Supporting Blind Students in Navigation and Manipulation of Mathematical Expressions: Basic Requirements and Strategies

Bernhard Stoeger, Mario Batusic, Klaus Miesenberger, and Philipp Haindl

Institute of Integrated Study, Johannes Kepler University Linz
Altenbergerstrasse 69, 4020 Linz, Austria
{bernhard.stoeger, mario.batusic, k.miesenberger, philipp.haindl}@jku.at

Abstract. In [10], the problems faced by a blind or visually impaired student in doing Mathematics were analyzed, and the basic ideas of a MAWEN (Mathematical Working Environment), a software solution to help overcome these pressing difficulties, were described. The present paper builds upon the latter one, refining the ideas sketched there. After a thorough description of the state of the art, we present some general considerations on the problems met by a blind pupil when navigating within mathematical expressions and when doing calculations. Finally, through several case studies taken from mainstream school books, strategies to provide computer aided support to overcome the problems are outlined.

1 Introduction

In [10], four basic problems faced by a blind or visually impaired student working in Mathematics were formulated:

– Access to mathematical literature (books, teaching materials, papers etc.)
– Preparation of mathematical information (presenting school exercises, writing papers etc.)
– Navigation in mathematical expressions
– Actually doing Mathematics (carrying out calculations and computations at all levels, doing formal manipulation, solving exercises)

A fifth basic problem, which is closely related to all the above mentioned, is to be added: The problem of mathematical communication between sighted and blind/visually impaired people, typically, communication between a sighted teacher and a blind/visually impaired pupil.

Whereas some work towards the first and second problems done in [8] led to practical solutions, problems 3 to 5, although touched by several research initiatives, are still waiting for solutions applicable in practice.

Our idea to design a comprehensive multi-modal software solution to address the above listed problems, especially problems 3 to 5, which we call a MAWEN (Mathematical Working Environment), was outlined in [10]. the present paper is to refine the ideas set forth there, by:

K. Miesenberger et al. (Eds.): ICCHP 2006, LNCS 4061, pp. 1235–1242, 2006.
© Springer-Verlag Berlin Heidelberg 2006

– analyzing current mathematical school book literature, in order to isolate some of the implicit methodology used by mainstream mathematical teaching;
– providing some more case studies than in [10]
– discussing several key problems in navigation through complex mathematical expressions
– describing several standard mathematical exercises, outlining the problems met by a blind or visually impaired pupil in doing them, and discussing supportive tools to perform these exercises, to be integrated in a MAWEN

2 State of the Art

One of the first steps to be taken to mathematically enable blind people was the creation of so-called "mathematical notations". Most of them are codes that build upon traditional Braille [12] to [9]. They have the advantage that they are quite well optimized for a Braille user, but they do not permit easy sharing of information between blind and sighted people. One of their further disadvantages is the fact that a very large number of such codes have been developed over time - every attempt towards a unification or standardization of Braille maths codes failed by now.

In order to make communication somewhat easier, several notations based on ASCII were defined [13]. Such ASCII maths notations enable communication between blind and sighted people through a compromise: Whereas the sighted partner does not see a formula in its familiar graphical rendering, the blind partner gives up the optimization of maths notation towards Braille. On the other hand, most ASCII maths notations are easy to learn, so they are practicable to some extent.

One special case of an ASCII notation is the source code of the TeX typesetting system, especially that of the widely used LaTeX macro package. Since, at least after some revision, mathematical documents in LaTeX source are accessible to the blind, LaTeX offers a partial solution to basic problem number 1. This is, to a limited extent, also true for problem number 2, since a student who writes his/her exercises in LaTeX source needs just to invoke the LaTeX compiler in order to have a well-formatted and very attractive hard copy of his/her work for the sighted teacher.

In order to get more practical solutions out of braille maths codes, ASCII math codes, and LaTeX source, two initiatives were started: the LabraDoor software package [8] developed by our research group converts LaTeX documents into the Marbourg System [3], one of the most widely used Braille codes in German speaking countries, or into HRTeX, a new notation that resembles TeX, but eliminates some of that code's more awkward aspects. LabraDoor is now the basis for the production of school books for blind pupils in Austria, largely overcoming problems 1 and 2 on a primary, secondary, and high school level.

The second initiative comes from the University of Pierre and Marie Curie in Paris. It is called UMCL [1], which means "Universal Math Conversion Library".

It is a C API for converting various maths notations between each other, together with several concrete conversions already implemented. In the end, UMCL shall allow every partner involved in mathematical communication, blind or sighted, to use the notation of his/her choice through the UMCL and the applications designed for it shall do the necessary conversions in the background, via an internal representation of maths expressions following the MathML standard from W3C [14].

Whereas problems 1 and 2 will be largely overcome by the initiatives listed above, much more work has to be done towards problems 3 to 5. One first step addressing problem 3 was taken by the University of South Florida, who are developing the MathGenie, a tool that enables one to browse through a mathematical expression by synthetic speech [6]. It is one of the goals of MAWEN to implement navigation support that is to work for a Braille user as well.

A tool that might support a pupil both in navigation and manipulation, i.e., would address problems 3 and 4, is the Virtual Pencil, developed by the company Henter Math, LLC [5]. It addresses the needs of people who are unable to operate a pencil effectively, which is a key part in learning and dealing with maths. As it virtually takes on the role of a pencil while at the same time avoiding an overload of the student with unsolicited educational support it is an appropriate solution for in-class maths education.

A further project on access to Mathematics is Lambda [7], which provides a braille and audio device based system for the management of scientific documents. Hereby the focus clearly is put on a widely customizable user interface, which allows the pupils to edit mathematical notations as well as supports the learning progress by offering different interfaces for team and software collaboration for interactive maths education.

A further project on the topic is Infty [4], which is dedicated to the development of a software for mathematical OCR. They also build a tool to input and to edit mathematical formulae based on voice output.

Finally, we would like to mention the EU-funded project MICOLE [2], which means "Multimodal Environment for Inclusion of Visually Impaired Children". Much work undertaken by us for the MAWEN is being carried out within MICOLE, the MAWEN being one component of the environment to be designed within this project.

3 Research Methodology

In order to supply usable support of a blind individual in his/her mathematical tasks, it is of key importance to gain an understanding of the methods currently exercised by blind people working in Mathematics. As begun in [10], these methods have to be analyzed, exposing the problems, and inefficiencies, presently met when applying them. Hereafter, suggestions to improve the methods in order to better meet the needs of the target group have to be worked out.

Since present mathematical teaching, also that applied to blind pupils, is strongly influenced by visual presentation, the programme outlined above will need an understanding of the methodology currently applied by sighted people when doing mathematical exercises, methodology currently conveyed by today's mathematical teaching. It is strange that such methodology appears not to be reasonably documented in the literature. Rather, the methodology seems to be implicit in teaching, especially in school books. We are therefore analyzing some of these books, in order to isolate that implicit methodology. This analysis shall lead to a deeper understanding of the way how Mathematics is presently done. It shall be the basis for the formulation of a set of functions, also called support or helper routines, to be implemented into a software in order to aid a blind or visually impaired student in his/her mathematical exercises. In the MAWEN, these functions shall be implemented, and some of them shall be combined into wizards, as shown, e.g., in our MICOLE prototypes [2].

4 Findings

4.1 Considerations on Text Selection

When dealing with mathematical formulae on the computer, it is very often necessary to select, or mark, a piece of an expression for further use. As an example, you might have a complex expression in parentheses, which you need to simplify prior to being able to continue your calculation. you will then want to start a Simplification Assistant, as outlined in [10], to be applied to just that particular expression in parentheses. To tell the assistant that it should handle the expression, you need to mark it in your working environment.

Of course, you could do this by standard GUI methods - traditional keyboard shortcuts, usually involving the Shift key, or mouse actions, typically Drag and Drop. But this procedure is not always what a blind pupil would desire: For first it is not always easy to navigate within a selected portion of text; in particular, you cannot use the cursor in such a case, since selection will be destroyed as soon as you move the cursor. Secondly, you want to have a possibility to mark particular expressions in a quick way, not involving a walk through the whole expression. To remain with the above example of a parenthesized expression to be selected, it would be desirable to have a function that, when pointing at an opening parenthesis, automatically selects everything between that parenthesis and the matching closing parenthesis.

The first of these two arguments, the problem that the cursor cannot be used when something is selected, would inspire the idea to create a concept of selection that replaces the one inherited from standard GUI, a type of selection that is independent from the cursor. Within such a setting, one could facilitate selection of a portion of text by just pointing at the start and at the end position of the text to be selected. The second argument, the desire for automatic selection tools based on the structure of an expression, would motivate routines such as:

- Selecting the current summand of a sum,
- Selecting the current factor of a product,
- Selecting the current side of an equation, etc.

4.2 Considerations for Navigation Support (Problem 3)

One of the simplest aspects of navigation support is the ability to collapse and expand an expression, similar to what is offered by modern programming environments: When confronted with a very long and complex formula, the user should be able to temporarily hide portions of it from the display. At a later time, he or she should be able to make these concealed pieces visible again. Such a technique poses several questions, e.g.:

– How should the user specify the portion to be concealed?
– By which means - braille characters etc. - should the user be informed about the presence of a concealed piece in a formula?
– By which means should expansion of a collapsed portion be commanded

When examining these questions, it becomes apparent that one needs to distinguish between representational and structural view of a formula: If a long and complex expression is presented, it might be desirable to specify a piece to be collapsed by just selecting its start and end position within the formula. However, such a technique, although fairly easy to implement, would not prove too efficient when the piece to collapse gets lengthy: What one might desire would be a condensed view of an expression, telling the user only coarse structural information about it. To take an example: When dealing with an intricate fraction, a condensed view might only convey the information that a numerator, a fraction line, and a denominator are there. Should the user need more information about, say, the numerator, a command needs to be implemented to expand that part of the expression, but, perhaps, not to full depth.

Another family of helpful functions might be what we call path support: Inside an expression, clicking at a spot should give detailed information on the path leading to that spot within the structural tree of the formula. This could be conveyed by a path string similar to a path name in a file system. As an additional support feature, clicking at a branch in such a path string could display the subexpression corresponding to that branch.

4.3 Considerations for Manipulation Support (Problem 4)

In order to aid blind students in formal manipulation, i.e., in doing standard mathematical exercises, elaborate, efficient, and flexible navigation support is essential. In addition, the main classes of mathematical tasks met in education need to be identified - this will be done through our analysis of school books announced in the previous section. Such standard tasks might be:

– Multiplication of sums in parentheses [10]
– Simplification of a sum [10]
– Solving a linear equation
– Working with fractions
– Doing basic calculations in arithmetics
– Multiplying and dividing polynomials
– Doing elementary vector and matrix operations
– Solving systems of linear equations

Through analyzing tasks of that kind, a set of standard support functions shall be isolated. The ideal situation would be to implement this set of standard functions, such that support for every standard task like the above may easily be given by combining several of those functions, either in terms of a wizard, or in terms of an environment that lets a student choose freely which support functions he or she would like to use to perform a particular task.

In what follows, we shall present several aspects of manipulation based on standard exercises taken from a school book. For every such aspect, the problems it poses for a blind pupil are considered, and ideas to give support in overcoming the problems are outlined.

Support for Side Calculations, and Calculator. In an exercise like 559,c in [11]:

$$7,1x + 4,8x - 0,5x =$$

you need to add the numbers $7,1$, $4,8$, and $-0,5$ in order to get the coefficient of the variable x. You should not be expected to do such an addition in your memory. Rather, you should be able to organize it in terms of a side calculation: The working environment should allow you to open a second instance of itself, with the sum $7,1+4,8-0,5$ as input. This input could be constructed by walking through the original exercise, marking each of the three numeric summands, and issuing a command that presents the three marked numbers as input values of a side calculation, whose type is Addition. From that point, the side calculation could be solved either by starting the arithmetic addition assistant described in [10], or, in case of a more advanced teaching setting where the use of a calculator is permitted, by invoking a calculator emulation built into the MAWEN.

In any case, side calculations should be easily identified for further reference, which could be done by assigning names to them, or by automatically numbering them.

Support for Parentheses; Passive Assistants. In an exercise like 587,a from [11]:

$$5a - 3b - (2a - 4b + 8) + (2b + 4)$$

the second parenthesized term is no problem, because the parentheses can simply be omitted. However, the first parenthesized term needs consideration, because resolving the parentheses involves swapping of sign in every summand. As always in such situations, it would be inconvenient for the pupil to be required to remember every summand in order to copy it into the interim result with swapped sign. A convenient way of support would be furnished as follows: first, an identical copy of the input string is made, connected to the input through an $=$ sign:

$$5a - 3b - (2a - 4b + 8) + (2b + 4) = 5a - 3b - (2a - 4b + 8) + (2b + 4)$$

Then, the pupil walks through the right side of the equation, deleting the opening parenthesis after the $-$ sign, manually inverting the sign of the two remaining summands, $-4b$ and $+8$, and deleting the closing parenthesis. In such a way, the parentheses are resolved conveniently. We would like to call such a kind of assistant a passive assistant, because no active software support is given - the pupil does everything that has to be done manually, but still without inconvenient load on his/her memory. The passive assistant could be made partially active if it marks every sub-term whose sign has been manually swapped by a checked sign, but such high-level support should be optional.

In many cases, parentheses are nested, just like in Exercise 589,a from [11]:

$$6s - [3s + 2t - (s + 2t)]$$

Here, one must first resolve the inner parentheses, such that the contents of the outer ones, which are here written as square brackets, can be simplified. After doing so, the whole expression must be simplified again.

This is a typical example for a situation where a sub-calculation should be initiated: First select the contents of the square brackets, in order to start a Simplification Assistant with the selected expression as input. Within this sub-calculation, a passive assistant to resolve the inner parentheses, as described above, may be used. Once the expression, which originally was the contents of the outer parentheses, has been successfully simplified, it is time to return from the sub-calculation into the main one, with the contents of the outer parentheses replaced by their simplified version. What remains to be done is a simplification of an expression without parentheses.

Doing Tests for an Algebraic Calculation. In elementary algebraic teaching, you are often required to test the correctness of a manipulation by assigning concrete numeric values to a variable. As an example, Exercise 566,a in [11] instructs you to simplify the expression,

$$6a + 4b - a + 5b - 3a$$

which results in $2a + 9b$, and to test the correctness of your calculation by assigning the value 2 to the variable ”a$and3to$”b.

For a blind pupil to carry out such a test, one problem will arise, namely, to remember the values, in our case 2 and 3, to be assigned to the variables. This could be supported by a kind of passive assistant: the pupil copies both expressions whose equality is to be tested, the original input and the result, into an edit buffer. Now, in order to compute the numerical value of one of these expressions resulting when the variables are assigned to numbers, the pupil will walk through the expression, replacing every occurrence of the variables by the numbers. this can be done by simple editing - it is a passive assistant, leaving the work to the pupil. However, in order to give support in remembering the number to be used instead of a variable, the assistant could prompt that number whenever the pupil points to the corresponding variable.

References

1. D. Archambault. UMCL Homepage. On-line resource: http:\\inova.snv.jussieu.fr\iGuma\umcl\, Apr 2006. (accessed 11 Apr 2006).
2. University of Tampere Finland Department of Computer Sciences, Tampere Unit for Computer-Human Interaction (TAUCHI). The MICOLE Project. On-line resource: http:\\micole.cs.uta.fi, Apr 2006. (accessed 11 Apr 2006).
3. H. Epheser, D. Progranicznа, K. Britz, and B. Betz-Weitzel. *Neufassung und Vervollstaendigung des Systems der internationalen Mathematikschrift fuer Blinde. Marburger Systematiken der Blindenschrift.* Number 6. Deutsche Blindenstudienanstalt, 1986.
4. Faculty of Mathematics and Graduate School of Mathematics, Kyushu University, Japan. Infty Project Homepage. On-line resource: http:\\www.inftyproject.org\, Apr 2006. (accessed 11 Apr 2006).
5. Henter Math LLC. Virtual Pencil Homepage. On-line resource: http:\\www.hentermath.com\, Apr 2006. (accessed 11 Apr 2006).
6. A. Karshmer. MathGenie Homepage. On-line resource: http:\\karshmer.lklnd.usf.edu\~mathgenie\, Apr 2006. (accessed 11 Apr 2006).
7. Lambda Consortium. Lamda Project Homepage. On-line resource: http:\\www.lambdaproject.org\, Apr 2006. (accessed 11 Apr 2006).
8. K. Miesenberger, M. Batusic, and M. Stoeger. LABRADOOR, a contribution to making mathematics accessible for the blind. In A. Edwards, N. Arato, and W. Zagler, editors, *Computers and Assistive Technology - 6th International Conference on Computers Helping People with Special Needs ICCHP 98.* Oldenbourg, Wien, Muenchen, 1998.
9. A. Nemeth. *The Nemeth Braille Code for Mathematics and Science Notation: 1972 Revision.* American Printing House For the Blind, New York, 1972.
10. B. Stoeger, K. Miesenberger, and M. Batusic. *Multimodal Interaction with Mathematical Formulae: Access, Communication and Support for Blind People in Doing Mathematics.* L. Lawrence Erlbaum Associates, Inc. (LEA), 2005.
11. W. M. Stutzka. *Lebendige Mathematik 3*, volume 1. OEBV/HPT Wien, 2001. ISBN 3-209-03756-6.
12. The Shodor Education Foundation Inc. BRL: Braille Through Remote Learning. On-line resource: http:\\www.brl.org\, Apr 2006. (accessed 11 Apr 2006).
13. Universitaet Karlsruhe, Studienzentrum fuer Sehgeschaedigte, Karlsruhe. ASCII-Mathematikschrift Karlsruhe/Dresden. On-line resource: http:\\web.inf.tu-dresden.de\~js28\material\ams\h-000001.htm, Apr 2006. (accessed 11 Apr 2006).
14. World Wide Web Consortium. MathML Homepage. On-line resource: http:\\www.w3.org\Math\, Apr 2006. (accessed 11 Apr 2006).