Lambda: A multimodal approach to making mathematics accessible to blind students

Alistair D N Edwards
University of York
Department of Computer Science
York, UK, YO10 5DD
+44 1904 432775
alistair@cs.york.ac.uk

Heather McCartney
University of York
Department of Computer Science
York, UK, YO10 5DD

Flavio Fogarolo
Ministry of Education
CSA of Vicenza
Italy
flavio.fogarolo@tin.it

ABSTRACT

The study of mathematics is all but precluded to most blind students because of the reliance on visual notations. The Lambda System is an attempt to overcome this barrier to access through the development of a linear mathematical notation which can be manipulated by a multimodal mathematical editor. This provides access through braille, synthetic speech and a visual display. Initial results from a longitudinal study with prospective users are encouraging.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human factors, **K.3.1 [Computer Uses in Education], K.4.2 [Social Issues]:** Assistive technologies for persons with disabilities.

General Terms

Human Factors.

Keywords

Mathematics education, blind students, synthetic speech, braille, MathML.

1. INTRODUCTION

Mathematics relies on visual notations. Whenever mathematicians communicate there is always a chalkboard or pencil and paper nearby, and there is also a need for written representations whenever an individual is performing mathematics. Thus, anyone who does not have access to written notations is at a serious disadvantage in the study of mathematics. This applies in particular to blind people. This paper reports the results of the *Lambda* Project which has attempted to provide non-visual alternatives through the use of information technology.

2. BACKGROUND

Before describing the technology developed within the Lambda Project, it is necessary to fill in and clarify a certain amount of background information regarding mathematics and access to materials for blind people.

2.1 Mathematical notations

There are a number of features of mathematics which make the use of visual notations both vital and appropriate. Correspondingly a variety of notations and representations have evolved. For instance, algebra is a means of representing many of the abstractions of mathematics. Meaning in mathematics is very precise and the notation reflects that; there is never redundancy in a mathematical expression and two syntactically similar expressions can have completely different meanings. At the same time the syntax of algebra has evolved such that it make good use of the medium – which amounts to a two-dimensional planar representation.

Take the following well-known equation as an example:

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

Nothing can be omitted from that equation if it is to retain its meaning. Visually the following Equation (2) is very similar; mathematically it is completely different. (The fact that the horizontal bar of the square root symbol does not extend over the term $4ac$ means that it does not apply to that term). Any alternative notation (such as a non-visual one) must maintain these features.

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

Notice also the compactness of the equations. That the right-hand side is a fraction is immediately apparent and it is easy to recognize its numerator and denominator. The expression is also persistent. The mathematician can use the power of vision to extract the meaning that is pertinent to the current task. In trying to solve the equation (by substituting for *a, b* and *c*) the mathematician does not have to remember very much of the equation, because writing it down serves as a form of *external memory*[13].

There is an interesting discussion to be had as to what extent an external representation forms part of the mathematician's cognitive model of the mathematics [8], and thus to what extent the absence of such a representation inhibits a person's ability to perform mathematics. However, that discussion is beyond the scope of this paper. What is clear is that blind people do not have access to the visual notations – with all their advantages. The question is to what extent alternative, non-visual notations can substitute. Given the power of vision, it seems certain that any alternative needs to make of as many alternative modalities as possible [3]. Hence an ICT-based approach, such as that embodied by the Lambda Project, is appropriate.

2.2 Non-visual alternative modalities

The best-known non-visual notation for blind people is braille. Braille mathematical notations exist and they are a vital part of the Lambda solution, but braille is not without its problems. Conventional braille consists of *cells* of up to six tactile dots, arranged in a 2 x 3 matrix, as in Figure 1. Six dots can represent 2^6 = 64 distinct symbols, which is perfectly adequate to represent literary texts (essentially made up of 26 letters, 10 digits and a small number of punctuation symbols).

Figure 1. The six dots of a conventional braille cell.

The 64 distinct braille cells are more than adequate to represent literary texts, but mathematics requires many more symbols. There are two approaches to extending the scope of braille (both of which are used in the Lambda System). The first is to use combinations of cells to represent a single mathematical symbol. As a simple example (in most braille codes), the digits 1, 2,..0 are not represented by distinct braille cells. Rather the same cells as are used to represent the letters A, B,..J are used, preceded by a special *number* sign. Thus, 1 is #A, 2 is #B and so on. (See Figure 2).

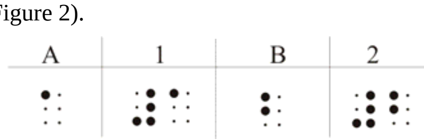

Figure 2. Braille representation of letters and numbers. Digits 1, 2,…0 are represented by the same cells as the letters A, B,…J preceded by a special *number* sign.

In 6-dot braille further multiple cells have to be used to represent the full range of mathematical symbols. So, for instance, in UK Mathematical braille [10], the basic operators +, – and x are each represented by a pair of cells and more complex (and obscure) symbols may require three or more cells. Thus, it takes 31 braille cells to represent Equation (1).

The main problem with the extended versions of braille for mathematical use is that there is no one agreed notation. Essentially each country has its own version of mathematical braille. Even English-speaking countries which share the same (or almost the same) literary braille systems (such as the UK and the USA) have their own versions of mathematical braille. It is ironic that whereas mathematics is often seen as an international language, independent of spoken languages, when translated into braille that universality is lost.

Computers have revolutionized the use of braille in two forms. Firstly, the computer can be used to generate braille texts. Any electronic document can be printed directly by a computer attached to a braille *embosser* printer. The software can take care of translating the text into braille. This includes not only the translation that would be required, for instance, to represent digits in the above example by pairs of cells, but also more complex translations into compressed ‘contracted’ braille codes. Printed braille is known as *hard* braille, by analogy with ‘hard copy’ printed text and is ‘printed’ by braille embossers. The second form of braille used with computers is ‘soft’ braille, used to represent information from a computer screen. A soft braille display consists of a set of cells of pins that can be raised and lowered under the computer’s control [17], thereby generating readable braille. As electro-mechanical devices, braille displays are expensive and so they are usually available as a single line of 40 or 80 cells. In other words, they represent a one-dimensional display, not like the two dimensions of a screen or printed paper.

The Lambda Project is sponsored by the European Commission and it is thus most important that it be as widely applicable across the European Union as possible. To that end a great deal of effort has been expended on making materials available in a wide variety of national braille codes, as explained further in Section 3.

Another approach to the limited space of symbols in a six-dot notation, is to increase the number of dots in a braille cell to eight and this is also exploited in the Lambda system. Lambda is an acronym of *Linear Access to Mathematics for Braille Device and Audio Synthesis* and the word ‘linear’ is important. Braille is a linear notation. There has been an attempt to devise a two-dimensional braille notation for mathematics (*DotsPlus*, [11]) but it can only generate hard copy output and cannot be displayed on a single braille line. Thus there is a need for a mathematical notation which is compact and linear, and that is the objective of the *Lambda Code*, which has been developed within the project. The Lambda Code is a new mathematical notation, based on 8-dot braille. Fortunately most braille displays do incorporate 8-dot cells.

The Lambda Code uses single symbols (visual and braille) to represent each mathematical symbol. This means that the notation is linear, compact and easy to examine in auditory, tactile and visual forms. It is a universal code in that it is not tied to any one national braille code.

Notice that the Lambda acronym also mentions audio synthesis. In other words, the Lambda approach is multimodal; braille is supplemented by a vocal representation of the mathematics. Again, speech is essentially a linear medium in time and so the same linearization of the material is necessary. Each Lambda Code symbol has a spoken representation. Screenreaders enable blind people to access computers. These transform the contents of the computer screen into either braille or speech – or both [14]. Using a mouse is out of the question, so blind users rely on keyboard alternatives [18]. The keyboard does not pose a substantial problem; touch-typing is a skill routinely taught to blind people.

2.3 Conventional approaches to teaching mathematics

Few blind students attain a high level of competence in mathematics (though there are some notable exceptions, [6]). This is at least partly due to the barriers to access to notations.

Where mathematics is taught braille is often the main medium of record and communication, usually generated by conventional braille typewriters (i.e. without the use of any ICT).

Frequently teachers avoid the complexities of the formal mathematical braille codes by using simpler, self-defined encodings. Where necessary (particularly for examinations), these will be transcribed by the teacher into conventional print notation for access by sighted readers (i.e. examiners). The use of non-standard notations naturally limits the level of mathematics that can be studied, but that is not a problem for most students, who only want to attain a minimum qualification (in the UK, usually no further than the basic General Certificate of Secondary Education, or GCSE, normally taken at age 16).

Those students who do want to study to school-leaving level (known in the UK as Advanced Level) or beyond will generally learn a standard maths braille code. An alternative approach is to use a word processor on text files containing mathematical material expressed in *Latex* [7]. The advantage of this approach is that conventional print rendering of the mathematics can be easily generated. The main disadvantage is that Latex is designed to facilitate the manipulation of the visual representation of the material not the mathematical meaning (i.e. it is a printer's markup language). Other research on ways of making Latex mathematics more accessible can be found in [15], [9] and [4].

3. LAMBDA

Having set out the background in terms of technology and current approaches to mathematical education, we can describe how the Lambda System attempts to address the problems.

MathML [12] is an XML-based notation for representing mathematics. It is a comprehensive notation with a wide coverage of mathematical expression. There are two styles of representation: *presentation* and *content.* 'Presentation elements describe mathematical notations' visually oriented two-dimensional structure.'[1], whereas content elements are based on the mathematical semantics.

While MathML is a comprehensive notation it is also complex and hierarchically structured. Lambda required a linear notation. The Lambda Code is based on MathML but is simpler and linear, yet is easy to translate back and forth between Lambda and MathML. Another advantage of using MathML is that translators already exist for conversion between it and other popular mathematical notations, notably Tex and Latex.

The Lambda Code includes new symbols which make it possible to represent the mathematics in a linear form. The symbols can be rendered visually and in braille. An important aspect of the braille notation is that it is based on 8-dot braille cells. This means that nearly all of the Lambda symbols can be represented by a single cell and that there is a one-to-one correspondence between the braille cells and the visual symbols.

Figure 3 shows how Equation (1) is rendered visually in Lambda Code. The Lambda symbols used are explained in Table 1, which also shows the braille representation.

[1] http://www.w3.org/TR/MathML2/chapter2.html#id.2.1.1

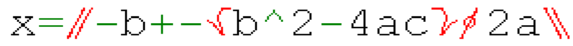

Figure 3. Equation (1) as rendered in the Lambda Code. The Lambda symbols used are explained in Table 1.

That the Lambda braille code is new is both an advantage and a disadvantage. An advantage is its universality. Unlike existing mathematical brailles, it is international. On the other hand, it is another braille notation to be learned by the user. Naturally it has been designed to bear as much resemblance as possible to conventional codes to facilitate learning. At the same time, it is possible to translate from the Lambda code into conventional (6-dot) braille for printing on an embosser. A great deal of effort has been expended within the project to the building of a database which embodies most of the major braille mathematical codes which can then be used as the basis of translation. This means that users can get hard copy of mathematical materials in the (6-dot) code with which they are familiar.

Table 1. The Lambda Code symbols use in Figure 4, their braille and spoken representation

Sym-bol	Lambda braille cell	Speech	Meaning
//		*compound fraction*	Open compound fraction
ø		*denominator*	Numerator -denominator separator
\\		*end fraction*	Close compound fraction
√		*compound root*	Open compound square root
⟩		*close compound root*	Close square root
^		*to the power of*	Simple power
+-		*plus or minus*	Plus or minus

Another very important component of the Lambda System is the Editor (Figure 4). This uses the Lambda Code directly. The Lambda Code uses special symbols to represent mathematics. Those symbols can be rendered in print, speech or in 8-dot braille (via a screenreader[2]). For instance, Equation (1) would be rendered visually as in Figure 3. The spoken representation also relies on a simple one-to-one translation (See Table 1). The spoken representation is based on [1] and Figure 3 is spoken by Lambda as:

[2] Currently the Editor works only with the Jaws screenreader, but it is being developed to be compatible with other popular screenreaders.

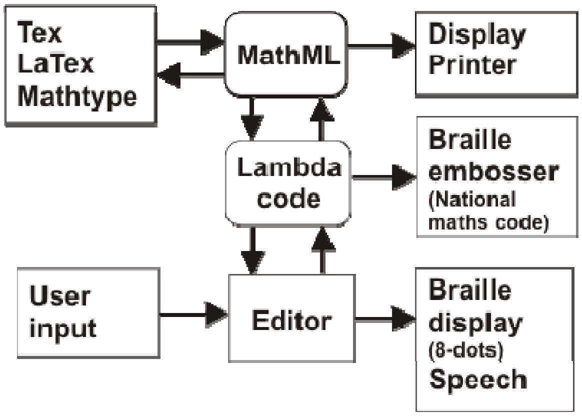

Figure 4. The Lambda system.

x equals the fraction, numerator minus b plus or minus compound root b to the power of 2 minus 4 a c close compound root, denominator 2 a, end fraction

Notice that the equation is represented in 20 8-dot Lambda code cells, in comparison with the 31 cells required in UK Mathematical braille. Given that most braille displays have just 40 cells, such a reduction can be important.

The braille and speech are accessible to the blind student and have a direct correspondence to the visual representation – which is intended mainly for the convenience of sighted teachers. Similarly with the addition of mathematical visualization software, *MathPlayer*[3] it is possible to render the Lambda Code in conventional, two-dimensional notation.

The Lambda Editor resembles a text editor, but has been designed specifically to work on mathematics expressed in the code. It is a wysiwyg GUI editor which runs under Microsoft Windows and can be used by blind people with the assistance of a screenreader.

Figure 5 illustrates how Equation (1) might be solved in the Lambda editor for a = 1, b = 5, c = 6. Notice how lines in the solution are duplicated, giving a reference line and a working line.

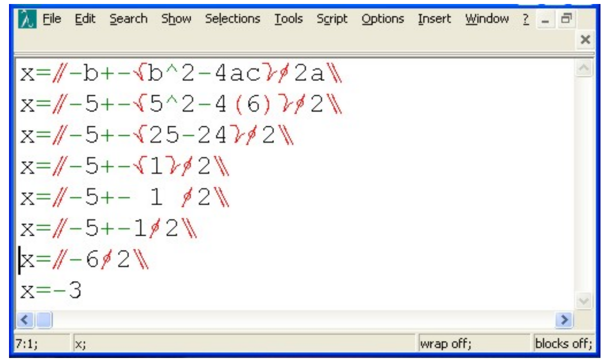

Figure 5. Solving Equation 1 in the Lambda Editor, for *a* = 1, *b* = 5, *c* = 6.

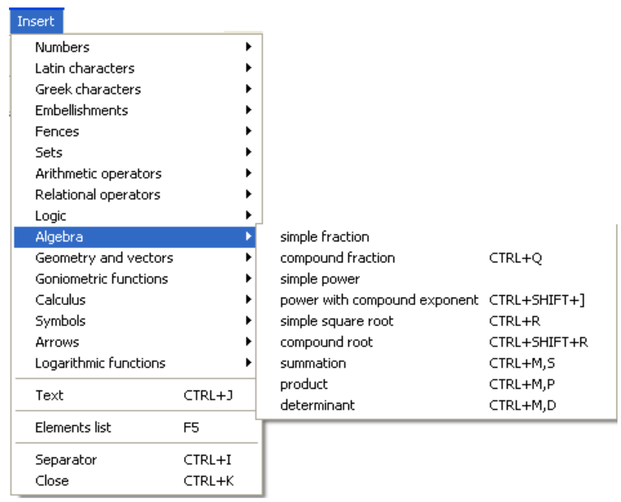

Figure 6. The menu mechanism for insertion of Lambda Symbols. The symbols are grouped according to their mathematical context. Note that the menu entries tend to be full names – which can be expressed unambiguously in speech, but that there are also a rich set of keyboard alternatives for expert users.

Input can be generated from the keyboard but with additional facilities for the insertion of Lambda symbols. There are several alternative ways of generating the symbols, but the most accessible method for blind users is menu-based. The Insert menu is illustrated in Figure 6 and shows how symbols are grouped mathematically. Notice also that many of the menu entries have keyboard shortcuts ('hot keys'). Given the large number of symbols, the keyboard sequences can be quite complex.

[3] Available free-of-charge from http://www.dessci.com/en/products/mathplayer/download.htm

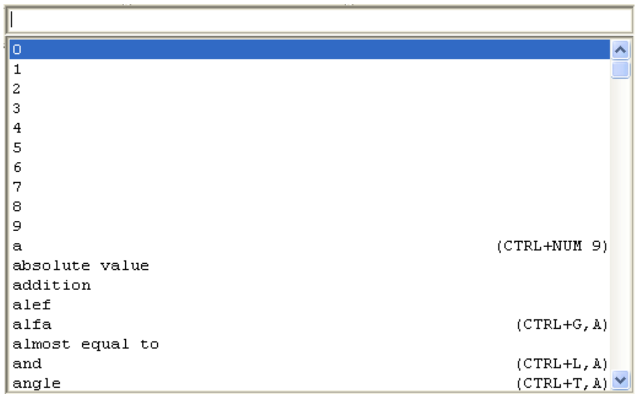

Figure 7. The Element List menu. When first opened the menu contains all of the available symbols, but as the user types, the number of elements is cut down, as in Figure 7.

Should the user not know which group is likely to contain the required symbol and hence be unable to find it in the menu, then he or she can select the Element List entry in the Insert menu. This leads to an alphabetical menu (Figure 7) which can also be searched textually. For instance, if the user wants to insert a greater-than sign, then starting to type 'greater than' into this menu will reduce it to a small number of selections (Figure 8).

These features are all designed to ease input for blind users. Another feature assists in the closure of complex constructs. Typing *Ctrl-k* will cause the most deeply nested compound construct (e.g. a compound square root or fraction) to be closed. Similarly, some structures have two components (such as the numerator and denominator of a fraction). Typing *Ctrl-i* inserts the appropriate separator.

As well as the Insert menu, there are graphical palettes from which the user can choose symbols to insert. This will only be accessible to sighted users, but is nevertheless considered important to support teachers using the system.

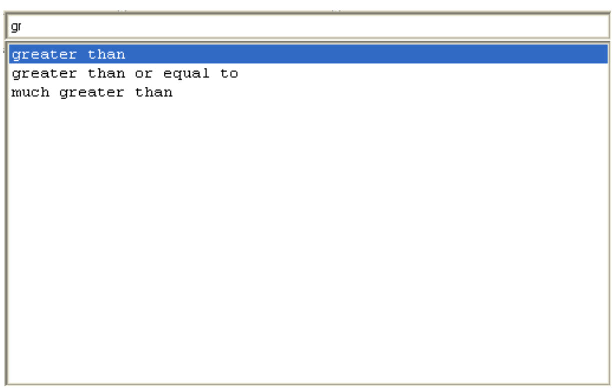

Figure 8. As the user types the letters 'gr' of 'greater than', the elements in the menu narrow down.

Solving a mathematical equation usually involves writing down different versions of the equation each one (ideally) being a simplification of the previous one. The Lambda Editor is designed to facilitate such an incremental approach through the inclusion of a simple *duplicate* command. This creates two new copies of the current line, one which the student can edit and work on and one to which he or she can refer back as an aid to remembering the content of the current line.

One powerful feature is a calculator function. Any expression in which all the operands are numbers (i.e. all the variables have been substituted by numerical values) can be calculated by the editor, so that the expression can easily be replaced by its value. This feature could be used in the example in Figure 5. The terms in fraction in the penultimate line (⫽−6∤2⑊) are entirely numerical, so the value can be calculated for the user, using the calculator function. While this might seem to be giving the blind student an unfair advantage, in practice it can relieve them of a lot of mechanical effort that would be involved in remembering a (possibly complex) expression and evaluating it.

x=⫽∤⑊

Figure 9. A *compressed* view of the equation in Figure 4 (and Equation 1).

The editor has a function to compress and expand structures. By compressing a structure the user can hide details and get a higher-level overview of an expression. For instance, compressing the equation in Figure 3 results in the representation in Figure 9, which would be spoken as '*x* equals the fraction, numerator, denominator end fraction.' This corresponds to the kind of overall impression that a sighted mathematician might get by glancing at an equation.

Otherwise the editor works much as a text editor. Material can be selected (using a variety of methods) and be cut, copied and pasted. Find and Replace functions are also available. These do not require the entry of the search string via a separate dialogue, but rather the currently selected text is automatically the target string. Non-mathematical text can also be included, so that accompanying commentary can be included. Material can also be imported into the editor (Figure 4). Thus, textbooks and other materials (which might originally have been in a notation such as Latex) can be imported for the student to read and work on within the editor.

In principle there is almost no limit to the flexibility of the editor, since it has a scripting facility built in, based on the Python language. This facility has been used to implement a number of additional commands in response to user feedback. These include Invert Fraction, Goto Next Denominator and Goto Previous Denominator.

4. RELATIONSHIP TO OTHER WORK

Lambda is not the first attempt to address these challenges using IT. Notably the Maths Project [16] had many of the same objectives and was also based on a combination of braille and sounds. There are a number of differences between the two approaches, however. These include:

Technology

Both projects recognized the importance of using a well-defined, standard representation. Maths used SGML, but by the time Lambda came into existence, the rather more appropriate MathML had been devised.

Linearity

The Maths Project attempted to use parallelism in sounds (speech and non-speech). This was not entirely successful in that testers tended not to use it [16] and the Lambda Project chose to concentrate on linearity.

Braille codes

The Maths Project sidestepped the problem of multiple mathematical braille codes by standardizing on one code (*Eurobraille*). The Lambda Project has put great effort into supporting multiple braille codes (in hard copy), although it has also introduced a new standard for soft display.

Educational philosophy

The Maths Project was based on the idea that it was providing an equivalent of paper and pencil, that there should be no mathematical knowledge built in to the system so that the student should perform all the mathematics. While the motivation was understandable, in practice 'simple' manipulations are so difficult in a non-visual mode that it seems reasonable to provide some support. Thus, for example, the Lambda Editor calculator function will calculate the value of any expression in which all the operands are numerical[4].

5. EVALUATION

There are a number of problems in attempting to evaluate any innovation such as Lambda. Specifically, the difficulties of studying mathematics for blind people have already been discussed. The result is that there are very few competent blind mathematicians – and hence few suitable test participants. Looking at it another way, since there is no existing comparable technology it is not possible to perform any kind of controlled test. As with most educational innovations, the ultimate objective is to improve the educational attainment of students. This makes evaluation very difficult because controlled testing is very difficult. If one runs a within-group study and measures the performance before and after the 'treatment', then how does one know that the same change would not have occurred if conventional educational approaches had been used? To use a cross-group study, one needs well-matched groups of participants. Finding *any* suitably qualified blind mathematics students is hard; finding two well matched groups is impossible.

On the other hand, it has been suggested that the simplified notation of the Lambda Code would be of benefit to the more general teaching of mathematics – including sighted students. This might give the opportunity for more of a controlled evaluation.

In these terms, it would be difficult to generate any quantitative evidence of the efficacy of the Lambda System. Instead it was decided to stage a number of Lambda *Information Days*, in the different countries represented in the Lambda consortium. Students, their parents and teachers are invited to attend the day. There each student is given an individual introduction to the use of the Editor and then encouraged to explore it and its capabilities. Feedback is obtained through observation and questionnaires. At the end of the day they are given a copy of the software and encouraged to use it at home and school. One month later, the participant is contacted by telephone and asked to provide further feedback.

Evaluation started in 2004, as soon as a prototype of the software was available. As such, the evaluation is an integral part of the iterative development of an ever-improving product. Many enhancements have incorporated in successive versions of the software. The project has been extended to the end of 2006 and evaluations are still continuing. Nevertheless, some initial observations and conclusions have already been drawn. By the end of the project further data will be available and a more full account of the evaluation will be published.

It is apparent that the Lambda Editor is well used by students who are already comfortable with using the PC. Those who are not, really need to master the appropriate computer skills so that they can concentrate on the mathematics when using Lambda. Similarly, users who are already accustomed to using computer braille displays seem to have little difficulty in using and learning the 8-dot Lambda Code. It was noted above that the keyboard shortcut key sequences can be quite convoluted. Nevertheless, it has been observed that this is the preferred method of entry for many users once they have gained some experience. Evidently they find the overhead of remembering the sequences less of an effort than that required to go through the sequence of actions to make menu selections [5, 19]. The facility to easily close and delimit components of expressions (i.e. the *Ctrl-k* and *Ctrl-i* commands) is universally found to be useful.

None of the testers has (yet) used the Editor through speech alone. This might be seen as providing support for the choice of such a multimodal approach, except that many testers preferred not to use speech at all, to rely on the braille alone and to switch the speech off. It is thought that this might be partly due to the environment in which they were working in which participants feared that sound might disturb others working around them. In such cases, they were encouraged to try the speech when they were working in a private environment, at home. Braille clearly has the advantage of conveying spatial information and it seems that users are able to build up a good representation of the two dimensions of a page, even through a single line of braille.

It has been interesting to observe some of the tactics and techniques that users have developed in using the editor. These are generally shortcuts through which information can be picked up quickly or ways of rapidly navigating the cursor to the required position.

One technique is the use of space characters for highlighting. In conventional braille it is often convenient to place spaces around a symbol of interest. When that item is to be re-located, the finger can quickly scan to the blank space. Accordingly spacing is maintained in the Lambda Editor and spaces have no syntactic

[4] Naturally this function can be disabled as might be necessary in certain educational contexts, such as in some examinations.

meaning. This also facilitates another tactic that has been observed. The Duplicate command is used frequently. When working on the second line, users often like to maintain the vertical alignment of symbols. That way they can quickly switch between the two lines and compare corresponding terms. In order to achieve this, students will use the spacebar as a means of deleting material, effectively over-writing it with spaces. (See the fifth line of the solution in Figure 5).

The Editor treats the screen like a sheet of paper, in that text can be placed anywhere on it. For instance, a new entry may be placed to the right of an existing line or at the bottom of a page with blank space from the existing lines. These features allow different students to apply different techniques. Some prefer to start each calculation on a new line, others prefer to build to the right of the existing line and yet others like to open new windows. They can then work in a way analogous to their sighted peers, with one 'best' page and another for rough work.

6. CONCLUSIONS

While the image of mathematics is poor and most adults seem almost proud of how little of it they learned in school (at least in the UK)[5] in truth anyone who is denied access to the subject is severely disadvantaged. To not have at least a basic qualification (GCSE) in the subject is a severe limitation and a higher level of qualification is a prerequisite to many other areas of education and employment. There have been a number of prominent mathematicians who were blind [6] but it is clear that they have been exceptional and have had to struggle to get over the problem of just accessing – and creating and manipulating mathematical material. The Lambda Project represents a new approach to the problem. It is based upon what is mathematically necessary but adapted to the abilities of blind people.

The positive reception of Lambda augers well for the future in which the availability of the system may have a significant effect on the education of blind people.

7. ACKNOWLEDGEMENTS

The Lambda Project is funded by the European Union through the Information Society Technologies initiative, Action Line: Persons with special needs, including the disabled and the elderly - Systems for independent living IST PROGRAMME - 2001-37139. Partners in the project are: Arca Progetti SRL (Italy), Associação dos Cegos e Ambliopes de Portugal, Dodecanese Association in Rhodes (Greece), European Blind Union (France), Italian Library of the Blind, Ministero dell'Istruzione (Italy), Moscow Logos Center (Russia), Organización Nacional de Ciegos Españoles (Spain), Royal National Institute of the Blind (UK), Unione Italiana Ciechi – Verona (Italy), Université Paul Sabatier Irit-Tobia (France), Università Statale di Milano (Italy), Universität Stuttgart (Germany) and University of York (UK).

Evaluation copies of the Lambda Editor are freely available from the project web site: http://www.lambdaproject.org/

8 REFERENCES

[1] Chang, L.A., *Handbook for Spoken Mathematics (Larry's Speakeasy)*, Lawrence Livermore Laboratory, The Regents of the University of California

[2] Cockcroft, W.H., *Mathematics Counts: Report of the Committee of Enquiry into the Teaching of Mathematics in Schools*. 1982, London: HMSO. pp.311.

[3] Edwards, A.D.N., et al., *Multiple modalities in adapted interfaces*, in *Extra-Ordinary Human-Computer Interaction*, A.D.N. Edwards, Editor. 1995, Cambridge University Press: New York. pp. 221–244.

[4] Fitzpatrick, D. and Monaghan, A.I.C. *TechRead: A system for deriving braille and spoken output from Latex documents*. in *Computers and Assistive Technology, ICCHP '98: Proceedings of the XV IFIP World Computer Congress*. 1998. Vienna & Budapest: Austrian Computer Society.

[5] Grudin, J. and Norman, D.A. *Language evolution and human-computer interaction*. in *Proceedings of the 13th Annual Conference of the Cognitive Science Society*. 1991: Lawrence Earlbaum Associates.

[6] Jackson, A., *The world of blind mathematicians.* Notices of the AMS, 2002. **49**(10): pp. 1246-1251.

[7] Lamport, L., *Latex: Users guide and reference manual*. 1988, New York: Addison Wesley.

[8] Netz, R., *The Shaping of Deduction in Greek Mathematics: A study in cognitive history*. 2003: Cambridge University Press. pp.349.

[9] Raman, T.V., *Audio Systems for Technical Reading*. 1994, Cornell University.

[10] RNIB, *Braille Mathematics Notation 1987*. 1987, Peterborough: Royal National Institute for the Blind. pp.51.

[11] Sahyun, S.C. and Gardner, J.A. *Audio and haptic access to Maths and Science: Audio Graphs, Triangle, the MathPlus Toolbox and the Tiger Printer*. in *Computers and Assistive Technology, ICCHP '98: Proceedings of the XV IFIP World Computer Congress*. 1998. Vienna & Budapest.

[12] Sandhu, P., *The MathML Handbook*. 2003, Hingham, Massachusetts: Charles River Media. pp.518.

[13] Schönfplug, W., *The trade-off between internal and external information storage.* Journal of Memory and Language, 1986. **25**: pp. 657-675.

[14] Schwerdtfeger, R., *Making the GUI Talk.* Byte Magazine, 1991(December): pp. 118–128.

[15] Stevens, R., *Principles for the design of auditory interfaces to present complex information to blind computer users*. 1996, University of York, UK.

[16] Stevens, R.D., Edwards, A.D.N., and Harling, P.A., *Access to mathematics for visually disabled students through multi-modal interaction.* Human-Computer Interaction, 1997. **12**(1 & 2): pp. 47–92.

[17] Weber, G., *Reading and pointing – New interaction methods for braille displays*, in *Extra-ordinary Human-Computer Interaction: Interfaces for Users with Disabilities*, A.D.N. Edwards, Editor. 1995, Cambridge University Press: New York. pp. 183–200.

[18] Williamson, R., *Microsoft Windows without a mouse: An accessibility guide*, University of York, Department of Computer Science, http://www-users.cs.york.ac.uk/~alistair/mouselessmanual.pdf

[19] Zipf, G.K., *Human Behavior and the Principle of Least Effort*. 1949, New York: Hafner. pp.573.

[5] For example [2] concluded (p.7) 'The extent to which the need to undertake even an apparently simple and straightforward piece of mathematics could induce feelings of anxiety, helplessness, fear and even guilt in some of those interviewed was, perhaps, the most striking feature of the study.'