form. He was, however, a very keen computer user and programmer. At school his method of using algebra was to use a linear notation, in a programming language style, in a word-processor.

F4 had an Open University foundation course in mathematics. This is an equivalent to ‘A-Level’ mathematics. He did not currently use his mathematics. During his recent course he used a linear, programming style notation in a word-processor to perform mathematical tasks. He was an experienced computer user and programmer. F4 was adventitiously blind as an adult.

Procedure

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 training for each condition proceeded as related above. The participant was told he could ask any question about the presentation style or the mathematics. During the mathematical tasks, the experimenter held any intermediate values and would offer help about performing the tasks if necessary. After the mathematical tasks, the questions were asked and the TLX scales marked. Each condition took approximately 90 minutes to run and a 15 minute break was taken between conditions. The speech and non-speech audio were presented to the participants using external loudspeakers.

6.4 Results and Discussion

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. Strong support for this came from the participants’ comments, preference and mental workload ratings.

Mathtalk allows a wider range of views of an algebra expression and these were exploited by the

	Participants				
Condition	F1	F2	F3	F4	mean
Mathtalk	527	239	322	341	285.3
word-processor	642	661	617	549	617.3

Table 6.2: Total number of commands for each participant and means for each condition.

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.

In mathematical terms, little may be said about the effectiveness of the two presentation styles. Whilst there was some evidence of more appropriate views of an expression being used, the mathematical ability of the participants obscured any judgement of the effectiveness of the interfaces in terms of correct answers to the mathematical tasks. Most of the participants had to be coaxed through some of the tasks, some not understanding or remembering the order of precedence for multiplication, exponents and parentheses. It was felt that this lack of ability in the participants obscured some of the usefulness of the features available in the Mathtalk program. In addition, both presentation styles were able to convey all the grouping information and symbol names, so that in both methods the participants had the ability to reach a correct answer. However, with the Mathtalk program, results were achieved more easily.

6.4.1 Commands and Strategies

The frequency of each type of command used in the navigation and evaluation tasks of each condition were collated and recorded in Table D.1 of Appendix D. Table 6.2 shows the total number of commands used in each condition. Each participant used many more commands in the word-processor condition than in the Mathtalk condition. A system failure meant F3's count was not recorded, so the mean of the other participants' keystroke count was substituted and the condition mean taken as the mean of these four scores.

As will be seen below, 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.

Both systems provide access to algebra. The difference comes in how participants used the system and the descriptions below demonstrate that the Mathtalk program gave the more usable access to

F1		F2		F3		F4	
K	**P**	**K**	**P**	**K**	**P**	**K**	**P**
Next-char	0.35	Next-char	0.71	Next-char	0.71	Next-char	0.51
Next-word	0.16	Prev-char	0.14	Next-line	0.09	Prev-line	0.12
Prev-line	0.14	Next-line	0.07	Prev-line	0.09	Next-line	0.11
Next-line	0.12	Prev-line	0.04	Line-start	0.07	Prev-char	0.10
Prev-char	0.14	Line-start	0.03	Prev-word	0.07	Line-start	0.04
Prev-word	0.07	Doc-top	0.01	Prev-line	0.05		
Line-end	0.02	Doc-end	0.00	Next-word	0.04		
Doc-top	0.01	Next-word	0.00	Doc-start	0.01		
Line-end	0.00	Doc-end	0.00				
Total	642		661		617		549

Table 6.3: Proportion of total commands issued for each of the commands used by each participant in the Word-processor condition. **K** = Keystroke and **P** = Proportion.

algebra notation. However some of the representations in Mathtalk caused some problems that led to modifications in the design. Thus, the efficacy of co-operative evaluation in guiding design was seen.

The word-processor Condition

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. Table 6.3 shows the proportion of the total keystrokes used contributed by each command. Most keystrokes are accounted for by only a few commands. For the majority of the time three participants simply read the expression one character at a time with the cursor keys. Only F1 used the ability to move word-by-word throughout the experiment.

The whole expression was sometimes read with the **current line** command, but the resulting output was often silenced. Even when short expressions were spoken in full the participants only gave a description after further browsing of the expression. The most common strategy was to read the expression character-by-character and build up an expression from the components. For example, F2 when asked to describe Expression six listened to the full utterance, but only retried the circumflex. He proceeded to read the expression character-by-character until the number of the next expression was heard:

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

F2 I'll skip through it.

E *Why's that?*

F2 It was too long. I know it's got a circumflex in it.

E *What does that mean?*

F2 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 ....`

F2 y equals two with a power x plus one, minus five.

For the shorter expressions this strategy was adequate. However for the longer, more complex expressions the process became long and error prone. For example, F3 when reading Expression twelve, noted the fraction, but by the end had forgotten the overall structure. This expression was a difficult one, but similar incidents occurred with other complex expressions.

Using only the cursor keys complex objects such as parenthesised 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 Chapter 3 in which little structure or content was reliably recovered. As soon as the participant moved to the target expression, that expression started being spoken in full. On most occasions, if the expression did not conclude within a few terms or complex structures, the user muted the speech with either the mute button or by performing another small move, that as a by-product also muted the speech. For example, F2 described well the reasons for using the mute in this condition and not in the Mathtalk condition:
F2 said 'How do you mute it?'
'On the keypad there are two big buttons, the lower is the mute button. You didn't ask me about mute in the other condition. Is there any particular reason for that?'
'On the first bit of software? 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 line. …you have more control in the last one.'
This view was repeated by other participants. It was interesting to note F2 using exactly the form of words for needing the mute facility that formed the basis of the Mathtalk design. This behaviour

confirms the usability problems described in Chapter 3 and a contrast to what was seen in the Mathtalk condition.

Despite the expressions being laid out so that words acted like terms, movement between terms using the control key to modify actions of the cursor keys to move a word at a time was not extensively used by three of the participants. They claimed that the movement was ‘unreliable’. F4’s movement through Expression four in Question one of the navigation tasks was typical of use of the control key plus left or right cursor. One result of the use of this key was the tendency to wander to different lines and thus different expressions. After this first use of the control key, F4 rarely used it.

Down four period one nine minus three x.

right period, _, one, nine, _, hyphen.

ctrl-right five period y equals 2 x circumflex two hyphen three.

ctrl-right y, equals two x circumflex three.

ctrl-left y, five period.

E *I want you to be at the end of expression four.*

left _, _ (space at end of expression four.)

E *Can you move back to the beginning of expression four?*

right _, _

ctrl-right Five period y equals 2 x circumflex two hyphen three.

E (explanation of keypad keys)

current line Five period y equals 2 x circumflex two hyphen three.

right _, _.

ctrl-right five period y equals two x circumflex two hyphen three.

E *I want you to be at the beginning of expression four now.*

ctrl-left five period.

right period.

up four period one nine hyphen three x.

etc. (Several iterations until F4 arrives at the appropriate location in expression four.)

This underuse may have been due to the larger movements giving a larger amount of speech and the larger movement taking the user deeper in to the next expression by accident. The large amount of speech associated with one word movement, by including words such as ‘left brace’ and ‘right paren’ may have proved difficult. Movement backwards term-by-term was rarely seen. It may be that the greatest level of control over information flow was gained by using the smallest moves possible.

Larger movements within the text were more rare than within the Mathtalk condition. Occasionally the participants moved to the extremes of lines with the home and end keys or to the first and last expression with **page-up** and **page-down**. However moves still centred around the cursor star. The participants adapted a strategy that had interesting parallels with the evaluation in Chapter 4. On leaving an expression or arriving at a new expression the user moved back to the beginning of the line. On enquiry, F3 said this was to make sure he knew where he was in the expression and that it was a strategy he commonly used when working on other tasks, especially programming. Some participants adopted this strategy in Mathtalk in the current experiment, despite being told there was no real need. However the use of this strategy was less prevalent.

One of the major frustrations for the participants in this condition was the 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 one case, F4 moved several expressions away from the target without noticing. This example is shown above.

The Mathtalk Condition

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 Tables 6.4 and 6.3 where the proportion 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, straight to a fraction or quantity and showing that object as one item. An interesting contrast with the prior evaluation of the browsing language was the greater use of the **current expression** command to speak whole expressions. Before, the verbal glance of **current level** had been used in preference to this command. The audio glance may account for some increased use of the **current expression** command. Re-use of previously learnt strategies may be another factor. These will be discussed further below. Renderings of the whole expression were rarely used in the word-processor condition. That they were used in the Mathtalk condition may

F1		F2		F3		F4	
C	**P**	**C**	**P**	**C**	**P**	**C**	**P**
Default	0.19	Default	0.47	ne	0.15	Default	0.17
ge	0.14	ne	0.13	ge	0.12	cl	0.15
ce	0.09	ce	0.12	nt	0.08	ge	0.13
ni	0.09	ge	0.07	ce	0.07	ne	0.11
ne	0.08	Multiple	0.06	cl	0.06	Multiple	0.05
Multiple	0.07	we	0.05	be	0.05	sq	0.04
cl	0.06	be	0.03	ni	0.05	sf	0.04
Errors	0.06	cl	0.03	Multiple	0.04	we	0.04
pe	0.04	pe	0.02	we	0.03	ce	0.04
be	0.04	ct	0.01	Errors	0.03	Errors	0.04
Total Commands	257		239		322		341

Table 6.4: Proportion 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.

well be due to the improved spoken presentation due to the addition of prosody. Whilst heavy use of the full utterance was not expected in the Mathtalk condition, it does demonstrate the increased usability due to the use of prosody.

Muting of full utterances was frequent in the word-processor condition, but was requested only once in the Mathtalk condition. The example shown in the previous section indicated that F2 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. Whilst the non-implementation of a mute in Mathtalk will ultimately need to be rectified, it was obviously not a problem for the users.

The error rate in this condition was very low with only 0.3% of commands issued being erroneous. In the word-processor condition no mistakes were made in issuing commands. There was little scope for this sort of errors with the word-processor, but a far larger one in the Mathtalk condition.

F2 issued no incorrect commands during the navigation and evaluation tasks. More errors occurred during training, but these were not recorded. This low error rate and the wide range of commands used by three of the participants highlights the learnability of the browsing language and the efficacy of the training. F4 made the only significant mis-perception of the language during any of the evaluations. After having read an object and wishing to repeat the utterance, this participant issued the command **previous next**. This may have been an attempt to move back to the object just spoken: ‘the previous next object I moved to’. In the training the construction of commands from one action and one target was emphasised and the location of the focus of attention on last object spoken was reinforced and this seemed to remove the problem.

F3 made several errors by reversing the order of letters in the command. The largest number of errors were due to **into** and **show** commands being used with an inappropriate focus of attention. The clash between the **speak** (now **show**) and the **current** action were not observed. The change of command and removal of the inconsistency described in Chapter 4 seem to have been successful. Despite the general learnability of the browsing language the workload involved in recalling a command appropriate for a certain situation meant that users often made inefficient use of the browsing language. However the examples throughout this section point towards emerging strategies and it is hoped that further exposure to the system would see an increase in this trend. The word-processor presentation was used to its full extent and found lacking, whereas the Mathtalk presentation was found, in general, to be more usable, but with room for development.

The rapidity of the issuing of commands was a minor problem. Simply using the cursor keys in the word-processor was very fast, even if the reading itself was no faster (see below). F1 in particular had difficulties. Whilst an experienced computer user, F1 had only used a keyboard on a portable computer and had great difficulties in using the desktop computer style keyboard. The relative complexity of the browsing language and the relatively large amount of motor action involved made the issuing of commands in the Mathtalk program slower for all participants.

The style of usage of the browsing commands varied between the participants. Some common features were present. All used the facility to multiply the **next** and **previous** actions to move around the expression list. All used the glance in the navigation condition and a mixture of **current expression** and **current level** to gain views of the whole expression. Another general feature was the reliance on term-by-term reading, with three of the participants making heavy use of the default strategy and F3 using **next term** to accomplish the same end. For example, F4 unfolded Expression seven and simply substituted the appropriate value:

Current expression `y equals x super two plus four x plus two`.

Space `y`.

Space `equals x super two`.

F4 What was x?

E *six*.

F4 So six squared is thirty six.

Space `plus four x`.

F4 Plus twenty four. Plus thirty six is sixty.

Space `plus two` <end sound>

F4 That's the end, plus two, sixty two.

The quartic was usually evaluated term-by-term, with occasional browsing of the items. For example F3 used the **next term** command, combined with movement item by item. F3 was asked why he tended to use only the left and right cursor in the word-processor presentation and term-by-term browsing in Mathtalk; he replied:

> 'I don't know really, it's just the nature of the program I think. It's more user friendly for a start, you can just move straight to the terms, I think it was a bit unreliable in word-perfect.'

F2 used the narrowest range of commands. He used a general sequence of gaining an overall view of an expression with either **current expression** or **current level**; then used the default browsing to unfold the expression term-by-term, simply moving back to the beginning of the expression to re-read any sections. Whilst F2 had learnt the form of the commands his difficulty with mathematics and using the new browsing language at the same time may have decreased the usability of the system for him, although he was much more keen on the Mathtalk program.

For the expression with nested sub-expressions, F2 like the others, became confused by the Mathtalk representation. F2 used **current level** and **current expression** to gain overview of the expression. He then used the default strategy to move through the expression. Every time he became confused he simply used **beginning expression** to start all over again. F2 had difficulties in both representations with nested structures. However, as will be examined later, the Mathtalk representation of nested objects seems to be one major fault in its style of presentation.

It seemed that F2 simply re-adopted his strategies of the word-processor condition, presumably those he used in everyday working. The only difference was the term-by-term working within the tasks reducing the amount of material he had to move through. Even this change in granularity seems to have offered an increase in usability for this participant. F2's reluctance to use other commands that could have made the tasks easier may be explained by this obvious unfamiliarity with some of the expression forms and this workload precluded any extensive use of commands. In the post-condition questions he asked if it was possible to move to the previous term as he thought this was missing. The task of recalling actions and targets from a short training will be hard for some people and this means that extensive use of the command set will take time to emerge.

It was interesting to note that all participants tended to move to the start of an expression either at the start or end of a task, despite Mathtalk always starting from that position. As described above, the participants said this was to maintain orientation. This supports the idea of strategy transfer and also indicates that orientation within the information space is generally a difficult task for which users have developed strategies.

F1made more extensive use of the commands available. However, at first F1 made greater use of moving character-to-character with **next item** than all other participants. F1 was the participant who made least use of this method in the word-processor. This finest level of control may have been a lack of confidence in either remembering or coping with larger amounts of information. F1 goes on to use the default strategy and other commands to move to and speak complex objects as a whole.

Towards the end of the evaluation tasks another strategy emerged. F1started using a repeated full utterance when the length of the expression permitted. For example in Expression six, F1 uses a glance to look at the expression then simply uses two full utterances to evaluate the expression:

Current expression `y equals one over two times x plus five super 2 plus five`.

E *x equals three.*

F1 Five squared is twenty five, half is twelve and a half; so 12.5 minus 5.

Current expression `y equals one over two times x plus five super 2 plus five`.

F1 Right, so what'd you say x was?

E *Three.*

F1 seven, eight.

Current expression

Current expression `y equals one over two times x plus five super 2 plus five.`

F1 sixty four, thirty two, thirty seven.

F1's difficulty with the keyboard may account for his unwillingness to make greater use of a wider range of commands. However he did perform the mathematical tasks as effectively as the others and did use more extensive browsing moves, such as moving to and speaking complex objects, in larger and complex expressions. That the presentation enabled F1 to use a full utterance was probably due to the pauses within the utterance separating objects within the expression affording time to capture objects and ignore other output. The example above was an example of this ability to use the full utterance effectively.

F1 was the only participant to use the earcons during the evaluation and substitution tasks. All had used them spontaneously during the reading tasks. In combination with the full utterance they may have been used to determine the complexity of the expression, create expectations and then choose either the full utterance or the unfolding strategy. Like participant F2, F1 often preferred moving back to the start of an expression and then unfolding again, rather than using other commands for more local browsing. This may have been direct transfer of strategy from their own word-processor style of working or a way of avoiding the overheads of remembering and using all the Mathtalk commands.

F3 and F4 used the most extensive set of commands in performing the tasks. The difference between these participants and particularly F2 was the extensive use of the actions **show** and **current** to speak the contents of complex objects as a whole and the use of **next** and **previous** with complex targets to move directly to objects of interest in the expression. It is this set of commands that separates the functionality of the Mathtalk program from the word-processor and makes it more effective. The order of precedence means that complex objects need to be evaluated or otherwise dealt with before the simple terms of an expression. Being able to move straight to the objects that have to be dealt with first and treat them as a single object should make the user of the Mathtalk program more effective than the user of a word-processor reading character-by-character. This effective use of commands also makes the user of Mathtalk more efficient in terms of the number of commands used.

F4's evaluation of

$$y = (x+3)(x-2)$$

proceeded in the following way:

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

Next quantity `a quantity`.

Show quantity `x plus three`.

F4 is eight.

Next item `a quantity`. <end sound>

Show quantity `x minus two`.

F4 times two is Sixteen.

Such a strategy was very effective: The verbal glance told the user the salient part of the expression was two sub-expressions; one command took the user straight to the first sub-expression (rather than having to move through each item); one command revealed the contents, which were calculated and the process repeated on the second quantity to yield two numbers that gave the answer.

Similar strategies were seen with other expressions, for example F3on Expression six:

$$y = \frac{1}{2}(x+3)^2 + 5$$

a sequence of **next fraction**; **show fraction**; **next quantity** and **show quantity** allowed F3 to perform the bulk of the calculation with very few moves. A final **next term** rounded off the calculation.

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 evaluation tasks the audio glance was only used on four occasions, by participant F1.

The basic training for the algebra earcons worked and during the navigation tasks all participants were able to give suitable descriptions of expressions. Using the terminal sounds to associate musical timbre with type worked well. The adaptations to the terminus sounds proposed in Chapter 4 improved the usability of these sounds. All participants were able to tell which part of an expression had started or finished and the special end of expression sound ensured that objects were not missed from expressions due to misconceptions of expression form. The examples throughout this section contain comments from users noticing the ends of objects and the expression itself. The inability to move past the end of an expression or internal object and become mixed with the next improved the usability of the presentation.

In the navigation tasks, a typical sequence of events would be for the participant to glance at the expression; give a high-level description; then speak the whole expression and then browse the expression in more detail. The question was usually answered appropriately with only the algebra earcon, but all participants usually went on to explore the expression in more detail with no prompting from the experimenter. For example, F4 explored Expression six in the following way:

Glance expression <algebra earcon>

F4 It’s got a superscript.

E *What else did it tell you?…was it short or long?*

F4 Medium. It has a fraction, an equals and the beginning was very short.

Current level `y equals a fraction times a quantity super two plus five`.

F4 There's a quantity as well with a superscript.

Next fraction `a fraction`.

Show fraction `one over two`.

Next quantity `a quantity super two`.

Show quantity `x plus five`.

Show quantity `x plus five`.

Current level `y equals a fraction times a quantity super two plus five`

Using the glance F3 could give the following description of the main features in Expression eight: 'A fraction, with longer numerator than denominator, the bit on top is much larger.' For Expression seven F1 gave the following account from the glance: 'That's something equals a bit of a fraction, times a quantity, to the power of something …probably a simple term, plus–add another term.'

This apparent willingness to explore the expression and gain a series of views was in contrast to the word-processor condition, where the minimum set of moves to answer the question were used.

Two participants exclusively used the algebra earcons in performing navigation tasks six and seven. A sequence of **glance expression** and **next expression** commands were used to move through the list, glance and give judgements on the expressions. The glance was also used to find the quartic by one participant, who simply searched for the correct pattern of sounds. F3 moved through the list listening for superscript sounds and then looking at those expressions in more detail:

> 'First I found an expression with a superscript, used the glance to check what the sound was, then looked through the rest for those sounds and checked them.'

That the algebra earcons were readily used by the participants and seemed to help in accomplishing tasks, after short training, indicated their intrinsic usability and usefulness in accomplishing tasks. Some of the answers to navigation tasks six and seven indicate some problems with the audio glance. In tasks six and seven, many people judged complexity by the number of sounds present in the earcon, rather than the length and type of some of those sounds. So an expression such as

$$y = \frac{x+4}{x+8}$$

would have fewer sounds than

$$v = \frac{4}{3}\pi r^3$$

which would be judged to be the more complex of the two. Though the concept of hidden objects seemed to be well understood and useful in other respects, the participants lost a lot of information from the glance by not using the length information. The use of length information was not emphasised in the training and this was obviously a mistake.

The nature of the navigation and evaluation tasks may account for the discrepancy in the use of the algebra earcons in the two conditions. Describing an expression should take much less mental workload than evaluating that expression. The increased mental workload in the evaluation tasks (particularly as the participants found the mathematics difficult) probably meant that moves not vital to the goal were dropped from the strategy. Most participants stuck to a rigid left-to-right evaluation strategy, where knowing the overall structure was not so important. Those that did use the presence of complex objects to plan their evaluation strategy seemed to use the **current level** glance to determine their presence. So the overheads involved in using the algebra earcons to glance and misconceptions about the information content may account for the discrepancy in usage. Longer term studies would have to be performed to find whether use of algebra earcons increased as all the features of the Mathtalk program become overlearnt.

Nested structures caused the participants problems in both conditions, but particularly when using Mathtalk. Three of the participants managed to describe the expression as ‘nested’ during the navigation tasks, but failed to take this into account during the evaluation. Perhaps the difficulty of the task interfered with the use of the unfamiliar system and exacerbated the problems.

F4, F1 and F2 became lost during the evaluation, causing long calculation times and ineffective management of the task. (F2 and F4 also had problems in the word-processor condition, but not so profound). The following example shows the general conceptual difficulty encountered: F2 on reading Expression two in the word-processor condition used the right cursor key and heard: `y, _, equals, three, right paren, right paren`. He commented ‘what does right paren right paren mean?’ This was after encountering the similar structure in Expression eleven in the Mathtalk condition.

F1 managed well with the word-processor condition, simply cursoring through and substituting the given value, retaining the information and calculating the correct answer. This was not the case in the Mathtalk condition. For example, F3 performed in the following way:

Current expression `y equals two times x plus five times x plus two minus three`.

Next quantity `equals two times a quantity`.

Next quantity `a quantity`.

Into quantity <start sound> `x`.

Current level `the quantity x plus five times a quantity`.

F3 Right x plus five, what was x?

E *Two*.

F3 So that's two times five, ten.

Next quantity `a quantity.` <end sound>

Into quantity <start sound> `x`.

Current level `the quantity x plus two`.

F3 Right, plus seven.

Out-of quantity `a quantity`. <end sound>

Into quantity <start sound> `x`.

F3 I think I've gone into this one before.

Current level *the quantity, x plus two*.

F3 Yes, Ah, I've gone into that one before.

Out-of quantity `a quantity`. <end sound>

Out-of quantity `a quantity`.

Next term `minus three`. <end expression sound>

F3 Equals whatever it was.

F3 became confused about location. A better strategy would have been to use **show quantity** on the inner sub-expression. It is this type of situation that the ambient sounds described in Section 5.10 were designed to benefit. F3 also makes mathematical mistakes, perhaps due to his strict left-to-right method of evaluation. F4 made similar errors, but with worse consequences. After using the default strategy to move to the inner sub-expression he became confused about his location in the whole expression. He wanted to be told when he was on the outer quantity, that it contained a nested sub-expression (he would have had to use an extra move, **show quantity**, to find this out). He also became confused by the output of **current level**. This was prefixed with `the quantity`, which F4 took to be another quantity and tried to move to that object.

One contributing factor to this difficulty in Mathtalk may have been the expressions themselves: the LATEX expression started with `y =3((`, so that simply by cursoring the participant would know that there were two open parentheses and thus one must be nested within the other. This also meant that the innermost could be calculated first, without any complex browsing moves. In contrast, the corresponding Mathtalk expression had the nested sub-expression at the end of the first sub-expression: $(2x+5(x+2))$, so that the most effective way of evaluating the expression meant moving out of sequence from the default reading strategy, which was preferred by most participants. The need to work out which browsing move to use, with an unfamiliar and complex system, probably added to the difficulties. However, the main factor must have been the presentation of the expression by Mathtalk. Neither the algebra earcon nor the **current level** command could indicate the more complex structure of the expression. The full utterance would be one that may have stretched the ability of prosody to indicate such complex structure, particularly to novice listeners (it should be noted that the full utterance was not used in the word-processor condition). The **show quantity** command which gave the output `two x plus five times a quantity` seemed to help in the navigation tasks but not in the evaluation tasks.

The confusion caused by the prefix `the quantity` can easily be solved. The use of the terminus sounds will be extended to the **current** commands acting upon complex objects. This removes the prefixes and makes the command consistent with others in the language. The word ‘quantity’ seemed unsatisfactory to the participants and will be substituted with ‘group’ as described in Section 3.4.

6.4.2 Timing

The times were taken for completion of the tasks in the navigation and evaluation tasks. Times were taken from a tape recording of the session. The co-operative style of the evaluation made taking clean measures of task completion time difficult. This demonstrated a difficulty in combining qualitative and quantitative forms of evaluation. As far as was possible, extensive dialogue between the participant and experimenter was omitted from the timing. A stop clock was started from the first move the user made in completing the task. Sometimes this was made before the task statement was completed; usually after the number of the expression had been given. The clock was stopped when an answer was given to the task. The raw times for each task for each participant can be seen in Tables D.3 to D.6 of Appendix D.

Completion of the navigation tasks was difficult to ascertain, particularly in the Mathtalk condition. A task such as ‘describe the general shape of the expression’ was often answered with the glance (earconic or from current level command). Usually the reader then went on to explore the expression more fully with extensive use of the browsing language, building up more and more

	Participants				
Condition	F1	F2	F3	F4	Mean
Mathtalk	46.8	93.1	45.4	89.8	68.8
word-processor	85.9	139.9	63.1	73.9	90.7

Table 6.5: Participant and overall mean time in seconds for the navigation tasks for each condition.

detail. This apparent willingness to explore the expression in a variety of views demonstrates the usability of the Mathtalk system. In both conditions the time was taken when a full answer to the question had been given. The algebra earcons often resulted in a partial description or one that did not match the description given in the word-processor condition.

A paired sample two-tailed T-test was performed on the mean times for the navigation tasks. Summary values for the navigation task times can be seen in Table 6.5. A non-significant difference was found between the task completion times (t=-1.56; df=3; p= 0.11). Whilst the overall times were faster in the Mathtalk condition, the fact that one participant was faster overall in the word-processor condition means, that with only four participants, the overall difference would be non-significant. However, 72% of the Mathtalk condition times were faster.

Some of the navigation task times were highly variable. For instance the three tasks:

6 Find the longest expression in the list.

7 Find the most complex expression in the list.

8 Find the quartic.

provoked different responses, some of which indicated a poor ordering of the questions. Having answered Question six, some participants answered 7 from memory, while others went through the list again, looking at each expression. Having done this, some participants had a good idea of the contents of the list, remembered the location of the quartic and moved straight to that expression.

The time differences taken to complete the evaluation tasks were also non-significant (t= -0.3113; df=3; p=0.39). Table 6.6 shows that the means were much closer, with two participants completing the tasks faster in the word-processor condition. This time 60% of the tasks were performed more rapidly in the Mathtalk condition. Given the difficulties the participants experienced with the mathematical tasks, the similarity in times may not be surprising. A large amount of time spent thinking about how to mathematically complete the task may have confounded any difference that may have been seen between the conditions. A further confounding factor may have been the participants' prior familiarity with the word-processor being used and the more familiar form of presentation.

	Condition	
Participant	Mathtalk	word-processor
F1	99.00	85.82
F2	105.09	109.64
F3	66.82	79.91
F4	99.18	101.55
Mean	92.52	94.23

Table 6.6: Participant and overall means for the evaluation task times (in seconds) for the two conditions.

Whilst the increased efficiency and effectiveness of the Mathtalk program would be more convincing if task completion times were significantly faster, some positive points emerge from this analysis. Mathtalk has a more complex interface. It was encouraging that with minimal training participants perform no slower than the more familiar word-processor presentation. Taken with the apparent willingness of the participants to increase exploration and take a variety of views of an expression, and the trend towards increased speed it may be said that the Mathtalk program has the potential to become an efficient form for accessing algebra notation. As F1 states: ‘I can see that once the commands have been learnt, this could be a very, very fast way to read expressions.’

6.4.3 Post Condition Questions

The post-condition questions were in three parts: On the presentation; on the browsing; and on the evaluation tasks. These subjective comments revealed a strong preference for the Mathtalk condition, with many of the participants noting the features that were designed to increase the usability of the Mathtalk style of reading algebra notation. A marked feature of the responses was that participants noted that all the tasks were possible in the word-processor condition, but that the Mathtalk program made them easier to achieve.

In the questions on general presentation style the participants concentrated their comments on the browsing aspects rather than the overall presentation. Some general comments about the overall presentation were obtained and these supported the observations of the command usage. In the word-processor condition the full utterance was thought to give too much information. F4 described it as ‘clutter’ and F3 as presenting ‘too much at once’. F4 in particular was scathing about the word-processor condition describing it as ‘rubbish’ and on starting the navigation tasks saying: ‘if you’re asking is this as good as the other, then it isn’t.’ F1 was the most in favour of the word-processor system, saying it was most like what he used. He did however make some negative comments: ‘Straight off it was quite difficult. I was having to deal with the difference between braces and parentheses, once I’d got used to that, and the voice and speed then I’d find it quite easy’.

When asked if he could tell when objects began and ended F1 said: ‘Not initially, not during a full utterance, but moving through it was quite straight forward …provided you concentrated on what you were doing it should be pretty straight forward.’ This reflected a general feeling that the full utterance of an expression written in LaTeX was hard work, but that it did contain all the necessary information.

F2 felt that the algebra was no more difficult in the word-processor condition, but that it was ‘more difficult to work it.’ Later when talking about the whole expression, F2, said ‘in the other one [Mathtalk], it gives, up and down, the voice went up and down, where this stayed stationary all the way through.’ In contrast to his view of LaTeX F4 thought that the expressions were ‘nicely laid out’ in the Mathtalk presentation. These views confirm the observation in command usage that the prosodic presentation helped the participants.

In the word-processor condition participants felt that they could only tell parts of an expression apart during browsing. F3 related: ‘Fairly difficult, because you only have a set of basic commands, you can only go forward or back a determiner and left and right to a determiner.... it’d be a lot simpler if you could simply have the equation laid out in various chunks so you could go to the appropriate bits.’ Laying out the expression and supplying the commands to move easily to certain portions was part of Mathtalk’s design and the user’s comments supported the implementation of these features.

All users recognised that the braces and parentheses delimited complex objects in the word-processor style, but that it was sometimes difficult to judge exactly what objects these were. F2 said, ‘It did tell you what things were but it was difficult’. F1 said it was difficult to get an overall reference point for location within an expression. F1 was used to working with maths in this style, but said he divided up an expression so that each chunk was on a separate line. This enabled him to treat each unit as a separate object. This is analogous to how Mathtalk enables complex objects to be treated as single objects. However Mathtalk has the advantage of still being able to see the hidden object in its context.

For the word-processor condition the participants mentioned that using a full utterance as an overview was difficult. The method used to gain the overview was to cursor through the expression and build up the shape. F4 said he had to move backwards and forwards a lot to find out where he was in the expression. F1 got the overview ‘by moving to the expression, reading it char-by-char and comparing it to what you heard. You know how far you could listen before you got completely bushed.’ This corroborates the observation that the speaking of the line was often deliberately muted before the end was reached. F2 said ‘having the cues there in the speech meant that it was possible to get an overview, but the way the word-processor spoke it made it difficult.’

Three methods were mentioned as giving an overview in Mathtalk. The first was the audio glance, then a full utterance and the **current level** command was used. Several people mentioned that they liked the glance. F3 said he could use it to tell: 'whether it was long or not and to distinguish when certain things, when fractions, powers etc were ....' F1 gave a similar view, but felt that it was not quite what he wanted. For both the spoken hidden objects and the algebra earcon, he wanted a more detailed view, but not a full utterance.

F2 said he did not remember the associations of the timbres during the experiment. He did, however, give many full descriptions of expression just from the audio glance. F3's view of the glance was as follows: 'The glance is handy, because it tells you what sort of expression is there, but in a very brief sort of way, but that's what you want. Then you get an idea of what's like then you can go into it in more detail.'

These observations by the participants reflect the original design of the algebra earcons: To give a glance at the overall structure of an expression. This glance was seen to be useful when simply reading an expression, but not when the expression was to be evaluated. It may be that a variety of levels of glance are necessary for all styles of working with algebra. The spoken glance with **current level** was widely used during the evaluation and this may have given a more appropriate glance.

Browsing was seen as essential in both conditions. It was used both for disambiguation and breaking the expression down into smaller chunks. F1 was the only one to talk about term-to-term movement in the word-processor condition. All others described their strategies as simply moving character by character until they found what they wanted. F2 said: 'You could get anywhere eventually.' F3 went into more detail about the restrictions of the word-processor presentation: 'The fact you couldn't go to each individual part of the expression off the top of your head made it difficult, you had to navigate through all the rest of it in order to reach the bit you wanted.'

All participants liked the design of the browsing language with F4 stating: 'It's quite easy, the navigation, it's self-explanatory.' Both F1 and F3 said that they were hindered to some extent by not remembering all the commands as soon as they wanted to use them. Two of the participants commented that they thought the language was easy to learn. F3 said that reading the expression was 'relatively easy, because of the commands. The command structure made it very simple to navigate the expressions.' All mentioned that being very accustomed to the style of browsing in the word-processor was an advantage and made the word-processor condition easier to manage. This was supported by F1's comment: 'It would be more difficult because you have to remember specific commands rather than just skipping from quantity to quantity. The command set is straightforward though.' This disadvantage of Mathtalk has to be weighed against the frustration expressed by the participants having to move through all parts of the expression in the word-processor condition.

F2 liked the browsing within Mathtalk, particularly because it would reduce large things to single phrases like ‘a fraction’. All participants except F1 mentioned the default strategy as part of how they used the browsing. F4 and F3 mentioned the use of **show**, **current** and **next** as being very useful as part of the browsing and a distinct advantage over the word-processor condition.

F1 was the only participant to note that the ability to read up to or from the current position was not possible in the Mathtalk program. This will be added to the system. F3 wanted the ability to move directly to the beginning and end of the list of expressions. Otherwise all participants thought they could make all the moves they wanted. All participants said they found the navigation and orientation easier within Mathtalk, except in the case of nested structures. This aspect of the Mathtalk design was explored above.

6.4.4 Mental Workload

The NASA TLX subjective mental workload assessment gave another view on the usability of the two presentation styles. Reducing the mental workload over current practice was an essential goal in the development of usable access to algebra notation. Table 6.7 shows the summary scores for the TLX factors; the raw data may be seen in Tables D.1 and D.8 of Appendix D. Paired T-tests were used to assess the significance of any differences between ratings for the factors. The raw mental workload was calculated as described in Chapter 3. The overall mental workload was found to be significantly lower in the Mathtalk condition (t=-2.9; df=4; p=0.04) with a a mean of 5.5 for Mathtalk and 10.2 for the word-processor. This view was confirmed by many of the comments made during the evaluation.

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. This reduction in the mental workload further confirms the design for external memory and control of information flow.

The significant preference for the Mathtalk condition also supports the participants’ satisfaction with the system. The overall preference scores (Appendix D Table D.1) were adjusted so that the bias from the mid-point ‘no-preference’ point matched the actual conditions. The mean expressed preference score was 16, where 10 was no preference, 20 was totally favouring Mathtalk and 0 totally favouring the word-processor. Three of the four scores were at the extreme (17, 17, and 20). F1 indicated ‘no preference’, because he was so used to the word-processor style of working.

The identical mean scores for the perceived performance levels reflects the presence of the required information in each presentation style and the participant’s recognition of this fact. The means were 12.25 for both conditions. There was also no significant difference in the time pressure felt by the participants during the tasks (t=-0.5; df=3; p=0.63) with a mean of 6.5 for Mathtalk and 8.25 for the word-processor.

Factor	Word-processor	Mathtalk	Difference	% Difference
Mental Demand	14.8	7.0	7.8	210.7
Time Pressure	8.3	6.5	1.8	39.0
Effort Expended	9.8	3.5	6.3	30.0
Perceived Performance	12.3	12.3	0.0	0.0
Frustration Experienced	10.5	2.8	7.8	39.0

Table 6.7: Summary of TLX scores for each factor in the TLX for the final evaluation. For all factors, except perceived performance level, a low score is positive in terms of usability.

This reflects the control the participants had over the information flow in both conditions. The time pressure was felt to be slightly higher in the word-processor condition. One participant described this condition as ‘frantic’.

The mental demand was significantly lower in the Mathtalk condition (t=-7.52; df=3; p=0.005), with a mean of 7 in the Mathtalk condition and 14.75 in the word-processor condition.

The effort expended just failed to reach significance despite means of 3.5 for Mathtalk and 9.75 for the word-processor (t=-2.9; df=3; p=0.06). This factor mixes mental and physical effort and given low physical input to the interface it is close in nature to the mental demand factor.

The frustration experienced by the participants also failed to reach significance despite a large difference in the means of 2.75 for Mathtalk and 10.5 for the word-processor (t=-1.4; df=3; p=0.25). This non-significant value was due to F1 rating the Mathtalk condition as more frustrating than the word-processor condition. F1 attributed this to his dislike of the keyboard being used. The other participants found the word-processor condition much more frustrating (F4 described it as irritating). F4 rated the frustration in this condition at the maximum possible. This frustration can be attributed to the presentation of the raw notation, as opposed to the better placed and less cluttered Mathtalk presentation. Whilst both conditions allowed control, Mathtalk allowed control appropriate to the tasks that reduced the amount of speech generated.

6.4.5 Summary and Conclusions

A striking difference was seen in the pattern of command usage between the two conditions. Approximately twice the number of commands were used in performing the tasks in the word-processor presentation than in Mathtalk. The main feature of the Mathtalk condition was the use of higher-level objects in accomplishing the tasks. The common unit of movement was the term and the participants also started to move to and from complex objects and use the commands to treat those objects as single units. This type of usage is more appropriate for the evaluation tasks and in the future for manipulation tasks.

Some problems arose from the extensive nature of the browsing language and the short length of training. The prosodic presentation allowed full utterances to be of more use and consequently muting of the speech was not a prominent feature. The basic form of the browsing language was readily learnt and the error rate was low.

Both the presentation style and the increased control over information flow have increased the usability of the algebra notation. The audio glance was used extensively for gaining a description of the expression, but overheads involved in its use meant that it was dropped when the tasks became difficult. Similar overheads and re-use of already known strategies may also account for some restriction in the pattern of command usage.

While LaTeX in a word-processor has all the information required to perform the tasks the presentation has severe usability problems that forced the user to adopt sub-optimal strategies. The word-processor allows easy, error free, control over access to some elements of the structure. However the poor presentation means that only character-to-character styles were used meaning a large number of small units of information have to be integrated by the user. The inability to treat complex objects as discrete units makes the interaction style cumbersome and error prone. The time taken to accomplish each task did not differ significantly between the two conditions. A majority of times were shorter in the Mathtalk condition usage.

Together the participant’s comments and the task load index rating further supported the increased usability of the Mathtalk interface. The overall mental workload was reduced and this was supported by the participants describing the word-processor condition as ‘hard work’. The majority of the participants also seemed to find the word-processor presentation frustrating.

This evaluation, despite not investigating long term usage of Mathtalk, has demonstrated the increased usability of the Mathtalk interface. This has validated the major design principles based on compensation for a lack of external memory and controlling the information flow. The most important features of this design were the use of prosody to improve the presentation and the use of structure based browsing to give control.

6.5 Applying the Design Principles: The Treetalk Program

In this section a paper design for the Treetalk program will be described. This program will provide a speech based user interface for reading phrase structured grammar syntax trees. First a brief description will be given of phrase structured grammars and what information they provide for the reader. Phrase structured grammars are usually presented as tree diagrams. Like algebra this presentation method capitalises on the use of paper as an external memory (Gilmore 1986) and relies on the visual system’s ability to control information flow. Knowing the information content of the tree diagram and what knowledge the reader brings to bear upon the reading process and combining this with a design based on compensation for external memory and control of information flow enables a user interface to be designed that facilitates active reading.

In 1994 two blind students started language degrees at the University of York. A necessary part of their degree was to complete a basic course in syntax. The method of syntactic analysis taught was phrase structured grammars. The main method of presenting phrase structured grammars to all students is by tree diagrams (see Figure 6.1). An alternative method is to use a linear, character based notation where the grouping is indicated by brackets (see Figure 6.3).

There were two reasons that made it necessary to enable the blind students to use tree diagrams. The first was that they would be able to use the same resources as their sighted colleagues. These would be the same teaching materials; producing the same style of work and being able to interact with their colleagues and tutors with a common medium. Secondly, the alternative bracketing notation is cumbersome and difficult to use (as described below). The aim of this paper design was to enable usable access to tree-like diagrams by blind students.

A principal purpose of phrase structure grammar is to present the immediate constituency of a sentence (Lyons 1979). A constituent is simply a component of the sentence. Phrase structure analysis progressively breaks down a sentence into its components or constituents from complex chunks or phrases into simple elements such as words. It is this analysis that the tree diagram presents.

Central to this type of analysis is the notion that a sentence is not a simple linear string of elements, but a layered structure of immediate constituents, with each constituent, in turn, made up of further constituents; all lower level elements being part of those higher in the structure (Lyons 1979).

The two presentation styles described below were designed to describe this constituency. Added to these presentations are the labels that describe the types of each constituent. The presentation contains only the constituents and the labels. It is the reader who brings his or her knowledge to decide that any one phrase is a subject or that the adjective modifies the noun. Similarly, the rules for generating sentences with such constituent structures or how such analyses handle ambiguity are not part of the display and thus are not of concern here.

From the hierarchical presentation of the constituents of the sentence being analysed the syntactician can make inferences about the working of the language. For example, the grammar also shows the binding of these constituents: In Figure 6.1 The verb phrase ‘read the book’ is made from a verb and a noun phrase. The noun phrase is part of the verb phrase, rather than forming a separate branch of the tree. The verb acts upon the noun phrase (object) so forms a closer binding than with any other part of the sentence.

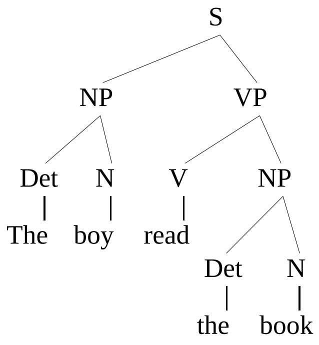


Figure 6.1: Tree diagram showing the phrase structure of the sentence ‘the boy read the book’.

This theoretical information is not explicit in the diagram; it is part of the reader’s knowledge. Just as the reader of algebra can have different levels of interpretation (see Section 3.3) the reader of a syntax tree can also make a range of judgements.

An expert linguist can look at a tree and make assumptions about the syntactic analysis the creator has made in the writing of that tree. For example, if a phrase is labeled DP (determiner phrase) instead of NP (Noun phrase) the reader can make the assumption that the creator is indicating that he or she believes the determiner to be the syntactic head of the phrase rather than the noun. The tree holds the information about the labels; the reader makes any interpretation about the syntactic implications of that presentation.

In a similar way, the auditory presentation should say nothing about the syntactic significance of the node labels or the shape of the tree; it simply presents what is there in such a way that the reader can extract the information in as usable form as possible. This design statement is an equivalent of presenting $y = x^2$ as ‘y equals x super two’, rather than ‘the quadratic, y equals x squared’.

The tree diagram is the standard way of presenting the phrase structure of a sentence. Such a tree is shown in Figure 6.1. Each node has a label naming the phrase or constituent that lies in the sub-tree below that node. For Figure 6.1 the root node **S** contains the whole sentence ‘the boy read the book.’ This tree, like most that would be used by the students, are binary trees. The left branch of the tree leads to the noun phrase (NP) and contains the phrase ‘the boy.’ This phrase is broken into two further constituents: A determiner (det) ‘the’ and noun (N) ‘boy.’ The right hand side of the tree similarly divides to show the structure of the verb phrase.

A more complex tree is shown in Figure 6.2. This tree shows an empty node, indicated by ∅, which is used to indicate where further constituents can be added to the tree. In this example a determiner can be added to the empty node to give the sentence ‘The boys read the antique books.’

Two further features of the trees used at this level are presented in this diagram. Where the

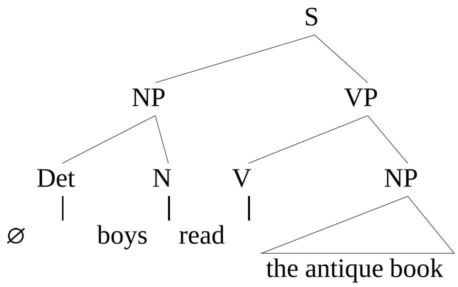


Figure 6.2: Tree diagram showing the phrase structure of the sentence ‘the boys read the antique book’.

[s [NP [DET The] [N boy]] [VP [V read] [NP [DET The] [N book]]]

Figure 6.3: The linear, bracketed notation for representing the constituency of the sentence ‘The boy read the book.’

syntactician wishes to indicate that the structure of a constituent is of no importance the node can be *collapsed*. The constituent ‘the antique book’ appears as the terminal node and the triangle covering the phrase indicates that it is collapsed. As this VP node has only one branch it appears as a vertical branch, rather than a left or right branch. This is the one deviation from the binary tree structure.

The other standard way of presenting this information is to use a linear notation that groups the constituents using brackets. The sentence ‘The boy read the book’ would appear as shown in Figure 6.3.

This style of presentation holds the same information on constituency as the tree diagram. However, such a linear bracketed notation is harder to read than the tree diagram (Kirshner 1989; Gilmore 1986). Even a relatively simple sentence, as used in Figure 6.3, has a large number of nested bracketed groups. Matching brackets is seen as a difficult task (Garnham 1989). As the tree diagram uses both dimensions of the page to present the information, the grouping within the sentence is much easier to apprehend. This is particularly true of the hierarchical aspects of the sentence. For example, that the tree holds the bulk of the information on the right hand side and what it has as constituents, is easier to determine from the tree diagram than the bracketed notation. The labeled nodes and the branches delimit each constituency in a more usable manner than simple linear grouping.

So the principal purpose of the tree diagram is to present the constituency of a clause to the reader. It shows the components of each phrase and to which phrase they belong by the labeling of the nodes. The creator of the tree can indicate which parts of the tree are of interest by collapsing certain nodes. Finally, developments in the structure can be indicated by the presence of empty nodes.

With a similar purpose to algebra notation a syntax tree's purpose is to present the grouping or constituency of a clause. This is the sole purpose of the external memory. It is the linguist, using his or her own syntactic knowledge, that makes any interpretation of the structure.

The audio display must enable a blind reader to apprehend this constituency, the relationships between the constituents, the labels of those constituents and allow the reader to gain a variety of views of the tree in order to carry out his or her linguistic tasks.

Tree diagrams can be complex, because, like algebra, the structure may become intricate. That is, the repetition of simple components can make the information structure complex, as judged by the reader. Like algebra each component of this structure is important. Despite representing English utterances, which can be remembered adequately as a gist, the utterance represented by a tree must be retained exactly. Moreover, the relationships within that structure are of vital importance; losing or transposing one relationship within the tree structure can radically alter its meaning or interpretation. The tree structure does group components together as some components are grouped together in an algebra expression. However, the complexity of a tree comes from simple repetition of branching within the tree, rather than by the introduction of extra symbols and spatial locations. By simple repetition of the divisions within a tree and the labeling of those branches the information becomes complex. The complexity of the tree is not simply the complexity of the sentence; it also arises from how that syntactic complexity is represented.

The following sections take each of the major components seen in the Mathtalk program and apply the same principles to the Treetalk program. Each of the design principles can be used to the same ends in presenting a tree. Both the tree diagram and algebra notation are examples of complex notation. However, the form of the information on the paper differs significantly. This means that whilst the same techniques can be used in both audio presentations, the emphasis given to each design principle may vary.

6.5.1 Using Prosody

Just as prosody was used to indicate the structure of an algebra expression, so it can be used to indicate the structure of a syntax tree. It was not the aim to give the natural prosody of the utterance that a human speaker would use. Instead, prosodic cues were used to indicate the division of the utterance between the two branches of the tree and the length of those branches. The prosodic cues were used to indicate the structure of the tree, and therefore the sentence, but not its meaning.

Prosody was used in a very simple manner. The basic form of control for the reader was to cause the sub-tree at the current focus of attention to be spoken. Invoking the speaking of the sub-tree when at the root node, causes the whole sentence to be spoken. Invoking the speaking of the sub-tree when on the verb-phrase node in Figure 6.1, would give the utterance ‘read the book’.

Two simple prosodic cues were inserted. The principle was the insertion of a pause between the two utterances given by each branch of the sub-tree below the current node. For the sentence in Figure 6.1, speaking the whole tree would give ‘the boy _ read the book’ (the symbol _ represents a pause). Speaking only the verb-phrase sub-tree would give ‘read _ the book’. Finally, uttering the noun-phrase component of the verb-phrase would give the utterance ‘the _ book’.

The second prosodic cue of pitch was used to reinforce the division of the sub-trees. Each sub-tree utterance terminated at a constant base pitch. Working backwards through the utterance, each word was spoken at a higher pitch, until a limit was reached that was determined by the speech synthesiser. So each sub-tree utterance started at a pitch proportional to its length and terminated at a constant pitch. This was the same use of the declination effect seen in Chapter 3.

The prosody imposes an information structure on the utterance. It may not be the prosodic structure used in natural language, but it is a structure suitable for displaying the structure of the tree (and hence the utterance). This was analogous to the technique used in the Mathtalk program.

The prosodic cues only show detail of the structure at one level below the reader’s current level. At any one node only gross information was given about the balance of information between the two branches of the tree. If a node was collapsed, this would be immediately obvious, because no pause would occur within the utterance. Similarly for nodes with only a single branch. A solution for empty nodes is described in the section on the use of non-speech audio below.

Trying to present the whole structure of the tree or any sub-tree would probably overwhelm the listener. So the structure below the succeeding level was *hidden* from the reader. This hiding of complexity in the structure was a direct analogy to that seen in the Mathtalk program and allows control over the information flow in a similar manner.

Just as the term was the basic unit of information in Mathtalk, the sub-tree becomes the basic unit of information in Treetalk. The amount of speech could become relatively large, but the amount of structural information is always restricted. However, other prosodic cues could be used to give more information. Pitch or amplitude could be manipulated to indicate two levels below the current node. Care would have to be taken not to present too much information.

6.5.2 Controlling the Information Flow

To make appropriate use of the information in the syntax tree the listening reader must be able to control the information flow from the improved audio display of the tree. The basic unit of presentation described above was the tree or sub-tree. Again a structure based browsing language

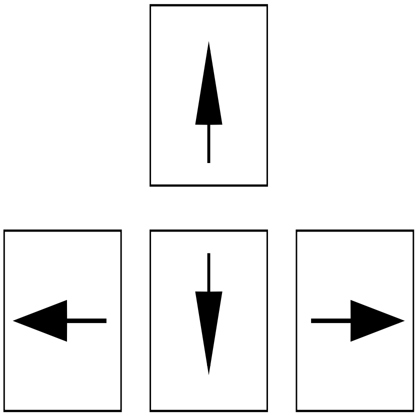


Figure 6.4: The cursor star of the standard PC keyboard with arrows indicating direction.

will be used to improve the information flow. The sub-tree or node will form the basis of the browsing. A very simple method of browsing could be implemented for the Treetalk program. This will be based on the cursor star found on most PC keyboards. (A picture of the cursor star can be seen in Figure 6.4). The layout of the cursor star represents the local layout of the tree structure.

In its simplest form a tree can be represented by an equilateral triangle, with the uppermost vertex representing the root and the other two vertices the branches of the tree. So the left and right cursor keys would take the user down the left and right branches of the tree and the up cursor would return the user to the parent of the current node. As described above, the trees used are not simple binary trees. Any node can have a single child. In these cases the down-cursor would be used to travel to one of these nodes.

The bottom three keys of the cursor triangle take the user down the tree, each key directly mapping onto the layout of the tree. Similarly the top of the cursor triangle maps onto the top of the triangle formed by each subtree.

What should be spoken on arrival at any node is an important question, just as it was with the Mathtalk program. Only the contents of the tree are to be spoken; no interpretation is to be made of those contents. So only the labels and constituents of the labels will be given as output. In addition, navigation and orientation information must be given as the reader moves around the tree. This is vital if the reader is to apprehend the overall structure of the tree and make his or her syntactic interpretations.

The information that has to be extracted is the structure of the tree. The spoken presentation outlined above gave the contents including some structure. A structure-based browsing language will enable the reader to focus upon any part of the tree or the whole tree and gain the information he or she needs.

A default browsing move can be designed based upon the basic unit of browsing. Simply pressing the space bar would speak the current sub-tree from the current node. At the root of the tree this would give a full utterance. This default can be supplemented by the ability to move from terminal element to terminal element. This would allow each of the ultimate constituents to be spoken in turn and allow quick movement to a constituent of interest.

On arrival at a node the label is spoken. This gives basic orientation information to the user on which he or she can base further browsing or speaking moves. The sub-tree will not automatically be spoken. This avoids overwhelming the listener and the hiding of the contents of the tree affords finer control over the information flow. The other information required to make this decision is what nodes are available below the current node. On arrival at a new node further browsing opportunities are presented to the reader. If the new node is a terminal node, the browsing opportunities are replaced by a signal that the node is terminal.

The following orientation information needs to be presented to the reader:

- a node is terminal;
- an empty node exists;
- a node is collapsed;
- that the user has moved either left, right or down a branch;
- there are either left and right or a single downward node available at the current node.

All this information could be given in speech. On arrival at a node the label is given in speech as its abbreviation. A left and right branch are available to the determiner and the noun. So the speech output could be: ‘np; left and right.’ On arrival at the terminal node the output could be: ‘terminal; det; The.’ The salient information is the node label and the contents if the node is terminal. The other information could mask this output and contravene the principle of reducing speech and maximising the information output. In the next section a variety of non-speech options for presenting this browsing information will be discussed.

The cursor star browsing language covers the local moves a reader needs to make. However, larger scale moves will need to be made. These too can be based around the cursor star. Modifying the cursor star, with for instance the control key, could move the reader to the extremes of the tree. So ctrl-up would move the user to the root. Ctrl-left would move the user to the left most terminal node of the current sub-tree etc. Basing the browsing on the cursor star gives the reader a simple and consistent structure based method for traversing the tree.

The labels themselves offer a method for moving around the tree by using the labels as a browsing language. The user would type the label that he or she wished to search and terminate that string with a cursor key. This termination would indicate to Treetalk in which direction to search for the label. For example, if the current focus was at the root of Figure 6.1, typing **np** and pressing the right-cursor key would take the focus to the noun phrase contained within the verb phrase of ‘The boy read the book’.

Such a language could result in movement through a large number of nodes. The route traveled must be presented to the reader in a way that will not interfere with the goal he or she is trying to attain; yet not having this information could result in the user becoming lost in the structure. Just as local movements may be presented in the non-speech audio mode, so could these larger scale movements.

6.5.3 Using Non-speech Audio

As outlined above, the main task of non-speech audio in the Treetalk interface will be to provide navigational and orientation information to the reader. This is the same type of information that was so useful in the Mathtalk program in the form of the terminus sounds.

Earcons for up, left, right and down moves could be designed. As in Mathtalkk these could be based on the prosody of the utterance. A rising tone for the leftmost branch, falling tone for the right and a neutral tone for a down branch. These would be played after the user makes the move and before the node label is spoken. The same sounds could be repeated after the spoken information to indicate what nodes are available to be browsed. A terminus and root sound would also be designed to reduce the amount of verbal clutter.

An extra layer of information could be added by associating musical timbres with the different phrases, in an equivalent manner to the terminus sounds in Mathtalk. There are a larger number of structural categories in the Treetalk program than the Mathtalk program. This presents problems for the designer and the user. The designer has to choose synthesised musical sounds that are sufficiently different that the user can discriminate the different categories. The user has to be able to learn the associations and reliably discriminate between the timbres. Hearing the *left sound* in the noun phrase timbre would add information to the confirmation of movement and the spoken label that follows. The node availability could also inform the user to the types of constituent available on subsequent nodes.

A non-speech sound will be used to indicate empty nodes. When a sub-tree is spoken an empty node would not currently be presented. The utterance would not be divided by prosody, but this could indicate a collapsed node. The word ‘empty’ could be confusing, for example ‘Empty _ boys’ in Figure 6.2. So an empty sound would be used to indicate such a node: ‘<empty sound> _ boys’ would be the output from speaking the noun phrase sub-tree in Figure 6.2.

An audio glance at the structure of the tree could be designed, based on the proposed association between musical timbre and constituent. An earcon for a glance at the structure of a tree based on the prosody of the utterance could be designed. Tones would be played representing each branch of the tree. The length of the sound would be proportional to the size of the sub-tree contained within the node. The tone would have the musical timbre associated with that particular phrase. This would only give information about the topmost nodes.

Notes that represented sub-trees could be played within these higher-level representations, giving chords that that represented the hidden structure within the complex objects. Doubts have to be expressed about the complexity of such a sound and the ability of listeners to reliably recover sufficient information. The presentation would also be strictly serial, giving a depth first presentation of the tree. This would mean required information could be masked or its extraction time consuming.

Brewster, Raty, and Kortekangas (1995) offers an earcon based solution for the presentation of a tree in sound. Each parameter of an earcon, with the addition of stereo position,was associated with a different level of the tree. Consistent variations of these parameters within a node indicate the availability of objects within that node. Such a map has been shown to be effective. However there are some potential problems with its use in this context. The earcons can represent the physical map of the tree. It would not be able to use the parameter of musical timbre to indicate the type of phrase represented at a node. It is also doubtful if there are enough parameters to reliably represent a deep and complex tree.

6.5.4 Conclusions

This section has described the Treetalk program. This was used to apply the design principles used in the Mathtalk program and demonstrate that these principles could apply to wider problems of presenting complex information in speech and non-speech audio. As with Mathtalk, the principle of non-interpretation was used to make a basic design decision.

Prosody was used to indicate the structure of the tree. A pause divided the speaking of any sub-tree into the double or single branches that existed below the current node. Pitch was used to indicate the length of each branch using the declination effect. The principle of hiding complex information was used to hide the complexity of any structure existing within the output from each branch.

The design of the speech output made the sub-tree the basic unit of information. A default mode of speaking was based on this unit. This was combined with a simple browsing language based on the cursor star of the PC keyboard. This basic browsing language was supplemented to make larger moves within the tree. The labels within the tree can form a basis for this language, just as did the structural targets within an algebra expression formed the core of the Mathtalk browsing language.

Finally, non-speech audio was designed to give structural as well as navigational information to the listening reader. A system of earcons were described that may indicate the current location in a tree. A direct analogy with algebra earcons was discussed that associated musical timbres with each type of phrase. These could be blended with the navigational earcons to give information about tree and the type of phrase at the current location. A system of earcons similar to the algebra earcons was described that may be capable of giving global information about the structure of a tree.

The basic principle of designing for absence of external memory and promotion of control over information flow can be readily extended to give a solid foundation for the auditory display of another form of complex information. The principle of non-interpretation indicates what information should be presented to the listener. Prosody can be used to present the structure of the tree within the uttering of the contents of that tree. A structure based browsing language can give control over information flow. An audio glance, based on the re-use of the prosodic cues can be used to give a rapid overview of the tree's structure and can be extended to give orientation and navigation information to the reader. Thus these design principles can be applied in the wider field of presenting complex information to promote active reading for a blind reader.

Chapter 7

Summary and Conclusions

7.1 Introduction

This final chapter summarises the work described in this thesis and the results achieved. It discusses some of the limitations of the work and how they could be overcome. It suggests areas for future investigation in the development of the Mathtalk program and in the general area of the display of complex information. It concludes by assessing the contribution of the thesis to the area of provision of tools for listening reading.

7.2 Summary of this Research

7.2.1 Control and External Memory

The foundation of the design of Mathtalk came from an analysis of the visual reading process and the contrast with reading by listening. The key features of the reading process were seen to be external memory and the fast and accurate control over information flow or selection that the visual system afforded the reader. The lack of these features in listening made the listener passive, where the sighted reader is the active partner in the reading process. Introducing some of the qualities of external memory and control to listening reading is the aim of the design principles laid out in this thesis and are themselves the fundamental design principles.

Arising from the themes of control and external memory is an analysis of what the external memory brings to the reading process. This enables the basic question of ‘what information to speak’ to be answered. In the domain of mathematics, algebra notation displays the grouping of symbols and the relationships between those symbols in a manner that helps the reader perform algebraic tasks. However, it is the reader who interprets the algebra notation to derive mathematical knowledge, not the paper itself. That the reader, not the medium, performs the tasks of interpretation guides how the information is to be presented throughout the design. Designing for fast and accurate control and for the qualities of external memory, together with only presenting structure and content form the foundations of the design process.

7.2.2 Improving the Presentation

Having decided what to speak, the first task was to improve the presentation of that information. This was the question of ‘how to present the information’. Initially, a set of rules for presenting algebraic structure using lexical cues was adapted from that of Chang (1983). These were refined to accord with the principle of non-interpretation.

The notion of simple and complex structure was used as a principle to guide the insertion of these lexical cues. Simple information could be left undelimited. Only complex objects, those with more than one term grouped together, needed to be delimited.

Chang’s method could make the grouping within an algebraic utterance unambiguous. However, it was argued that the potential increase in the number of lexical cues would lead to mental overload. The simple and complex information still needed to be indicated, but in a manner that did not overload the user’s memory. Prosody was investigated as a mechanism for making the simple and complex structure of an expression apparent.

The first stage of the investigation into the use of prosody to improve the display was to find whether the simple set of rules for algebraic prosody proposed by Streeter (1978) and O’Malley, Kloker, and Dara-Abrams (1973) could be extended. A much wider set of rules was then derived, including much more information of pitch contour and pause patterns.

Algebra presented with prosodic cues was compared to that presented with lexical cues and no-cues as a control. Prosodic cues were found to enhance the recovery of structure and retention of content from spoken expressions, over and above that possible with the expressions spoken with lexical cues. In addition to this effect, a major contribution of prosody was to reduce the mental workload associated with the listening task.

From this section of the work a major design principle was proposed: That prosody can be used to give some of the qualities of an external memory to the presentation of complex information in speech. It can make the information easier to apprehend and easier to remember. The prosodic cues can be thought to be the equivalent of the typographic rules for formatting algebra in print. The notion of simple and complex structure was used as a guide, during the whole of the design, to judge when cues, prosodic or lexical, should be used.

7.2.3 Controlling Information Flow

This section of the work addressed the second theme of the thesis, that is, the control of information flow. It was argued that structure or grouping in an expression was the important feature, as this made the information complex, and these units were the objects manipulated or read during algebraic tasks. Thus, a structure based browsing language was designed.

An informal, co-operative evaluation style was used during iterative development to produce a usable browsing language that would promote active reading by giving fast and accurate control of information flow. The initial design was based on a word-processor paradigm, with the cursor keys, plus modifiers, being used to move from object to object within the expression. The rich structure of the complex information of algebra rapidly led to contrived and arbitrary mappings for the moves available in the command language. One vital component missing was the function of uttering the *current* object or scope within the expression. This meant that in the auditory presentation where the signal is transient, the current focus could not be uttered without moving from and to the present scope.

As a result, a completely new command language was developed. This language was based on a stylised form of giving commands in speech. A set of command words such as **current**, **next** and **previous** were combined with structural targets such as **expression**, **term** and **item** to give a wide range of commands that covered all the moves a user might wish to make. This language gave commands of the style **current expression** and **next term**. This style of command was designed to be both easy to teach and learn, as well as giving a relatively simple structure to a language for a necessarily complex browsing environment.

During browsing, complex objects were not spoken in full as the focus of attention reached that point. Instead, as the focus landed on that object, only its type was spoken, hiding the detail lying inside. This became an important feature of controlling information flow. It cut down the amount of information presented at any one time into a manageable chunk and made the structure of the expression more explicit. Again, the hidden objects used the notion of simple and complex structure to guide the presentation of information. An emergent feature of the hidden objects was the overview of an expression. A large, complex expression could be reduced to a brief utterance.

The studies of prosody and the analysis of typography, both of which first divide an expression into terms, indicated that the term was the basic unit of information in an algebraic utterance. This was used to provide a default browsing strategy for listening readers. It was hoped that this default browsing style would give listening readers easy access to a strategy that would provide information in suitable chunks to make the reading process effective and comfortable.

This language and associated styles was refined during several iterations of informal, co-operative evaluation to give a browsing mechanism that offered a wide range of low-level moves that a listening reader could combine to give rapid and accurate movement to any part of an expression in the way he or she wished to formulate the move.

The evaluation of the browsing component suggested that the language would give appropriate control over information flow, so allowing active reading. The language was easily taught and rapidly learnt. Users spontaneously made up new commands from knowledge of the command words; the speech overview using hidden objects was widely used and the hidden objects generally appreciated; the default browsing was also widely used. The users also readily combined low-level browsing moves into higher-level tactics and strategies.

7.2.4 Gaining an Overview

This section of the work returned to the theme of improving the presentation of complex information in audio. An audio glance at an algebra expression was developed based on the ability of prosodic cues to present the structure of complex information. A glance was defined as:

> ‘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.’

The salient information for the listening reader of complex information is the structure of that information. To be a glance, it would have to lack the detail of all the instances of the types of objects, yet show the presence, type, location and size of those objects.

An audio glance, called algebra earcons, was developed from the prosodic cues for spoken algebra. The prosody of speech can indicate the structure of an expression, but the words in the utterance deliver all the detail. The criteria of the glance can be fulfilled by simply representing the classes of the object in the expression, rather than the instance. To achieve this, each type within the algebra notation was replaced with a different musical sound. These sounds were musically presented with a stylised set of prosodic rules, adapted to give the algebra earcons a strong rhythmic content.

A simple recognition experiment was run with a representative set of expressions, so that design faults in the earcon design could be ironed-out. The mean number of expressions correctly recognised from the earcons was 73%. Several flaws were discovered, some of which were addressed before a second trial of the same experiment was run.

This second trial also showed a similar high recognition rate, indicating the ability of algebra earcons to convey the structure of an expression rapidly, with no detail. The second trial also addressed the representation of the expression retained from the algebra earcon. The second experiment showed some improvements in the design of algebra earcons, but the timing structure, which was not redesigned, still presented some problems.

The investigation of internal representation of an expression showed a range of retention from a full account of the expression to a general impression of the complexity. All these representations could work as a glance, but the representations were heavily weighted towards a full account.

7.2.5 The Complete System

The last section of the work on Mathtalk addressed the evaluation of the integrated components of the system. Each component, the prosody, browsing and audio glance, had been evaluated separately. However, it was important to demonstrate that the full Mathtalk program could improve the reading of an expression in the auditory mode in an ecologically valid setting.

The Mathtalk program was compared to algebra written in the LaTeX typesetting language and presented in a word-processor. This format contained the same information as either a print or Mathtalk presentation. The LaTeX, however, it was concluded, gave that information in a less usable form that did not have the qualities of external memory included in the Mathtalk presentation.

The LaTeX was presented in a word-processor which gave browsing, but in a paradigm more suited to plain text than algebra notation. Most importantly, this alternative was akin to a form commonly used by blind students in educational settings.

Four blind students performed navigation and mathematical evaluation tasks in both presentation formats. Both qualitative and quantitative measures were taken during the evaluations. The participants showed a subjective preference for the Mathtalk presentation. Subjective mental workload was seen to be reduced in the Mathtalk condition, especially on the mental demand factor. The Mathtalk program was seen to give the type of active access that the participants required, despite the necessarily more complex interface and short learning and evaluation time.

The quantitative measures were more ambiguous, but indicated enhanced performance with the Mathtalk program. There was no significant decrease in the time taken to complete tasks in the Mathtalk condition, but the majority of tasks were completed in a shorter time. Significantly fewer commands were used in the Mathtalk condition than word-processor condition. This means the participants could achieve the same ends, with fewer commands, indicating a less demanding interface.

With Mathtalk, more commands were used that moved over larger portions of text, directly to the desired object. In the word-processor condition the dominant move was to read character by character. The commands used in Mathtalk tended to give the information the participant required, rather than a surplus of speech. These differences indicated that the participants had a better control over the access to information and thus more active reading.

The non-speech component was appreciated. This was particularly true of the terminus sounds, that indicated the end of different constructs in the expression. These had been designed to use the timbres associated with structure type. The algebra earcons were widely used in the navigation tasks, where the participants used them to give a rapid view of the expression, after which they could describe its type and move to the portion of the expression required.

Usage of the audio glance disappeared during the evaluation tasks. This probably reflected the increased mental demand of these tasks, which the participants admitted finding onerous.

In general, this evaluation showed that the Mathtalk program achieved its aims of providing an active reading of algebra notation. By addressing the themes of external memory and control of information flow a set of design principles can be given that will promote active reading of algebra notation.

7.2.6 General Applicability of Design Principles

In order to show that the design principles used in the Mathtalk project were generally applicable, a paper design for the Treetalk program was presented. Syntax trees used in linguistics and other disciplines offer another source of complex information. A full description of the tree is virtually unusable, not least because the listener is the passive recipient of a flow of unstructured, potentially overwhelming information.

The design principles used in Mathtalk were applied to the presentation of trees that displayed the syntax of an utterance according to phrase structured grammars.

Hidden information formed a large part of the design of the Treetalk program. There is too much structure to present all at once in any one tree. Structure subsequent to the level below the current was hidden to the listener at any one time. Prosodic cues were used to indicate the division of the structure below the current point into two sub-trees.

Simple browsing based on the cursor star of PC-keyboards was used to traverse the tree. The triangular shape of the cursor star was mapped onto the intrinsic shape of the tree. In this way the listening reader could gain active control over how the structure of the tree was presented. A more complex browsing would be available via the labels on the nodes of the tree. Thus, browsing based firmly on the structure of the complex information can give a mechanism capable of giving active reading.

Non-speech audio also had a potentially large effect on the reading interaction using Treetalk. One aspect of a tree's complexity is the nested repetition of identical structures to give the tree. Non-speech audio was proposed as a mechanism of indicating what moves were available and made at each transition. The association of musical timbre with phrase type have similar opportunities to indicate environment and provide glances at structure based on prosodic signals, as were used in the Mathtalk program.

Even though this design was not evaluated, it does demonstrate that the same principles can be applied, albeit in different ways, to enable active reading of another form of complex information. In summary, the major design principles derived from this work are:

- designing for external memory and control of information flow enables active reading;
- non-interpretation of output in the presentation is a basic principle in reading;
- simple and complex information structure can guide how information is presented;
- prosodic cues can be used to indicate structure within complex information;
- prosodic cues improve the presentation's performance as an external memory;
- structure based browsing supports active reading;
- hiding information aids in the control of information flow;
- direction giving in speech can form a basis for this active reading;
- an audio glance can give an overview of the information, thus aiding planning;
- earcons and prosody can be combined to give a glance at the structure of an expression;
- the concomitant association of musical timbre with structure can have design implications for use of sound throughout the interface.

7.3 Limitations of this Research

Despite the strong design principles arising from this work, there are some limitations to the work presented that will need to be addressed. Each of the components of the Mathtalk program are taken in turn and their limitations described. Finally the overall evaluation is discussed, together with a critique of the attempt to achieve listening reading.

This research only tackled the problem of reading algebra. Reading was an obvious first step in providing usable access to algebra notation and other complex information. However, for true access to be provided, the listening reader also has to be able to write and manipulate these types of complex information. The problem of writing and manipulating algebra notation is discussed below in the section on future work.

7.3.1 Improving the Presentation

The scope of the algebra notation used in the Mathtalk program was narrow, but all the basic constructs were used. By limiting the scope of the mathematics presented, some problems were avoided. Where certain constructs are overloaded with meaning, the naming of those objects could cause problems. Without extra information in the internal representation of the expression that would enable a richer presentation, such overloaded representations cannot be reliably discriminated.

This problem is linked to the principle of non-interpretation. This principle is severely limited by the need to name objects in speech. Print or tactual symbols are only interpreted by the reader, they are not named in the same way as spoken symbols. English is not rich enough to have a name for each construct that is abstracted from the intention of that object. As a wider range of mathematics is incorporated into such a reading device, attention will have to be given to how such objects can be named, either according to their intention or without reference to this intention and only the construct type. Nevertheless, an analysis of what information the external memory holds, what is brought to this information by the reader and the context of use, lays the foundations for an appropriate presentation of algebra notation.

There were several limitations in the work on the use of prosody to improve the use of spoken algebra as an information source. The size and scope of the expressions used to derive the rules for algebraic prosody were too small and too narrow. Despite the success of these rules, a deeper investigation of prosody would have provided a richer set of rules. It would be interesting to find if making the prosody more ‘natural’ would have any significant effect on the apprehension of structure in an utterance, or if any improvement would be in reducing mental workload or simply in user satisfaction.

The most interesting limitations are in the effectiveness of prosodic cues to facilitate discrimination of structural boundaries. In complex expressions, particularly where there was nesting of structures, listeners were unable to recover all the structure in an expression. This inability to recover all structure was also seen at the end of expressions. These two situations both reduce the redundancy in the use of prosodic cues. The true limitations of prosody and the reasons for this limitation need to be investigated.

Only the overall effect of adding prosody was investigated in this thesis. There was potentially more information in the presentation than was investigated. For example, the declination effect and signaling of expression length was not investigated. Similarly, subtle cues, such as the placing of a longer pause on the side of a relation adjacent to the longer side of an expression, were not investigated.

During the evaluation of the prosodic component the prosodic cues and lexical cues were only tested in isolation. Whilst the design was enough to show that prosodic cues alone were sufficient to improve the presentation, and were significantly better than lexical cues, it would have been interesting to combine the two forms of presentation. Lexical cues would have undoubtedly benefited from the addition of prosody, perhaps mitigating the effects of increased verbiage. In addition it would be useful to discover where the use of lexical cues is useful and support the use of prosodic cues. Prosodic cues significantly improve the presentation, but are not a panacea, nor are lexical cues likely to be all bad.

7.3.2 Controlling Information Flow

The use of the browsing language in the Mathtalk program demonstrated that controlling information flow could make the listening reader more active. It was assumed that a structure based method of browsing would provide suitable method of control appropriate to most mathematical tasks. A task analysis of both sighted and blind mathematicians performing mathematical tasks would have provided a rich source of information by which the browsing could be designed.

There were two other major limitations to this area of study. The first was with the nature of the control and the second was with the nature of the evaluation.

A simple command language was used to implement a structure based browsing style. Like all command languages, the one used in Mathtalk has the limitation that the words and structure used must be understandable by the users. Few problems were encountered on this front with the current study, but difficulties are likely to arise as the system is expanded.

The command language was very low-level and fine grained. This allowed all moves to be made and these moves could be combined to give higher level tactics. This design had obvious advantages of flexibility. However, the large number of commands that had to be issued to achieve some goals could be a hindrance to easy flowing control of information.

No other options, apart from a command language, were explored. The SpeechSkimmer developed by Arons (1993) offers a method for browsing the structure of speech. This might be applicable to this situation as the structure of the expression is directly reflected in the form of the utterance.

The command language used has obvious limitations. The actions cover most moves a reader would want to make. This may well not be the case with the targets. As the scope of the mathematics to be read increases so will the number of targets. The simple mnemonic mapping will not be extendable and the learning task will increase. This will be particularly true when the Mathtalk program will be extended to writing and manipulating algebra notation. In the future more general, direct methods of controlling information flow will have to be explored.

The evaluation of the information control was sufficient to show that it would provide the means for active reading. However, the evaluation lacked any longitudinal element. Such studies would have shown how rapid the reading could become. The start of strategies and tactics were observed. Longer term observation would have revealed more information on this aspect of the control system.

7.3.3 Gaining an Overview

The algebra earcons have two major limitations. The first is a limitation of their design and the second is a potential usability problem. The algebra earcons are simply a glance at the structure of an expression. This is the highest level view of an expression. However, while working with algebra several levels of detail will be required. An interesting avenue of future work would be to introduce a mixture of detail and higher-level views into the glance. This would probably involve mixing music and speech in the same glance, while retaining the musicality of the glance on which its utility is based.

Whilst the algebra earcons provided a glance, some of the participants' comments revealed that the evaluation tasks were mentally hard work. If this were true, it would limit their usefulness as a glance.

7.3.4 The Complete System

The evaluation of the integrated Mathtalk program had several limitations. The first was that there were few participants used in the evaluation. To an extent this reflects the problems that Mathtalk ultimately aims to solve. Blind school-children generally underachieve in mathematics. Mathtalk aims to provide usable access to algebra notation. So in undertaking the evaluation only a relatively small pool of participants was available. The additional need for computer skills and the unattractiveness of mathematics further compounded these problems.

As the writing and manipulation of the notation were not possible, it was difficult to design ecologically valid tasks for the evaluation. Without a mathematical task to give an expression context reading an expression becomes a shallow activity. This drove the inclusion of substitution and evaluation tasks in the final evaluation, but again, only being able to read made the design somewhat contrived.

Apart from the paucity in number of students, perhaps the most severe limitation in the final evaluation was the short-term nature of the study. The Mathtalk program was used for less than one hour in the final evaluation. To get a real feel for how much the design improves the access to algebra notation, much longer studies should be used. Such observations would reveal if the audio glance would begin to be used during more mentally taxing mathematical tasks. It would show if more complex tactics and strategies would be used in the reading of algebra notation.

One remaining limitation needs to be discussed: Is listening reading possible? As described earlier, it is not possible to give a completely non-interpretive presentation. Whilst the presentation has been improved, it is not as reliable as its print alternative. The control allowed through browsing is not the match of that given via the visual system. The listening reading allowed by Mathtalk is not a true equivalent of the mechanical aspects of visual reading. In this sense, reading is probably not an achievable goal in the audio modality. Nevertheless, the design proposed in Mathtalk does provide a usable means of accessing algebra notation that could be used to perform algebraic tasks.

7.4 Future Work

Future work arising from this research could take two directions. The work on algebra notation can be extended and the design principles could be extended to other types of complex information. Within the domain of mathematics the work would be extended to become more general and also move in the other direction to finer grained investigation of features of the present work.

7.4.1 The Maths Project

The work on the Mathtalk project has formed the core of the European Union funded Technology Initiative for Disabled and Elderly People (Tide) project called Maths. This has the aim of providing a multi-modal algebra workstation for visually disabled school-children (Edwards and Stevens 1994).

The Maths workstation has been designed to allow algebra notation to be read, written and manipulated using a variety of input and output modes. Reading algebra notation itself is something of a dead-end unless that reading can be accomplished in the context of surrounding text and unless the expression being read can be manipulated and new expressions be written. The Maths workstation has to accomplish all these tasks in order for the product to be useful. These tasks also have to be enabled in as usable a fashion as possible if the Maths workstation is to achieve its aim of enabling visually disabled school-children to progress in the field of mathematics education.

The design principles laid out in the Mathtalk project have been continued in the Maths project. The Maths workstation has been designed to allow visually disabled school-children to use algebra notation as their sighted colleagues might use pen and paper; it does not seek to teach mathematics, which is still the role of the teacher.

Speech, non-speech and braille are used to present the notation. The keyboard, braille keyboard and speech input are used to write manipulate and browse the notation. The twin themes of external memory and control of information flow are again used to guide the design of the presentation of the algebra notation in audio and braille.

Two new technologies (apart from braille) have been used in the design of the workstation. These are spatial sound as part of the output and speech recognition as an input technique.

At present, all the audio output appears from a single source, that is, either a loudspeaker or in mono over headphones. The technology now exists for the spatial display of audio information (Wenzel, Wightman, and Kistler 1991) and this potential has been recognised in the field of audio displays for visually disabled people (Crispien, Wuerz, and Weber 1994).

Adding a spatial component to the audio output could add another layer of information. Instead of coming from a single source, the expression would be laid out in front of the user, as print is on the page. Instead of viewing the information in purely structural terms, the listener could remember the spatial location of objects when formulating reading moves and manipulations.

As the reader moves around the expression the focus of attention would move in space within the display. This would help to confirm moves made by the user, by giving richer, but unobtrusive feedback. Having the location within an expression displayed in space could also help in orientation within the expression, perhaps reducing the potential for becoming lost. This extra information available during the interaction could reduce errors and make the process easier and more satisfying. A spatial component could also permit a direct manipulation style of interaction to take place with algebra notation by allowing interaction to take place with gestures, rather than commands issued from the keyboard. These ideas are explored in Harling, Stevens, and Edwards (1995).

The second innovation within the Maths workstation is the use of speech recognition as an input mode. As described above, use of the keyboard to give control over information flow has its limitations. Speech recognition would allow the browsing of an expression and writing of the notation to become separated. The browsing language described in Chapter 4 had a natural spoken form that would lend itself easily to speech input. Instead of typing **NT** to give the command **next term**, the user would simply utter the command ‘next term’. The technology now exists that such restricted vocabularies can be reliably recognised.

The writing and manipulation of the notation could also be transferred to speech input. Chang (1983) suggested that his method of inserting lexical cues into algebra notation to make the structure unambiguous could be used for input as well as output of mathematics. This approach is also being used in the Maths project. Determining how successfully each of the many modes work together and where they complement each other will provide many useful guidelines to the designers of auditory and multi-modal displays.

7.4.2 Work Within the Design of the Mathtalk Program

The section on limitations of the work presented many opportunities for avenues for future research on the design principles laid out in this research. Many of these were to investigate the fine detail of some of the prosodic effects. Each of the cues used could be investigated so that they used the optimum values. In addition, a suitable range of values could be specified to give more concrete guidelines to designers.

The algebra earcons also present opportunities for future research. Algebra earcons already give a rapid overview of an expression, but it would be interesting to find the limits of their usefulness when played at speed. Synthetic speech can be understood at high speeds (Schwab, Nusbaum, and Pisoni 1985) and if the algebra earcons are not usable at such high speeds, then speech may serve as well. In addition, speech glances deserve investigation in their own right.

As described in the section on limitations of the work, the audio glance only gives one type of view. Larkin (1989) suggests that readers use the external memory to gain views with different levels of detail. Future research would investigate the possibility of developing views that mix speech and non-speech sounds to give a variety of views. Finding the balance between richness of the rendering and how much work the user has to perform to gain such a view would provide useful guidelines to designers.

As the audio glance has proved successful it would be interesting to investigate its application in other areas. The Treetalk program offered one such application and glances at the structure of program source code could also be developed. More general situations could also benefit from this approach. One such field would be the graphical user interface. These are complex visual displays with many types of object displayed on the screen. Associating timbres with objects such as windows, menus and buttons, combined with a spatial sound component could provide a glance at the screen for a visually disabled user. Rendering all such objects in speech would give rise to a cacophonous display in which little could be found with any rapidity.

7.5 Conclusions and Contributions of the Thesis

Despite the limitations of the work described above, the work presented in this thesis is a progression in the knowledge on how to design auditory interfaces for usable access to complex textual information in the auditory modality.

The design principles laid out can move the passive listener towards being an active reader. Prior to this work the emphasis has been on access to information and that information is simply spoken at the user. Rather than concentrating on making the implementation of the computer system easier, this work has concentrated on what the user needs to become a reader.

The twin themes of external memory and control over information flow provide a foundation on which to base the design of interfaces for blind computer users. When the consequences of non-visual interaction with complex information are made explicit, then appropriate solutions can be designed. The user can now be the reader, rather than being read to by a computer.

7.5.1 Improving the Presentation

One contribution of the thesis is to give the designer a context in which the development of a reading system can be founded. The first requirement in the quest to improve information presentation is to know what information needs to be displayed. By examining what information is presented on the external memory and what knowledge the reader brings to the interaction the designer can decide what information the listening reader needs to access.

A major contribution of the thesis is the demonstration that prosodic cues can be used to give a dramatic improvement to the presentation of complex information in sound.

Without prosody, a spoken presentation is an undifferentiated stream of words, lacking much of the information present on the page. This thesis has shown that prosodic cues can be ascribed to structure in the complex information and they can be simulated in synthetic speech.

The addition of these cues was shown to increase apprehension of structure; increase retention of content and to decrease the workload associated with the task. By using prosodic cues, some of the qualities of the printed external memory can be introduced to an audio presentation. The use of prosody has applications wherever complex information has to be presented in synthetic speech.

Another significant contribution that arose from the work on prosody was the development of an audio glance at the structure of complex information. Algebra earcons allowed a rapid, high-level view of the structure of an expression. A remarkably large amount of information was recovered by listeners and the audio glance allowed expectation of expression type, and thus planning, to be made.

Importantly, algebra earcons provide a method for linking speech and non-speech in the interface via prosody. The glance itself was based on prosody. In addition, the association of timbre with types of object allows a consistent method of conveying information about navigation and orientation in the structure of the expression via non-speech audio.

7.5.2 Improving Control of Information Flow

The partner of the improved presentation is giving the listening reader control over information flow. This thesis has shown that giving a listening reader a suitable control over the flow of information makes that reader active and provides a much more usable interface. The idea of control is perhaps simple, but the design of the Mathtalk program has shown how vital it is for considerations of control and external memory to be made a central part of the development of any tool to facilitate reading.

The principal aim of the initial stages of reading complex information is to apprehend the structure of that information and how content objects are arranged in that structure. The design principles laid out in this thesis all emphasise that structure. This is especially true of the browsing language. Given that the purpose is to comprehend the context of the structure, the browsing language itself was based on the structure of the expression and those simple moves that might be made in apprehending that structure. This gave fast and accurate control and this needs to be the aim of any control that proposes to enable effective reading.

Perhaps the most valuable part of the control component was the development of the hidden objects. These hid complex information and thus avoided the automatic rendition of large portions of an expression. The hidden objects also made the structure of an expression more apparent.

The final evaluation showed that the design principles achieved their aim of promoting active reading of mathematics and that this aim was the correct approach. The strong emphasis on evaluation throughout this thesis has helped to ensure the usability of the final program and helped to validate the design principles.

In conclusion, this thesis has presented a set of design principles that enable an active reading of complex information using speech and non-speech audio. Basing the design on compensation for lack of external memory and promotion of control over information flow to give this active reading. Prosody was shown to be effective in improving the presentation of complex structures. Hiding complex information and giving the listening reader a structured based method for fast and accurate control over what is spoken was shown to give appropriate and usable access to the information. The last component was to develop an audio glance at the information, a feature missing from speech based interfaces for blind computer users. Each of these components was fully evaluated, as was the integrated system, giving the principles a solid foundation. The design principles presented here should enable designers to develop better tools for the access to complex information for blind children and adults in education and the work-place.

Appendix A

Spoken Algebra

A.1 Prosodic Investigation

The expressions used in the investigation into algebraic prosody are shown in their contrast pairs with data for pitch, timing and emphasis according to syllable in the spoken form of the expression. Accented syllables, are indicated by a emboldened typeface in the spoken form of the expression. Pauses after syllables are shown in milli-seconds. Pitch is shown in Hertz. A dot '.' after a number in this row indicates the frequency at the syllables start. A dot preceding the number indicates the pitch at the end of the syllable, otherwise pitch 'on' the syllable is assumed.

Expression	$x^n + 1$					
Syllables	1	2	3	4	5	6
Spoken	**x**	to	the	**n**	plus	**one**
Pitch (Hz)	192			.130	138.	.105
Pause after (ms)				200		

(a) Expression 3.2.

Expression	x^{n+1}					
Syllables	1	2	3	4	5	6
Spoken	**x**	to	the	**n**	plus	**1**
Pitch (Hz)	176					109

(b) Expression 3.3.

Exp	$ab+c-ef-g$										
Sy	1	2	3	4	5	6	7	8	9	10	11
Sp	**a**	**b**	plus	**c**	mi	nus	**e**	f	mi	**nus**	g
(Hz)	109-208	130-140	160	155	128	128		168	124-130		106
(ms)		300		450				700			

(c) Expression 3.10.

Exp	$a(b+c-e(f-g))$												
Sy	1	2	3	4	5	6	7	8	9	10	11	12	13
Sp	**a**	**times**	**b**	plus	**c**	mi	**nus**	e	**times**	f	mi	**nus**	g
P(ms)	1 000		364					285					

(d) Expression 3.11.

Expression	x^4n				
Syllables	1	2	3	4	5
Spoken	**x**	to	**the**	4	**n**
Pitch (Hz)	186				122

(e) Expression 3.4.

Expression	x^{4n}				
Syllables	1	2	3	4	5
Spoken	**x**	to	the	**4**	**n**
Pitch (Hz)	150				108

(f) Expression 3.5.

Expression	x_4	
Syllables	1	2
Spoken	x	**4**
Pitch (Hz)	135	176-118-142

(g) Expression 3.6.

Expression	x^4			
Syllables	1	2	3	4
Spoken	**x**	to	the	**4**
Pitch (Hz)	168			107

(h) Expression 3.7.

Exp	$y = ax + bx + c$									
Sy	1	2	3	4	5	6	7	8	9	10
Sp	**y**	**e**	quals	**a**	x	plus	**b**	x	plus	**c**
(Hz)	186	135	135	165	133	150		118	128	107
(ms)			175		141			110		

(i) Expression 3.8.

Expression	$y = ax^2 + bx + c$										
Syllables	1	2	3	4	5	6	7	8	9	10	11
Spoken	**y**	e	quals	**a**	x	**squared**	plus	**b**	x	plus	**c**
Pitch (Hz)	176		105	189			133	138			103

(j) Expression 3.9.

Expression	$1 + \frac{x}{y} + 4$							
Syllables	1	2	3	4	5	6	7	8
Spoken	**1**	plus	**x**	o	ver	**y**	plus	**4**
Pitch (Hz)	178					.118	138	115
Pause after (ms)						298		

(k) Expression 3.22.

Expression	$\frac{1+x}{y+4}$							
Syllables	1	2	3	4	5	6	7	8
Spoken	**1**	plus	**x**	**o**	ver	**y**	plus	**4**
Pitch (Hz)	196		117	125	125	140		110
Pause after (ms)			323		290			

(l) Expression 3.23.

Expression	$a - b + c$					
Syllables	1	2	3	4	5	6
Spoken	**a**	mi	nus	**b**	plus	**c**
Pitch (Hz)	168	140	140	150	155	113
Pause after (ms)	148			181		

(m) Expression 3.12.

Expression	$a - (b + c)$					
Syllables	1	2	3	4	5	6
Spoken	**a**	**mi**	**nus**	b	**plus**	c
Pitch (Hz)	165					101
Pause after(ms)	278		216			

(n) Expression 3.13.

Expression	$3x+4=7$						
Syllables	1	2	3	4	5	6	7
Spoken	**3**	x	plus	**4**	e	quals	**seven**
Pitch(Hz)	192	142-144	121		127		109
Pause after (ms)		115		211			

(o) Expression 3.14.

Expression	$3(x+4)=7$						
Syllables	1	2	3	4	5	6	7
Spoken	**3**	**x**	plus	**4**	**e**	quals	**seven**
Pitch (Hz)	198	117	117	133	138	138	114
Pause after (ms)	120						

(p) Expression 3.15.

Expression	$x+y^3$			
Syllables	1	2	3	4
Spoken	x	plus	y	cubed
Pitch (Hz)	168	133	138	107
Pause after (ms)	181			

(q) Expression 3.16.

Expression	$(x+y)^3$				
Syllables	1	2	3	4	5
Spoken	**x**	plus	**y**	all	**cubed**
Pitch (Hz)	182				103

(r) Expression 3.17.

Expression	$-(a+b)$				
Syllables	1	2	3	4	5
Spoken	**mi**	nus	**a**	plus	**b**
Pitch (Hz)	176	121	138		112
Pause after (ms)		252			

(s) Expression 3.18.

Expression	$-a+b$				
Syllables	1	2	3	4	5
Spoken	**mi**	nus	**a**	plus	**b**
Pitch (Hz)	173	173	118	135	110
Pause after (ms)			255		

(t) Expression 3.19.

Expression	$a+ba+b$					
Syllables	1	2	3	4	5	6
Spoken	**a**	plus	**b**	a	**plus**	**b**
Pitch (Hz)	173-186	130	168	124	124	115
Pause after (ms)	510					

(u) Expression 3.20.

Expression	$(a+b)(a-b)$							
Syllables	1	2	3	4	5	6	7	8
Spoken	**a**	plus	**b**	times	**a**	**mi**	nus	**b**
Pitch (Hz)	182	137	140	131	131			105
Pause after (ms)			218					

(v) Expression 3.21.

Expression	$\frac{a}{b}$			
Syllables	1	2	3	4
Spoken	**a**	o	ver	**b**
Pitch (Hz)	144			103

(w) Expression 3.24.

Expression	ab	
Syllables	1	2
Spoken	**a**	**b**
Pitch (Hz)	142	110

(x) Expression 3.25.

A.2 Evaluation of Prosodic Component

A.2.1 Training Expressions

1. $y = 4(x+7)+8y$
2. $3(4x+7)(x+4)$
3. $3(4x+7(x+4))$
4. $\frac{1}{2}+\frac{x+2}{x-3}$
5. $(\frac{9x}{3(x+4)})(x+9)$
6. $x^n+1 \neq x^{n+1}$

A.2.2 Raw Scores for Recall of Expressions

LP Lexical: Structure													
	LP1:1	LP2:1	LP3:1	LP4:1	LP5:1	LP6:1	LP7:1	LP8:1	LP9:1	LP10:1	LP11:1	LP12:1	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	1	1	1	1	1	1	1	1	1	1	1	1	12
Q3	1	1	1	1	1	0	0	1	0	1	1	1	9
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	1	1	1	0	0	0	1	0	0	0	1	1	6
Q6	0	0	1	0	0	1	0	0	1	0	1	1	5
Q7	0	0	0	0	0	0	0	0	0	0	0	0	0
Q8	1	0	0	1	1	1	0	1	1	0	1	0	7
Q9	1	1	1	1	0	1	1	1	1	1	1	1	11
Q10	1	0	0	0	1	1	0	0	1	0	1	0	5
Q11	1	1	1	1	1	1	1	0	1	0	0	1	9
Q12	0	1	1	1	0	1	1	1	1	0	0	1	8
TOTAL	9	8	9	8	7	9	7	7	9	5	9	9	96

Table A.1: Scores for the structure recall for the *lexical* condition of the lexical prosody (LP) group.

LP Lexical: Content													
	LP1:1	LP2:1	LP3:1	LP4:1	LP5:1	LP6:1	LP7:1	LP8:1	LP9:1	LP10:1	LP11:1	LP12:1	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	1	1	1	1	1	1	1	1	1	1	1	1	12
Q3	0	0	0	1	0	0	0	1	0	0	0	0	2
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	0	1	0	0	0	0	0	0	0	0	1	0	2
Q6	0	0	0	0	0	0	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	0	0	0	0	0	0	0
Q8	0	0	0	1	1	1	0	1	0	0	0	1	5
Q9	1	1	1	1	1	1	1	1	1	0	1	1	11
Q10	0	0	0	0	0	1	0	0	0	0	0	0	1
Q11	1	1	1	1	1	1	1	0	1	0	0	1	9
Q12	0	1	1	1	0	1	0	1	1	1	1	1	9
TOTAL	5	7	6	8	6	8	5	7	6	4	6	7	75

Table A.2: Scores for the content recall for the *lexical* condition of the lexical prosody (LP) group.

LP Lexical: Overall													
	LP1:1	LP2:1	LP3:1	LP4:1	LP5:1	LP6:1	LP7:1	LP8:1	LP9:1	LP10:1	LP11:1	LP12:1	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	1	1	1	1	1	1	1	1	1	1	1	1	12
Q3	0	0	0	1	0	0	0	1	0	0	0	0	2
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	0	1	0	0	0	0	0	0	0	0	1	0	2
Q6	0	0	0	0	0	0	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	0	0	0	0	0	0	0
Q8	0	0	0	1	1	1	0	1	0	0	0	0	4
Q9	1	1	1	1	0	1	1	1	1	0	1	1	10
Q10	0	0	0	0	0	1	0	0	0	0	0	0	1
Q11	1	1	1	1	1	1	1	0	1	0	0	1	9
Q12	0	1	1	1	0	1	0	1	1	0	0	1	7
TOTAL	5	7	6	8	5	8	5	7	6	3	5	6	71

Table A.3: Scores for the Overall recall for the *lexical* condition of the lexical prosody (LP) group.

LP Prosodic: Structure													
	LP1:2	LP2:2	LP3:2	LP4:2	LP5:2	LP6:2	LP7:2	LP8:2	LP9:2	LP10:2	LP11:2	LP12:2	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	0	1	1	1	1	1	1	1	1	1	1	1	11
Q3	1	1	1	1	1	1	1	1	1	1	1	1	12
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	1	1	1	1	1	1	0	1	1	0	0	0	8
Q6	1	1	1	1	1	1	1	1	1	1	1	1	12
Q7	1	1	1	1	1	1	1	1	1	1	1	1	12
Q8	1	0	0	1	1	0	0	0	1	0	1	1	6
Q9	1	1	1	1	1	1	1	1	1	0	1	1	11
Q10	1	1	1	1	1	0	1	1	1	0	1	1	10
Q11	1	1	1	1	0	1	0	1	1	0	0	1	8
Q12	1	1	1	1	1	1	1	1	1	1	1	1	12
TOTAL	11	11	11	12	11	10	9	11	12	7	10	11	126

Table A.4: Scores for the structural recall for the *prosodic* condition of the lexical prosody (LP) group.

LP Prosodic: Content													
	LP1:2	LP2:2	LP3:2	LP4:2	LP5:2	LP6:2	LP7:2	LP8:2	LP9:2	LP10:2	LP11:2	LP12:2	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	0	1	1	1	1	1	1	1	1	1	1	1	11
Q3	1	1	1	1	1	1	1	1	1	1	1	1	12
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	1	0	1	1	1	1	1	0	0	0	1	0	7
Q6	1	1	1	1	1	1	1	1	1	0	0	1	10
Q7	1	1	1	1	1	1	1	1	1	1	1	1	12
Q8	1	1	0	1	0	0	0	0	0	0	1	0	4
Q9	1	1	1	1	1	0	1	0	1	0	1	0	8
Q10	1	1	1	1	1	0	1	0	0	1	1	0	8
Q11	1	1	1	1	0	0	0	1	0	0	0	1	6
Q12	1	1	1	1	1	1	1	1	1	1	1	1	12
TOTAL	11	11	11	12	10	8	10	8	8	7	10	8	114

Table A.5: Scores for the content recall for the *prosod ic* condition of the lexical prosody (LP) group.

LP Prosodic: Overall													
	LP1	LP2	LP3	LP4	LP5	LP6	LP7	LP8	LP9	LP10	LP11	LP12	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	0	1	1	1	1	1	1	1	1	1	1	1	11
Q3	1	1	1	1	1	1	1	1	1	1	1	1	12
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	1	0	1	1	1	1	0	0	0	0	0	0	5
Q6	1	1	1	1	1	1	1	1	1	0	0	1	10
Q7	1	1	1	1	1	1	1	1	1	1	1	1	12
Q8	1	0	0	1	0	0	0	0	0	0	1	0	3
Q9	1	1	1	1	1	0	1	0	1	0	1	0	8
Q10	1	1	1	1	1	0	1	0	0	0	1	0	7
Q11	1	1	1	1	0	0	0	1	0	0	0	1	6
Q12	1	1	1	1	1	1	1	1	1	1	1	1	12
TOTAL	11	10	11	12	10	8	9	8	8	6	9	8	114

Table A.6: Scores for the overall recall for the *prosodic* condition of the lexical prosody (LP) group.

A.2.3 Task Load Index Scores

the participants were presented with the following descriptions for the NASA task load index (TLX) factors:

mental demand Low-High How much mental and auditory activity was required? (e.g. thinking, deciding calculating, looking, listening, cross-monitoring and remembering)

time pressure Low-High How much time pressure did you feel because of the rate at which things occurred? (e.g. slow, leisurely, rapid, frantic)

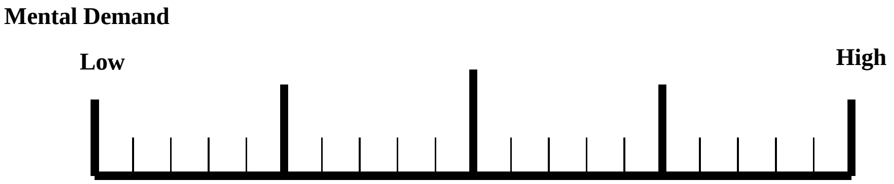
effort expended Low-High How hard did you work (mentally and physically) to accomplish your level of performance?

performance level achieved Poor-Good How successful do you think you were in accomplishing the mission goals?

frustration experienced Low-High How much frustration did you experience? (e.g., stress, irritation, annoyance, discouragement)

overall preference Condition One-Condition Two Rate the overall preference for the two conditions. With which one was the task the easiest?

The following scales were presented to the participants to mark scores for the TLX factors. The participants were asked to mark a cross on one of the upright lines on the scale.



A.2.4 Raw Scores for the TLX

LN Lexical: Structure													
	LN1:1	LN2:1	LN3:1	LN4:1	LN5:1	LN6:1	LN7:1	LN8:1	LN9:1	LN10:1	LN11:1	LN12:1	TOTAL
Q1	1	1	1	1	1	1	1	1	1	1	1	1	12
Q2	1	1	1	0	1	1	1	1	1	1	1	1	11
Q3	0	1	0	0	1	1	1	1	0	1	0	0	6
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	0	0	1	0	0	1	0	1	1	1	1	1	7
Q6	1	0	0	0	0	1	0	1	0	0	0	0	3
Q7	0	0	0	0	0	0	0	0	0	0	0	0	0
Q8	0	1	0	0	1	1	1	1	0	1	1	1	8
Q9	1	1	1	1	1	1	1	1	1	1	1	1	12
Q10	0	0	0	0	0	1	0	0	1	1	1	0	4
Q11	0	0	1	0	1	1	0	1	1	0	0	0	5
Q12	1	1	1	0	1	1	1	1	1	0	1	0	9
TOTAL	6	7	7	3	8	11	7	10	8	8	8	6	89

Table A.7: Scores for the structure recall for the *lexical* condition of the lexical no-cues (LN) group.

LN Lexical: Content													
	lN1:1	lN2:1	lN3:1	lN4:1	lN5:1	lN6:1	lN7:1	lN8:1	lN9:1	lN10:1	lN11:1	lN12:1	TotaL
Q1	1	1	0	1	1	1	1	1	1	1	1	1	11
Q2	1	1	1	1	1	1	1	1	1	1	1	1	12
Q3	0	0	0	0	1	0	0	0	0	0	0	0	1
Q4	0	1	0	1	1	0	1	1	1	1	1	1	9
Q5	0	0	1	0	0	0	0	0	0	0	0	0	1
Q6	0	0	0	0	1	0	0	0	0	0	0	0	1
Q7	0	1	0	0	0	0	0	0	0	0	0	0	1
Q8	0	1	0	0	1	0	0	1	0	0	0	0	3
Q9	1	1	1	1	1	1	1	1	1	1	1	1	12
Q10	0	0	0	0	0	0	0	0	0	0	0	0	0
Q11	0	1	1	0	0	1	1	1	0	1	1	0	7
Q12	1	1	0	0	1	0	1	1	1	0	1	0	7
TOTAL	48448467556465												

Table A.8: Scores for the content recall for the *lexical* condition of the lexical no-cues (LP) group.

LN Lexical: Overall													
	LN1	LN2	LN3	LN4	LN5	LN6	LN7	LN8	LN9	LN10	LN11	LN12	TOTAL
Q1	1	1	0	1	1	1	1	1	1	1	1	1	11
Q2	1	1	1	0	1	1	1	1	1	1	1	1	11
Q3	0	0	0	0	1	0	0	0	0	0	0	0	1
Q4	0	1	0	1	1	0	1	1	1	1	1	1	9
Q5	0	0	1	0	0	0	0	0	0	0	0	0	1
Q6	0	0	0	0	0	0	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	0	0	0	0	0	0	0
Q8	0	1	0	0	1	0	0	1	0	0	0	0	3
Q9	1	1	1	1	1	1	1	1	1	1	1	1	12
Q10	0	0	0	0	0	0	0	0	0	0	0	0	0
Q11	0	0	1	0	0	1	0	1	0	0	0	0	3
Q12	1	1	0	0	1	0	1	1	1	0	1	0	7
TOTAL	4	6	4	3	7	4	5	7	5	4	5	4	58

Table A.9: Scores for the overall recall for the *lexical* condition of the lexical no-cues (LN) group.

LN No-cues: Structure													
	LP1:1	LP2:1	LP3:1	LP4:1	LP5:1	LP6:1	LP7:1	LP8:1	LP9:1	LP10:1	LP11:1	LP12:1	TOTAL
Q1	0	0	1	0	1	1	1	0	0	0	0	1	5
Q2	0	0	0	0	0	0	0	0	1	0	0	0	1
Q3	1	1	1	1	1	1	1	1	1	1	1	1	12
Q4	1	1	1	1	1	1	0	1	1	1	0	1	10
Q5	0	0	1	0	0	0	0	0	0	0	0	0	1
Q6	0	0	0	0	0	0	1	0	0	0	1	0	2
Q7	0	0	1	1	1	1	0	0	1	0	0	1	6
Q8	0	0	0	0	0	0	0	0	0	0	0	0	0
Q9	0	0	0	0	0	0	0	0	0	1	1	0	2
Q10	0	0	0	0	0	0	0	1	0	0	0	0	1
Q11	0	1	0	1	1	0	0	0	0	0	0	0	3
Q12	1	1	1	1	1	1	0	1	1	1	0	1	10
TOTAL	3	4	6	5	6	5	3	4	5	4	3	5	53

Table A.10: Scores for the structure recall for the *no-cues* condition of the lexical no-cues (LN) group.

LN No-cues: Content													
	LN1	LN2	LN3	LN4	LN5	LN6	LN7	LN8	LN9	LN10	LN11	LN12	TOTAL
Q1	1	1	1	1	1	1	1	1	1	0	1	1	11
Q2	1	1	1	1	1	1	1	1	1	1	1	1	12
Q3	1	1	1	1	1	1	1	1	1	1	1	1	12
Q4	1	1	1	1	1	1	1	1	1	1	1	1	12
Q5	0	1	0	1	1	0	0	0	0	0	0	0	3
Q6	1	1	0	1	1	1	1	1	1	0	1	0	9
Q7	1	1	1	1	1	1	1	1	1	0	1	1	11
Q8	0	0	0	0	0	0	0	0	0	0	0	0	0
Q9	0	0	0	0	0	0	1	0	0	1	1	0	3
Q10	0	1	1	1	0	0	0	1	0	1	0	1	6
Q11	1	1	1	0	1	0	0	1	1	0	1	0	7
Q12	1	1	1	1	1	1	0	1	1	1	1	1	11
TOTAL	8	10	8	9	9	7	7	9	8	6	9	7	97

Table A.11: Scores for the content recall for the *no-cues* condition of the lexical no-cues (LN) group.

LN No-cues: Overall													
	LN1	LN2	LN3	LN4	LN5	LN6	LN7	LN8	LN9	LN10	LN11	LN12	TOTAL
Q1	0	0	1	0	1	1	1	0	0	0	0	1	5
Q2	0	0	0	0	0	0	0	0	1	0	0	0	1
Q3	1	1	1	1	1	1	1	1	1	1	1	1	12
Q4	1	1	1	1	1	1	0	1	1	1	0	1	10
Q5	0	0	0	0	0	0	0	0	0	0	0	0	0
Q6	0	0	0	0	0	0	1	0	0	0	1	0	2
Q7	0	0	1	1	1	1	0	0	1	0	0	1	6
Q8	0	0	0	0	0	0	0	0	0	0	0	0	0
Q9	0	0	0	0	0	0	0	0	0	1	1	0	2
Q10	0	0	0	0	0	0	0	1	0	0	0	0	1
Q11	0	1	0	0	1	0	0	0	0	0	0	0	2
Q12	1	1	1	1	1	1	0	1	1	1	0	1	10
TOTAL	3	4	5	4	6	5	3	4	5	4	3	5	51

Table A.12: Scores for the overall recall for the *no-cues* condition of the lexical no-cues (LN) group.

	TLX Factors					
Participants	Mental Demand	Time Pressure	Effort Expended	Performance Level	Frustration Experienced	Mean
LP1	16	13	11	8	9	12.2
LP2	16	15	11	7	13	13.6
LP3	14	18	10	4	20	15.6
LP4	18	18	16	5	13	16
LP5	16	18	16	9	13	14.8
LP6	15	18	10	9	16	14
LP7	10	3	3	12	0	4.8
LP8	18	15	15	6	15	15.4
LP9	18	17	18	5	17	17
LP10	18	18	15	9	15	15.4
LP11	16	8	12	14	7	9.8
LP12	16	6	11	1	20	14.4
means	15.92	13.92	12.33	7.42	13.17	

Table A.13: Individual and mean rating for TLX factors in lexical condition of LP group.

	TLX Factors						
Participants	Mental	Time	Effort	Performance	Frustration	Preferance	Mean
	Demand	Pressure	Expended	Level	Experienced		
LP1	13	11	9	10	11	15	11
LP2	12	11	11	11	8	15	11.4
LP3	16	8	14	12	4	20	12.4
LP4	16	14	14	15	4	17	13
LP5	13	12	13	14	7	20	13
LP6	13	15	7	15	10	15	11.4
LP7	1	1	1	16	0	18	8
LP8	15	13	13	10	13	15	12.2
LP9	18	14	17	11	14	15	13.2
LP10	15	15	9	10	14	12	11.2
LP11	16	6	9	14	5	12	9.4
LP12	14	6	11	4	16	20	11.4
means	13.50	10.50	10.67	11.83	8.83	16.17	

Table A.14: Individual and mean rating for TLX factors in prosodic condition of LP group.

	TLX Factors					
Participants	Mental Demand	Time Pressure	Effort Expended	Performance Level	Frustration Experienced	Mean
LN1	17	13	16	6	6	13.2
LN2	15	12	14	8	12	13
LN3	15	17	15	5	17	15.8
LN4	20	20	20	0	20	20
LN5	16	3	16	10	15	12
LN6	17	16	14	8	18	15.4
LN7	14	13	13	10	11	12.2
LN8	15	15	15	5	15	15
LN9	17	16	17	6	18	16.4
LN10	17	8	17	3	9	13.6
LN11	16	11	12	6	10	12.6
LN12	17	10	18	4	20	16.2
Means	16.33	12.83	15.58	5.92	14.25	

Table A.15: Individual and mean rating for TLX factors in lexical condition of LN group.

	TLX Factors						
Participants	Mental Demand	Time Pressure	Effort Expended	Performance Level	Frustration Experienced	Preference	Mean
LN1	12	11	13	11	11	13	11.4
LN2	13	11	13	10	9	14	11.6
LN3	14	15	13	7	13	14	12.4
LN4	15	15	15	8	15	20	14
LN5	11	3	16	14	4	18	11.4
LN6	16	17	15	7	19	2	10.8
LN7	16	13	15	8	10	8	11.2
LN8	10	13	13	8	16	13	11.8
LN9	13	11	11	8	9	13	11
LN10	12	4	15	4	5	14	10.6
LN11	9	10	10	10	4	15	11
LN12	10	5	10	10	10	17	10.4
means	12.58	10.67	13.25	8.75	10.42	13.42	

Table A.16: Individual and mean rating for TLX factors in no-cues condition of LN group.

Appendix B

The Mathtalk Browsing Language

action	Targets								
	Expression	**Term**	**Item**	**Level**	**Quantity**	**Fraction**	**Numerator**	**Denominator**	**Superscript**
Current	✓	✓	✓	✓	✓	✓	✓	✓	✓
Next	✓	✓	✓	✓	✓	✓			✓
Previous	✓	✓	✓	✓	✓	✓			✓
speak	✓				✓	✓	✓	✓	✓
Into					✓	✓	✓	✓	✓
Out-of				✓	✓	✓	✓	✓	✓
Beginning	✓	✓		✓	✓	✓	✓	✓	✓
End	✓	✓		✓	✓	✓	✓	✓	✓

Table B.1: Table of all valid command pairs in the browsing language for the Mathtalk program. An ‘✓’ represents a valid command pairing.

action	**Targets**								
	Expression	**Term**	**Item**	**Level**	**Quantity**	**Fraction**	**Numerator**	**Denominator**	**Superscript**
Current	✓	✓	✓	✓	✓	✓	✓	✓	✓
Next	✓	✓	✓	✓	✓	✓			✓
Previous	✓	✓	✓	✓	✓	✓			✓
Which	✓								
Show			✓		✓	✓	✓	✓	✓
Into			✓		✓	✓	✓	✓	✓
Out-of				✓	✓	✓	✓	✓	✓
Beginning	✓	✓		✓	✓	✓	✓	✓	✓
End	✓	✓		✓	✓	✓	✓	✓	✓
Glance	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table B.2: Table of all valid command pairs in the final version of the browsing language for the Mathtalk program. An '✓' represents a valid command pairing. The action **glance** was added to accomodate the audio glance described in Chapter 5.

Appendix C

Algebra Earcons

C.1 Multiple Choice Questions

C.1.1 Simple Condition

1. **A** $a/b = c$

   B $a = bc$

   C $\boxed{ab = c}$

   D $a + b = c$

2. **A** a^b

   B $\boxed{ab^c}$

   C ab

   D $a + b^c$

3. **A** $\boxed{ab + cd}$

   B $ab + c$

   C $ab = cd$

   D $a + bc$

4. **A** $a^b = c^d$

   B $ab + c^d = e^f$

   C $\boxed{a^b + c^d = e^f}$

   D $a^b + c = d^e$

5. **A** $\boxed{ax^2 + bx + c = d}$

   B $ax^2 + bx + c$

   C $ax + bx + c = d$

   D $ax^2 + c = d$

6. **A** $\frac{ab}{c} = \frac{d}{e}$

   B $\boxed{\frac{ab}{c} + \frac{d}{e} = f}$

   C $\frac{ab}{cd} = e$

   D $\frac{ab}{c} + \frac{d}{e}$

7. **A** $ab + \frac{c}{d} = e$

   B $\frac{a}{b} + cd = e$

   C $\frac{a}{bc} + d = e$

   D $\boxed{\frac{a}{b} + \frac{c}{d} = e}$

8. **A** $a = b + c$

   B $a + bc = d$

   C $a = bc^d$

   D $\boxed{a = bc + d}$

9. **A** $\boxed{ax^4 + bx^3 + cx^2 + dx + e = 0}$

   B $ax^3 + bx^2 + cx + d = 0$

   C $ax^4 + bx^3 + cx^2 + dx + e$

   D $0 = a + bx + cx^2 + dx^3 + ex^4$

10. **A** $\boxed{ab+cd=e}$

B $a+bc=d$

C $ab+cd$

D $ab+cd^{e}$

11. **A** $ab^{c}=d$

B $ab=c^{d}+e$

C $\boxed{ab^{c}=d+e}$

D $a^{b}=c+d$

12. **A** $\boxed{a+b=cd}$

B $a=b+cd$

C $ab=cd$

D $a+b=c$

13. **A** $\frac{ab}{cd+ef}=g+h$

B $\frac{a}{b}=c+d$

C $\boxed{\frac{a}{b}+\frac{cd}{ef}=g+h}$

D $\frac{a}{b}+\frac{cd}{ef}=g$

14. **A** $abc^{d}+ef^{g}$

B $\boxed{abc^{d}+ef^{g}=h}$

C $ab^{c}+d^{e}=f$

D $ab^{c}+de^{f}=g$

15. **A** $\boxed{ax^3+bx^2+cx+d}$

B x^3+ax^2+bx+c

C $ax^2+bx+cx+d$

D $ax^3+bx^2+cx+d=e$

C.1.2 Complex Condition

1. **A** $(a+b)^{c-d}$

   B $\frac{a+b}{c-d}$

   C $\boxed{(a+b)(c-d)}$

   D $(a+b(c-d))$

2. **A** $a=\frac{b}{c^d+ef+g}$

   B $a=bc^d-ef+g$

   C $a+b(c^d-ef+g)$

   D $\boxed{a=b(c^d-ef+g)}$

3. **A** $\frac{ab}{c+d}\times(e+f)$

   B $\boxed{\frac{ab}{(c+d)(e+f)}}$

   C $\frac{ab}{c+d}$

   D $\frac{(a+b)(c+d)}{ef}$

4. **A** $a+(b^c-d)^{e+f}$

   B $a(b^c-d)^e+f$

   C $\boxed{a+(b^c-d)^e+f}$

   D $(b^c-d)^e+f$

5. **A** $\boxed{\frac{a}{b}(cd+e)=f+g}$

   B $ab(cd+e)=f+g$

   C $\frac{a}{b}+(cd+e)=f+g$

   D $\frac{a}{b}(cd+e)+fg$

6. **A** $(ab+c)^{de}+f$

   B $\boxed{(ab+c)^{de+f}}$

   C $(ab+c)(de+f)$

   D $\frac{ab+c}{de+f}$

7. **A** $a = \frac{bc}{d+(e^g-hij)}$

B $a = (b+(c^d-efg))(hi)$

C $\frac{-a+(b^c-def)}{gh} = i$

D $\boxed{a = \frac{b+(c^d-efg)}{hi}}$

8. **A** $\boxed{a = (b+c)^d - e}$

B $a = (b+c)^{d-e}$

C $a = (b+c^d) - e$

D $a(b+c)^d - e$

9. **A** $a(bc+d-e-f)^g = h$

B $a(x+b)(x-d)^e = f$

C $\boxed{a(b+c) - d(e-f)^g = h}$

D $a(b+c)^d - e(f-g) = h$

10. **A** $a+b^{c+d}$

B $(a+b)^c + d$

C $a = b^{c+d}$

D $\boxed{(a+b)^{c+d}}$

11. **A** $\boxed{a(b+c(d+ef)) = g}$

B $a(b+c(d+ef)) + g$

C $ab + c(d+ef) = g$

D $a(b+c(d+ef))^g$

12. **A** $ab(c+d) + \frac{ef}{g+h}$

B $a+b\,\frac{(c+d)+ef}{g+h}$

C $\boxed{a+b(c+d)+\frac{ef}{g+h}}$

D $a+(b+c)+\frac{de}{f+g}$

13. **A** $(a+bc+de)^f = g$

B $(a+b)^c(d+e) = f$

C $(a+b)(c+d)^e = f$

D $\boxed{(a+b)^c(d+e)^f = g}$

14. **A** $\boxed{\frac{a}{b+c}(\frac{de}{f})^g}$

B $(\frac{de}{f})^g$

C $(\frac{ab+c}{d+ef})^g$

D $\frac{a}{b+c}(\frac{de}{f})$

15. **A** $(a+b)+(c+d)(e+f)^{g+h} = i^j$

B $(a+b)+\frac{c}{d}(e+f)^{g+h} = i$

C $\boxed{(a+b)+\frac{c}{d}(e+f)^{g+h} = i^j}$

D $\frac{a}{b}(c+d)^{e+f} = g^h$

C.2 Raw Scores for Experiment one

Participants	Questions															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
Correct	c	b	a	c	a	b	d	d	a	a	c	a	c	b	a	
E1	a	b	a	c	a	b	d	b	a	a	c	d	c	d	c	10.00
E2	c	b	a	c	a	b	d	d	b	a	c	a	c	d	a	13.00
E3	c	b	a	c	a	c	d	c	a	a	c	a	a	d	a	11.00
E4	c	b	a	c	a	c	d	d	a	a	c	a	c	b	c	13.00
E5	c	b	a	c	d	a	d	b	c	a	c	a	c	b	a	11.00
E6	a	b	a	b	c	b	d	b	a	a	a	a	d	b	b	8.00
E7	c	d	a	c	a	b	d	d	b	a	c	a	c	b	c	12.00
E8	c	b	a	c	d	b	d	d	a	a	c	a	c	d	a	13.00
E9	c	b	a	c	a	c	d	d	c	a	c	a	d	b	a	12.00
E10	d	b	a	c	a	c	d	d	a	a	c	c	a	c	b	9.00
E11	c	b	a	c	c	b	d	b	a	a	a	a	a	b	a	11.00
E12	c	b	a	c	a	c	d	b	a	a	a	a	d	b	a	11.00
Total	9	11	12	11	8	6	12	6	8	12	9	10	6	7	7	

Table C.1: Raw scores for the simple condition of experiment one.

Participants	Questions															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
Correct	c	d	b	c	a	b	d	a	c	d	a	c	d	a	c	
E1	c	d	c	d	b	b	d	c	c	d	d	b	d	d	d	7
E2	c	d	b	c	b	b	d	a	c	d	a	c	d	a	b	13
E3	c	d	b	c	c	b	d	b	c	d	a	a	d	c	c	11
E4	c	d	b	b	c	b	d	c	c	d	a	c	d	a	c	12
E5	c	d	c	b	b	b	a	c	c	d	a	a	d	a	a	8
E6	c	d	b	a	a	b	b	a	c	d	c	c	c	a	c	11
E7	d	d	b	c	b	b	d	a	c	d	a	c	d	a	a	12
E8	c	d	b	c	c	b	d	a	c	d	a	c	d	a	b	13
E9	c	d	b	a	b	b	d	a	c	d	a	a	d	a	c	12
E10	c	d	b	c	b	b	d	a	b	d	a	c	d	a	b	12
E11	b	c	c	b	a	b	d	b	c	d	b	a	d	d	c	7
E12	c	d	c	b	a	d	d	a	b	d	a	a	b	a	b	9
Total	10	11	8	5	3	11	10	7	10	12	9	6	10	9	5	

Table C.2: Raw scores for the complex condition of the first experiment.

C.3 Raw Scores for Experiment Two

Participants	Questions														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
Correct	c	b	a	c	a	b	d	d	a	a	c	a	c	b	a	
E4	c	b	a	c	a	b	d	d	b	a	c	a	D	B	D	12
E7	C	b	A	C	A	S	D	D	A	A	C	A	C	d	A	14
E1	C	b	A	C	A	S	D	C	A	A	C	D	D	D	b	10
E8	C	b	A	C	A	S	D	D	b	A	C	A	C	C	A	13
E11	D	b	A	C	C	S	D	b	A	A	C	A	D	D	A	10
E5	C	b	A	C	A	S	D	b	b	A	b	A	C	C	A	11
Total	5	6	6	6	5	6	6	3	3	6	5	5	3	1	4	

Table C.3: Raw scores for the simple condition of the second algebra earcons trial.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
Correct	c	d	b	c	a	b	d	a	c	d	a	c	d	a	c	
E4	C	D	B	B	A	B	C	C	C	D	A	b	D	A	b	10
E7	C	D	B	b	C	B	D	A	C	D	A	b	D	A	C	12
E1	C	D	B	A	b	B	D	A	C	D	C	C	b	A	D	10
E8	C	D	B	C	A	B	D	A	C	D	A	C	D	A	b	14
E11	C	A	B	C	A	B	D	A	C	D	A	A	D	D	b	11
E5	C	D	B	C	C	B	A	C	C	D	A	A	D	A	A	10
Total	6	5	6	3	3	6	4	4	6	6	5	2	5	5	1	

Table C.4: Raw scores for the complex condition of the second algebra earcons trial.

Appendix D

Final Evaluation

D.1 Keystrokes used

The initial letters of the commands are taken from the first letter of the *action* words given below and the second letters from the first letter of the *targets*. The actions are: Show, current, next, previous, beginning, end, into, a out-of and glance.

The targets are: Expression, term, item, level, superscript, fraction, numerator, denominator and quantity.

F1		F2		F3		F4	
C	**F**	**C**	**F**	**C**	**F**	**C**	**F**
Default	49	Default	113	Default	9	Default	58
Multiple	18	Multiple	14	Multiple	13	Multiple	18
Errors	15	Errors	0	Errors	11	Errors	13
ge	35	ne	30	ne	47	cl	52
ce	24	ce	29	ge	40	ge	45
ni	23	ge	17	ce	21	ne	36
ne	20	we	13	nt	26	sq	15
cl	15	be	8	cl	20	sf	15
pe	11	cl	6	be	17	ce	13
be	10	pe	5	ni	15	we	13
we	9	ct	2	we	11	nf	12
sq	5	nq	1	if	9	nq	12
nq	5	sq	1	nq	9	be	9
cf	3			iq	9	pe	8
gl	3			nf	7	nt	6
ct	2			id	7	sd	4
sf	2			ol	7	ci	2
ee	2			cq	6	ct	2
ci	1			in	5	sn	2
cq	1			of	4	id	1
if	1			oq	4	in	1
iq	1			sq	4	of	1
nf	1			ct	3	om	1
nt	1			pt	3	oq	1
				sf	2	bq	1
				ss	2		
				gq	2		
				pe	2		
				is	1		
				om	1		
				os	1		
				pf	1		
				cf	1		
				ci	1		
				cn	1		

Table D.1: Frequency table of commands used in the Mathtalk condition of the final evaluation. A key for the command names may be seen in the text. **C** = Command and **F** = Freqeuncy.

F1		F2		F3		F4	
Keystroke	Frequency	**Keystroke**	Frequency	**Keystroke**	Frequency	**Keystroke**	Frequency
Next-word	101	Previous-char	90	Next-line	11	Next-line	62
Next-char	224	Next-char	467	Next-char	90	Next-char	278
Previous-line	88	Next-line	47	Previous-line	11	Previous-line	66
Next-line	75	Previous-line	27	Line-start	9	Previous-char	57
Previous-char	88	Line-start	20	Previous-line	6	Line-start	24
Previous-word	48	Document-top	5	Previous-word	9		
Line-end	10	Document-end	3	Next-word	5		
Document-top	5	Next-word	1	Document-start	1		
Line-end	1	Document-end	2				
Totals	642		661		127		549

Table D.2: Frequency of keystrokes used in the word-processor condition of the final evaluation.

Mathtalk Condition											
	Task Number										
Participant	1	2	3	4	5	6	7	8	9	10	Mean
F1	28	185	17	12	41	94	2	14	31	44	46.8
F2	25	20	34	42	31	276	50	115	280	58	93.1
F3	17	20	13	18	20	79	145	77	55	10	45.4
F4	16	24	21	42	45	127	514	50	22	37	89.8

Table D.3: Task times in seconds for the navigation tasks of the Mathtalk condition.

Word-processor Condition											
	Task Number										
Participant	1	2	3	4	5	6	7	8	9	10	Mean
F1	39	46	203	43	73	127	88	27	51	162	85.9
F2	24	121	284	92	107	287	227	42	164	51	139.9
F3	33	23	123	162	43	103	20	43	55	26	63.1
F4	109	66	172	73	114	68	5	47	58	27	73.9

Table D.4: Task times in seconds for the navigation tasks of the word-processor condition.

Mathtalk Condition												
participant	Task Number											
	1	2	3	4	5	6	7	8	9	10	11	Mean
F1	441	86	40	115	49	20	35	50	73	124	56	99.00
F2	207	78	129	45	28	57	59	133	150	241	29	105.09
F3	47	151	121	53	23	42	53	56	111	43	35	66.82
F4	268	73	171	66	159	31	35	36	84	137	31	99.18

Table D.5: Task times in seconds for the evaluation tasks of the Mathtalk condition.

Word-processor Condition												
Participant	Task Number											
	1	2	3	4	5	6	7	8	9	10	11	Mean
F1	22	48	61	104	43	70	120	80	99	74	223	85.82
F2	56	96	23	177	188	183	132	108	118	73	52	109.64
F3	30	57	102	166	52	66	102	82	70	119	33	79.91
F4	65	111	61	158	50	58	146	119	231	65	53	101.55

Table D.6: Task times in seconds for the evaluation tasks of the Word-processor condition.

Factor	F1	F2	F3	F4	Total	Mean
Mental Demand	7	4	5	12	28	7.0
Time Pressure	4	5	7	10	26	6.50
Expended	4	3	5	2	14	3.5
Perceived Performance	10	17	12	10	49	12.3
Frustration Experienced	5	2	4	0	11	2.8
Preference	10	17	17	20 64	16	

Table D.7: Raw scores for the TLX factors in the Mathtalk Condition of the final evaluation.

Factor	F1	F2	F3	F4	Total	Mean
Mental Demand	15	14	10	20	59	14.75
Time Pressure	12	10	8	3	33	8.25
Effort Expended	5	14	10	10	39	9.75
Perceived Performance	10	14	15	10	49	12.25
FrustrationExperienced	0	15	7	20	42	10.50

Table D.8: Raw scores for the TLX factors in the word-processor Condition of the final evaluation.

Bibliography

Aldrich, F. and A. Parkin (1988). Improving the retention of aurally presented information. In M. Gruneberg, P. Morris, and R. Sykes (Eds.), *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*, pp. 187–196.

Baddeley, A. D. (1986). *Working Memory*. Oxford University Press.

Baddeley, A. D. (1990). *Human Memory: Theory and Practice*. Lawrence Erlbaum Associates Ltd.

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

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

Bates, M. J. (1990). Where should the person stop and the information search interface start? *Information Processing and Management 26*(5), 575–591.

BAUK (1987). *Final Draft Version of Braille Mathematics Notation*. Braille Authority of the United Kingdom, RNIB, Peterborough.

Bauwens, B., J. Engelen, and F. Evenepoel (1994). Structuring documents: The key to increasing access for the print disabled. In W. L. Zagler, G. Busby, and R. L. Wagner (Eds.), *Computers for Handicapped Persons: Proceedings of ICCHP '94, Lecture Notes in Computer Science 860*, pp. 214–221. Berlin: Springer-Verlag.

Beard, D. V. and J. Q. Walker (1990). Navigational techniques to improve the display of large two dimensional spaces. *Behaviour and Information Technology 9*(6), 451–466.

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

Berkeley Speech Technology (1986). *Berkeley Best Speech Synthesizer*. Berkeley, California, USA: Berkeley Speech Technology.

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

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

Bolinger, D. (1972). Accent is predictable, if you're a mind reader. *Language 48*(3), 633–644.

Borgman, C. L. (1986). The user's mental model of an information retrieval system: An experiment on a prototype online catalogue. *International Journal of Man-Machine Studies 24*, 47–64.

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

Boyd, L. H., W. L. Boyd, and G. C. Vanderheiden (1990). The graphical user interface: Crisis, danger, and opportunity. *Journal of Visual Impairment and Blindness 25*(Dec.), 498–502.

Braselton., S. and B. C. Decker (1984). Using graphic organisers to improve the reading of mathematics. *The Reading Teacher 48*(3), 276–281.

Brewster, S. A. (1994). *Providing a Structured Method for Integrating Non-speech Audio into Human-computer Interfaces*. Ph. D. thesis, Department of Computer Science, University of York, UK.

Brewster, S. A., V.-P. Raty, and A. Kortekangas (1995). Representing complex hierarchies with earcons. Technical report ERCIM-05/95R037, ERCIM.

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

Brewster, S. A., P. C. Wright, and A. D. N. Edwards (1994b). The design and evaluation of an auditory enhanced scrollbar. In B. Adelson, S. Dumais, and J. Olson (Eds.), *Celebrating Interdependence: Proceedings of Chi '94*, pp. 173–179. ACM Press: Addison-Wesley.

Buxton, W., W. Gaver, and S. Bly (1991). Tutorial no. 8: The use of non-speech audio at the interface. In *CHI'91 Conference proceedings, Humantems*. ACM Press: Addison-Wesley.

Byers, J., A. Bittner, and S. Hill (1989). Traditional and raw task load index (tlx) correlations: Are paired comparisons necessary? In A. Mital (Ed.), *Advances in Industrial Ergonomics*, pp. 481–485. Taylor and Francis.

Cahill, H. and G. Boormans (1994). Problem analysis: A formative evaluation of the mathematical and computer access problems as experienced by visually impaired students. Technical Report Tide Maths project 1033 D1, Tide Office, Brussels, University College Cork, Ireland.

Carroll, L. (1982). Through the looking-glass. In *The Complete Illustrated Works of Lewis Carroll*. Chancellor Press.

Chang, L. A. (1983). *Handbook for Spoken Mathematics (Larry's Speakeasy)*. Lawrence Livermore Laboratory, The Regents of the University of California.

Crispien, K. and H. Petrie (1993). Providing access to GUIs for blind people. In *19th Convention of the Audio Engineering Society*.

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

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

Crystal, D. (1987). *The Cambridge Encyclopædia of Language*. Cambridge University Press.

Deutsch, D. (1982). *Psychology of Music*. Academic Press, London.

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

Edwards, A. D. N. (1993). International workshop on access to mathematics. Unpublished Workshop Proceedings; available from A. D. N. Edwards, Dept. Computer Science, University of York, York, UK YO1 5DD.

Edwards, A. D. N., I. J. Pitt, S. A. Brewster, and R. D. Stevens (1995). Multiple modalities in adapted interfaces. In A. D. N. Edwards (Ed.), *Extra-Ordinary Human-Computer Interaction*, pp. 221–244. New York: Cambridge University Press.

Edwards, A. D. N. and R. D. Stevens (1993, March). Mathematical representations: Graphs, curves and formulas. In D. Burger and J.-C. Sperandio (Eds.), *Non-visual Human-Computer Interactions: Prospects for the Visually Handicapped*, pp. 181–194. Proceedings of the INSERM Seminar Non-visual Presentations of Data in Human-computer Interactions: John Libbey Eurotext.

Edwards, A. D. N. and R. D. Stevens (1994). 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*, Université Pierre et Marie Curie, B23, 9 Quai Saint-Bernard, 75252, Paris Cedex 05 and ANPEA (ISSN 0010-2520), pp. 97–104. INSERM.

Edwards, A. D. N., R. D. Stevens, and I. J. Pitt (1995). Représentation non visuelle des mathématiques. In A. B. Safran and A. Assimacopoulos (Eds.), *Le DÉficit Visuel, Editions Masson*, pp. 169–178. translated by A. Assimacopoulos.

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

Elovitz, H., R. Johnson, A. McHugh, and J. Shaw (1976). Letter-to-sound rules for automatic translation of English text to phonetics. *IEEE Transactions on Acoustics, Speech, and Signal Processing 24*(6), 446–459.

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

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

Gerth, J. M. (1992). *Performance Based Refinement of a Synthetic Auditory Ambience: Identifying and Discriminating Auditory Sources*. Ph. D. thesis, Georgia Institute of Technology.

Gilmore, D. J. (1986). Structural Visibility and Program Comprehension. In M. D. Harrison and A. F. Monk (Eds.), *People and Computers: Designing for Usability*, BCS Workshop Series, pp. 527–545. Cambridge University Press.

Griffith, D. (1990). Access to computers by blind people: Human factors issues. *Human Factors 32*(4), 467–475.

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

Handel, S. (1989). *Listening: An Introduction to the Perception of Auditory Effects*. MIT press, Cambridge, Massachusetss.

Harling, P. A., R. Stevens, and A. Edwards (1995). Mathgrasp: The design of an algebra manipulation tool for visually disabled mathematicians using spatial-sound and manual gestures. Submitted to the HCI Journal special issue on multi-modal interfaces.

Hart, S. and L. Staveland (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, pp. 139–183. Amsterdam: North Holland B.V.

Hart, S. G. and C. Wickens (1990). Workload assessment and prediction. In H. R. Booher

(Ed.), *MANPRINT, an Approach to Systems Integration,*, pp. 257–296. New York: Van Nostrand Reinhold.

Harter, S. T. and A. Rogers-Peters (1985). Heuristics for online information retrieval: a typology and preliminary listing. *Online Review 9*(5), 407–424.

Hartley, J. (1980). Spatial cues in text. *Visible Language 14,* 62–79.

Howson, G. (1991). *National Curricula in Mathematics*. The Mathematical Association.

Hulme, C. (1984). Reading: Extracting information from printed and electronically presented text. In A. Monk (Ed.), *Fundamentals of Human Computer Interaction*, pp. 35–47. Academic Press Inc (London) Ltd.

ISO-9241 (1993, May). *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Part 11:Guidance on specifying and measuring usability*. International Standards Organization. Committee Draft.

Kerr, S. T. (1990). Wayfinding in an electronic database: The relative importance of navigational clues vs. mental models. *Information Processing and Management 26*(4), 511–523.

Kim, Y. and S. B. Servais (1985). Vocational, educational, and recreational aids for the blind. In J. G. Webster, A. M. Cook, W. J. Tompkins, and G. C. Vanderheiden (Eds.), *Electronic Devices for Rehabilitation*, pp. 101–115. Chapman and Hall.

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.

Kwasnik, B. H. (1992). A descriptive study of the functional components of browsing. In J. Larsen and C. Unger (Eds.), *Engineering for Human Computer Interaction*, pp. 191–203. Elsevier Science Publishers B.V. (North Holland).

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

Landau, B. (1988). The construction and use of spatial knowledge in blind and sighted children. In J. Stiles-Davis, M. Kritchevsky, and U. Bellugi (Eds.), *Spatial Construction: Brain Bases and Development*, pp. 343–371. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

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

Lehiste, E. (1970). *Suprasegmentals*. MIT Press.

Lowenfeld, B. (1980). Psychological problems of children with impaired vision. In W. Cruickshank (Ed.), *Psychology of Exceptional Children and Youth*, pp. 211–307. NJ: Prentice Hall.

Luce, P. A. and T. Feustel (1983). Capacity demands in short term memory for synthetic and natural speech. *Human Factors 25*(1), 17–32.

Lunney, D. and R. C. Morrison (1981). High technology laboratory aids for visually handicapped chemistry students. *Journal of Chemical Education 58*(3), 228–231.

Lyons, J. (1979). *Introduction to Theoretical Linguistics*. Cambridge University Press.

Mansur, D. L., M. Blattner, and K. Joy (1985). Sound graphs: Numerical data-analysis method for the blind. *Journal of Medical Systems 9*, 163–174.

Marchioni, G. and B. Shneiderman (1988). Finding facts vs. browsing knowledge in hypertext systems. *Computer 19*(1), 70–80.

Miller, G. A. (1956). The magical number seven; plus or minus 2. *Psychological Review 63*(2), 81–97.

Monahan, D. (1985). Teaching mathematics to visually impaired pupils.

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

Morrison, R. E. and A. W. Inhoff (1981). Visual factors and eye movements in reading. *Visible Language 15*, 129–146.

Murray, I. R., J. L. Arnott, and A. F. Newell (1988). Hamlet— simulating emotion in synthetic speech. In *Proceedings of Speech 88*, pp. 1217–1223.

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

Nakatani, L. H. and J. Schaffer (1978). Hearing words without words: Prosodic cues for word perception. *Journal of the Acoustical Society of America 63*, 234–244.

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

Nemeth, A. (1972). *The Nemeth Braille Code for Mathematics and Science Notation*. Louisville, Kentucky: AAWB-AEVH Braille Authority.

Nespor, M. and I. Vogel (1986). *Prosodic Phonology*. Foris.

Nielsen, J. (1990). The art of navigating through hypertext. *Communications of the ACM 33*(3), 298–310.

Norman, D. A. (1988). *The Psychology of Every Day Things*. Basic Books.

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

Ostendorf, M., S. Shattuck-Hufnagel, and C. Fogg (1991). The use of prosody in syntactic disambiguation. *Journal of the Acoustic Society of America 19*(6), 2956–2969.

Pike, K. (1945). *The Intonation of American English*. University of Michigan Press.

Pisoni, D. B., H. C. Nusbaum, and B. G. Greene (1985). Perception of synthetic speech generated by rule. *Proceedings of the IEEE 11*, 1665–1676.

Ralsten, J. V., D. Pisoni, S. E. Lively, B. G. Green, and J. B. Moulinix (1991). Comprehension of synthetic speech produced by rule. *Human Factors 33*(4), 471–491.

Raman, T. (1991). TEXtalk. *TUGboat 12*(1).

Raman, T. V. (1992, October). An audio view of latex documents. *TUGboat 13*(3), 372–377.

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

Raman, T. V. (1994b). Interactive audio documents. In *Assets '94: The First Annual ACM Conference on Assistive Technolgies*, pp. 62–68.

Ranney, M. (1987). The role of structural context in syntax in the recognition of algebraic expressions. *Memory and Cognition 15*(1), 29–40.

Rapp, D. W. and A. J. Rapp (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 A. Pollatsek (1989). *The Psychology of Reading*. Prentice Hall.

Reich, S. (1980). Significance of pauses for speech perception. *Journal of Psycholinguistic Research 9*(4), 379–389.

Rosson, M. B. (1985). Using synthetic speech for remote access to information. *Behaviour Research Methods: Instruments and Computers 17*(2), 250–252.

Scholl, G. T. (1993, June). Educational programs for blind children: A kaleidoscopic view. *Journal of Visual Impairment and Blindness*, 177–180.

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

Schwab, E., H. C. Nusbaum, and D. Pisoni (1985). Some effects of training on the perception of synthetic speech. *Human Factors 27*(4), 395–408.

Shneiderman, B., P. Shafer, R. Simon, and L. Weldon (1986). Display strategies for program browsing: Concepts and experiment. *IEEE Software 3*(5), 7–14.

Slowiaczek, M. L. and C. Clifton (1980). Sub-vocalisation and reading for meaning. *Journal of Verbal Learning and Verbal Behaviour 19*, 573–582.

Slowiaczek, M. L. and H. C. Nusbaum (1985). Effects of speech rate and pitch contour on the perception of synthetic speech. *Human Factors 27*(6), 701–712.

Smither, J. (1993). Short term memory demands in processing synthetic speech by old and young adults. *Behaviour and Information Technology 12*(6), 330–335.

Southall, R. (1988). Visual structure and the transmission of meaning. In J. C. van Vliet (Ed.), *Document Manipulation and Typography. Proceedings of the International Conference on Electronic Publishing, Document Manipulation and Typography.*, pp. 35–45. Cambridge University Press.

Stevens, R. D. (1991). Spoken mathematics. Master's thesis, Department of Biology, Department of Computer Science, University of York, UK. Masters dissertation: Internal publication.

Stevens, R. D., S. A. Brewster, P. C. Wright, and A. D. N. Edwards (1994). Design and evaluation of an auditory glance at algebra for blind readers. In G. Kramer (Ed.), *Auditory Display: The Proceedings of the Second International Conference on Auditory Display.* Addison-Wesley.

Stevens, R. D. and A. D. N. Edwards (1993, January). A sound interface to algebra. In *Proceedings of the IEE Colloquium on Special Needs and the Interface*. IEE Digest no. 1993/005.

Stevens, R. D. and A. D. N. Edwards (1994a). Mathtalk: The design of an interface for reading algebra using speech. In W. L. Zagler, G. Busby, and R. L. Wagner (Eds.), *Computers for Handicapped Persons: Proceedings of ICCHP '94, Lecture Notes in Computer Science 860*, pp. 313–320. Berlin: Springer-Verlag.

Stevens, R. D. and A. D. N. Edwards (1994b, November). Mathtalk: Usable access to mathematics. *Information Technology and Disabilities 1*. On-line Journal Available on Internet, see `http://www.rit.edu/˜easi/easijrnl/easijrnl.html`).

Stevens, R. D. and A. D. N. Edwards (1996, April). An approach to the evaluation of assistive technology. In *Assets '96*, pp. llli–64–71. New York: ACM.

Stevens, R. D., P. C. Wright, and A. D. N. Edwards (1994). Prosody improves a speech based interface. In D. England (Ed.), *Ancillary Proceedings of HCI'94*. London: British Computer Society.

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*, 1582–15:1 92.

Sumikawa, D., M. Blattner, and R. Greenberg (1986). Earcons: Structured audio messages. Unpublished paper.

Sumikawa, D., M. Blattner, K. Joy, and R. Greenberg (1986). Guidelines for the syntactic design of audio cues in computer interfaces. Technical Report UCRL 92925, Lawrence Livermore National Laboratory.

Sumikawa, D. A. (1985). Guidelines for the integration of audio cues into computer user interfaces. Technical Report UCRL 53656, Lawrence Livermore National Laboratory.

't Hart, J. and A. Cohen (1973). Intonation by rule: A perceptual quest. *Journal of Phonetics 1*, 309–327.

Tessler, L. (1981, August). The smalltalk environment. *Byte*, 90–147.

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

Vaissiere, J. (1983). Language-independent prosodic features. In A. Cutler and D. R. Ladd (Eds.), *Prosody: Models and Measurements*, pp. 53–66. Springer-Verlag, Berlin.

Vanderhieden, G. C. (1989). Non-visual alternative display techniques for output from graphics based computers. *Journal of Visual Impairment and Blindness 83*(8), 383–390.

Vincent, T. (1982). Computer assisted support for blind students - the use of a microcomputer linked voice synthesizer. *Computers and Education (GB) 6*(1), 55–60.

Wallace, J. N. and T. A. B. Wesley (1992, July). The application of information technology to the access of mathematical notation for the blind. In W. Zagler (Ed.), *3rd International Conference on Computers for Handicapped Persons*, R. Oldenbourg, Vienna, pp. 562–569.

Waterworth, J. A. (1983). Effect of intonation form and pause durations of automatic telephone number announcements on subjective preference and memory performance. *Applied Ergonomics 14*, 39–42.

Waterworth, J. A. (1987). The psychology and technology of speech, symbiotic relationship. In J. A. Waterworth (Ed.), *Speech and Language Based Interaction with Machines: Towards the Conversational Computer*. John Wiley.

Wenzel, E., F. Wightman, and D. Kistler (1991). Localization with non-individualized virtual display cues. In *CHI'91 conference*

Proceedings, pp. 351–359. ACM Press, Adison Wesley.

Wright, P. C. and A. F. Monk (1989). Evaluation for design. In *People and Computers 5, Proceedings of HCI'95*, pp. 345–358. Cambridge University Press.

Zhang, J. and D. A. Norman (1994). Representations in distributed cognitive tasks. *Cognitive Science 18*, 87–122.