Mathematics and Accessibility: a Survey

Arthur Karshmer,[a] Gopal Gupta[b] and Enrico Pontelli[c*]

[a]University of San Francisco , USA, E-mail:akarshmer@usfca.edu, [b]University of Texas at Dallas, USA, E-mail:gupta@utdallas.edu, and [c]New Mexico State University, USA. E-mail: epontell@cs.nmsu.edu

Abstract In this chapter we review the state-of-the-art in non-visual accessibility of mathematics. Making mathematics accessible is a significant challenge, due to its 2-dimensional, spatial nature and the inherently linear nature of speech and Braille displays. The relative arrangement of various mathematical symbols in the 2-dimensional print space is extensively exploited by standard mathematical notations to succinctly convey information to the sighted reader (e.g., square roots, exponentiations, fractions). The spatial layout implicitly encodes semantic information which is essential to the understanding of mathematical constructs, however, it poses a hurdle in making mathematics accessible in the two most traditional media adopted by visually impaired students, i.e., Braille and aural rendering. In both approaches, the information implicit in the spatial arrangements has to be made explicit, making accessing and understanding of mathematics by the visually impaired cumbersome. In this chapter, we consider various approaches proposed as well as in use for making mathematics accessible. We survey approaches based on Braille codes (e.g., the Nemeth and Marburg codes) as well as those based on aural rendering of mathematical expressions (such as AsTeR and MathGenie). We cover existing systems that are in use for making mathematics accessible, along with current research in the area.

Keywords: Mathematics, Accessibility, Visual Disabilities

1. Introduction

The study of mathematics is crucial in the preparation of students to enter careers in science, technology, engineering and related disciplines such as the social and behavioral sciences. For many sighted students, math education poses a serious roadblock in entering technical disciplines, which has a serious impact on our economic competitiveness and science-related capabilities. For the visually impaired student, the roadblock is even higher, due to the additional difficulties they have to face in accessing mathematics.

Presentation of written information to blind individuals has traditionally been accomplished through the use of Braille. For presenting text, while Braille may not have been an ideal solution, it has certainly been a satisfactory one. Traditional Braille utilizes a raised character set composed of six dots per character, which limits the character set to 64 possible characters. Even for simple text, this does not represent an adequate alphabet. To solve the problem, most Braille notations use multi-character representation. For example, "A" (capital a) is represented in American standard Braille by a sequence of two characters: ",a", i.e., the letter a preceded by a comma..

In recent years, there have proponents of an eight-dot system, which could then allow an alphabet of 256 unique characters (Schweikjardt 1998; Gardner 2005). For a variety of reasons, that resemble the "QWERTY" vs. the Dvorak[1] keyboard debate, the eight-dot systems have not gained much popularity. While the text Braille issue is complex, it pales in comparison to the difficulties in the representation of math in Braille. While text is linear in nature, anything but the simplest math is not normally represented linearly. Note that Braille itself is also linear. Thus, the multitude of mathematical operators and special symbols has to be simulated by various sequences of Braille characters. Additionally, the spatially arranged structures have to be linearized so that they can be represented in Braille (e.g., the square root operation). All these requirements make math learning and teaching very complex.

In 1990, the United States passed landmark legislation, the Americans with Disabilities Act, to directly address a broad range of problems. Similar legislations have been passed by many other countries. These laws have significantly improved accessibility for people with disabilities, but they have had marginal success with math and the visually impaired. Other legislation such as The Education Act of 1973, The Telecommunications Act of 1996 and the aggressive implementation of Section 508 of The Education Act of 1973, have resulted in many positive changes. Unfortunately, the math education of the visually impaired has not seen any noticeable improvement.

In response, numerous projects have surfaced to bring closure to the problem. A simple Google search on the words "math blind," currently returns 1.4 million hits. The problem however is complex, and dual pronged: to make meaningful headway, the needs of both the students and their teachers have to be taken into account. Thus, any solution to the problem must make it easy for blind students to internalize ("visualize") mathematical expressions. Likewise, a good solution should not place undue burden on the teachers, in terms of preparing the material for the student, or having to learn substantial new material (e.g., learning a Braille Math notation).

There have been several interesting projects over the years that have addressed these issues, and they will be discussed later in this chapter. While a generic solution has yet to be developed, progress is being made. With current inter disciplinary research projects, the

[1] Dvorak International, P.O. Box 11985, Eugene, OR, 97440

underlying issues are being exposed by teaming mathematicians, cognitive psychologists, human computer interaction experts and the user community.

2. A Classification of Math Accessibility Approaches

The goal in making mathematics accessible to a blind student is to ensure that entire mathematical document is accessible. This means that not only individual mathematical expressions are accessible, but sequences of expressions that may arise, for example, in a proof, as well as text embedded between mathematical expressions are also accessible. The approaches to making Mathematics accessible can be classified into two types: static and dynamic.

- **Static Approaches:** the mathematical content is statically converted into a format that is reproducible using assistive devices (such as Braille refreshable displays and other tactile devices) or that can be printed on Braille paper. In these approaches, the document is mostly viewed as a *passive* entity (akin to a printed document presented to a sighted user), while the *active* component is represented by the user, who uses an assistive device (e.g.., a refreshable Braille display) to move around the document, reading parts, skipping other parts, backtracking through it, etc.
- **Dynamic Approaches:** the mathematical content is presented in a dynamic, interactive fashion. These approaches require a conversion process which allows the user to navigate the mathematical content in accordance with its mathematical structure. In this case, the document itself becomes an *active* component; by performing intelligent transformation of the document, its semantic structure is exposed and information overload on the user reduced.

Note that static approaches render mathematical expressions in Braille, while dynamic approaches render it using audio – alone or along with other traditional techniques, such as refreshable Braille. It should be noted that the two are complementary to each other. Experience indicates that just as the ability to read and write is important for sighted individuals (in addition to being able to hear things), the ability to write using Braille-based codes is likewise important for blind individuals as audio hearing is not enough, howsoever interactive it may be (National Library of Service 2000).

3. Static Approaches

3.1. Introduction

In the static approaches sophisticated, special Braille notations have been developed for mathematics. Virtually every national Braille notation also has at least one special version for math. These math codes attempt to present complex mathematical expressions in a way that blind individuals can follow them. These notations include the Nemeth Math code (Nemeth 1972) in use in US, Canada, New Zealand, Greece, and India, the Marburg code (Epheser, Pograniczna et al. 1992) in use in Germany and Austria, the French Math code (Commission Evolution du Braille Francais 2001), ItalBra (Biblioteca Italiana per Ciechi "Regina Margherita" O.N.L.U.S. 1998), the Italian Math code in use in Italy, and the British Math Braille code (Braille Authority of the United Kingdom 1987). Unfortunately, these notations introduce their own problems.

The Braille-based mathematical notations devised in various countries constitute a new language, and consequently must be learned by the students. Of course, this learning has to be supervised by teachers, who themselves need to be proficient in these notations. In the United States and many other countries, there is a paucity of math and science teachers (Lee 2005) and especially trained special education teachers capable of teaching both math and complex Braille notations (Gill 2006). With the broad acceptance of “mainstreaming,” and the resulting demise of special education centers for the visually impaired, the problem has become even more difficult. This is because a Math teacher at a school, community college or school may come across one or two blind students over a span of few years, and thus does not have a very strong incentive to learn the Math codes. The situation is somewhat alleviated through engagement of special education teachers who know the Braille math code and help teachers with blind students in their classes on a demand basis.

There are several national codes that have been designed for encoding Mathematics in 6-dots Braille. We give an overview of these codes below. We also give an overview of assistive technology tools that have been developed to make the task of learning and teaching Mathematics for blind students and their teachers respectively.

3.2. 6-dots Braille

The 6-dot Braille was invented by Louis Braille in the 1820s. It uses 6 dots arranged in 3 rows of 2 dots each which are raised depending on the letter or symbol to be represented. There are 64 possible letters and symbols that can be represented by the 6-dot system (no raised dots represents a blank character, thus there are 63 letters/symbols where at least 1 dot is raised). The invention of Braille allowed blind individuals to read and write for the first time at speeds equal to or surpassing those of the sighted individuals in writing print text (Braille 1829). What Braille did was to provide an alphabet to allow blind individuals to read and write. This leap opened door to other possibilities, such as Braille-based code for Mathematics, Science, Music, etc. Braille-based codes for Mathematics started appearing in the mid to late 1900s: For example, the Nemeth code, designed by blind Mathematician Abraham Nemeth, was published in 1951, while the Spanish Math Braille code was developed in 1987.

3.2.1. Overview of Braille Mathematical Formats

As mentioned earlier, there are several standardized Math codes developed by different countries (USA, UK, France, Italy, Spain, etc.). Most of these codes attempt to cover all of Mathematics that can be expressed in print Math. These codes "linearize" Mathematics. Note that any 2-dimensional mathematical expression can be encoded in a linear sentence through the use of parentheses. Thus, the fraction $\frac{x+1}{x-1}$ can be represented as (x+1/x-1). However, this process of linearization introduces too many parentheses, thus most codes will provide a wide variety of grouping symbols for indicating the beginning of a group as well as its end, e.g., in Nemeth code the above fraction is represented as ?x+1/x-1#, where "?" is the begin fraction indicator and "#" is the end fraction indicator. Using different grouping symbols for different categories of expressions is important for a linear notation that will be read left to right. If regular parenthesis were used instead of "?" and "#", then the blind reader would know that they are reading a fraction only upon encountering the symbol "/". Using special symbols to indicate the reader what he/should anticipate is an important design feature while developing Braille Math codes. An equally important feature is keeping the user aware of the context they are in at all times. This is because at any given time, the user is focused on one symbol (the one on which his/her finger is placed) and the part to the right of the finger is not known to him/her at all (in contrast to a sighted user who can infer lot of structural information in one glance), while the text to the left is in the reader's memory. Thus, in Nemeth code, for example, while writing expressions involving exponents, the level of the exponent has to be indicated explicitly at each time. For example, the expression $x^{(z+2)^c+3}+y$ is coded as {x^{{z+2}^{c}+3}+y in LaTeX (Lamport 1985), a popular package for Math print typesetting, since to a cited reader (or a computer) the braces indicate the scope of exponents. However, this does not work well in a Braille setting, as the reader will quickly forget the exponent level as he/she moves left to right in the formula, and considerable number of backtrackings of the finger will be needed to understand it. To make the context explicit, in Nemeth code this formula is coded as: x^z+2^^c^+3"+y. Note that the number of ^ indicates the exponent level of the expression that is to follow. The context awareness that is built into Braille Math codes makes them extremely difficult to parse (Karshmer, Gupta et al. 1998), however.

The various Math Braille codes devised differ in

(i) The mapping of the Braille alphabet to ASCII as different countries use different standards; for example, unlike the American standard, there is no direct encoding of the + symbol in British Math Braille code.

(ii) The sequence of Braille characters used to denote various mathematical operators and special symbols; e.g., the begin fraction and end fraction indicators.

To illustrate these differences, consider the expression

$$\frac{x+1}{x-1}$$

This expression has different encodings in different Braille national standards. The following (Archambault, Batusic et al. 2005) are the encodings in some popular formats

⠹⠭⠬⠂⠌⠭⠤⠂⠼ (Nemeth)

⠰⠭⠖⠡⠆⠲⠰⠭⠤⠡⠆ (French 1)

⠢⠭⠖⠡⠔⠲⠢⠭⠤⠡⠔ (French 2)

⠫ ⠸⠭⠖⠾⠁⠝⠏ ⠸⠭⠤⠼⠁⠹ (ItalBra)

⠆⠄⠭ ⠬⠼⠁⠈ ⠦ ⠆⠭ ⠤⠼⠁⠈ (Marburg)

⠣⠰⠭ ⠰⠖⠼⠁⠜⠌⠣⠰⠭ ⠰⠤⠼⠁⠜ (British)

Observe that French has two encodings - the first is the historical format, while the second makes use of the recently introduced new French Braille format (the format was introduced so as to simplify the language to make it computer-processable. Markers are used to denote numerators and denominators. In all formats except for Nemeth code and ItalBra, the numerator and denominator blocks are the same, thus a reader is aware of the fraction only when the fraction symbol is reached. Nemeth and ItalBra avoid the use of a fraction symbol by using different block markers. Not all Math codes follow the design features mentioned earlier to the same degree.

Some Braille Math codes also allow for spatially arranged structures, such as matrices, determinants, continued fractions, and grade school level arithmetic sum, multiplication and division problems. Two examples of spatial arrangements (polynomial addition and a 3x2 matrix) for Nemeth code are shown below. Note that the sequence of “3s” is used in Nemeth code to draw a line needed in representing arithmetic sums, multiplication, long division, etc.

	ACSII Nemeth	Nemeth Braille code
Polynomial	3x^2"+2x-25	⠒⠭⠘⠆⠐⠬⠆⠭⠤⠆⠢
Addition	+6x^2"-5x	⠬⠖⠭⠘⠆⠐⠤⠢⠭ ⠒
	3333333333333	⠒⠒⠒⠒⠒⠒⠒⠒⠒⠒⠒⠒⠒
3x2	,(,a 0,)	⠠⠷⠠⠁ ⠴⠠⠾
Matrix	,(0 ?1/,asin^2 .b#,)	⠠⠷⠴ ⠹⠂⠌⠠⠁⠎⠊⠝⠘⠆ ⠨⠃⠼⠠⠾
	,(0 ?1/,c#+?cos .b/2#,)	⠠⠷⠴ ⠹⠂⠌⠠⠉⠼⠬⠹⠉⠕⠎ ⠨⠃⠌⠆⠼⠠⠾

3.2.2. Translation and Back-translation of Math codes

Given the various Math codes, one obvious problem faced is to convert print Math into a specific Math code (translation). Thus, tools need to be developed that achieve precisely this. Any embedded text should also be translated (to text Braille). Typically, print Math documents are prepared in LaTeX. Thus, the conversion task involves translating Latex document to a Braille document where Latex Math expressions are coded in Math Braille code while the embedded text is coded in text Braille. Translators have been built for translating LaTeX to various national codes. The most mature of these is the Scientific Notebook system that converts LaTeX to Nemeth code (Duxbury Systems 2000). Other systems include those for converting to Marburg, to the French code, and to ItalBra.

Once print Math & text has been translated into Braille for a blind student to "read," another problem arises, namely, how does the sighted teacher read the answers prepared by a blind student, say, for a homework or an exam. The blind student will typically write the answers in Braille. To facilitate this communication from student to teacher, we need tools that will *backtranslate* the Math code and Braille text to LaTeX, so that it can be read and graded by a sighted instructor. The task of building a backtranslator is significantly harder than the task of building a translator from LaTeX to Braille math, because typically the Braille Math codes were designed to be backtranslated by human translators, and thus have a large number of *context sensitive* features that make computer processing and parsing extremely difficult. As a result the problem of back-translation was widely considered unsolvable (Scadden 1996), until Gupta, Karshmer and Guo proposed advanced techniques based on *programming language semantics* and *logic programming* for achieving this backtranslation (Karshmer, Gupta et al. 1998; Annamalai, Gopal et al. 2003). In fact, the French Braille math code was revised and changed to make it more computer processable and parsable (Moço and Archambault 2003). A number of projects have attempted to build translators and backtranslators for various Math code

- **The Labradoor project** (Batusic, Miesenberger et al. 1998) was started for converting LaTeX to Marburg, and subsequently backtranslating Marburg to LaTeX (Batusic, Miesenberger et al. 2003).
- **MAVIS** (Karshmer, Gupta et al. 1998) was the pioneering project which showed how to solve the backtranslation problem using language semantics and logic programming, and developed the first Nemeth Braille code to LaTeX backtranslator.
- The **Insight** project (Karshmer, Gupta et al. 1998; Annamalai, Gopal et al. 2003; Gopal, Wang et al. 2007) further improved upon the MAVIS project to develop a complete system for backtranslating Math documents in Nemeth code with embedded text (in Grade II Braille) to LaTeX. The system takes, as its input, a scan of a Braille sheet (in JPG format), performs image processing to recognize the Braille dots and produces the corresponding ASCII Braille file. The Nemeth code and Level II Braille are automatically identified and separated and then each translated separately and merged to produce a single LaTeX file that a sighted person can then view.
- **Multi-Language Mathematical Braille Translator** (Moço and Archambault 2003), uses the approach developed in the UMA (Universal Mathematics Access) project (Karshmer, Gupta et al. 2004) to provide multi-lingual translation between the two French notations, the Marburg notation, and the Nemeth code. The approach is based on developing a common intermediate format the various notations can be translated to, back and forth.
- **BraMaNet** (Schwebel 2004) is a system that uses an XSL Style Sheet to translate MathML into French Math Braille code. It contains a user-friendly VB interface that can be used, in conjunction with MathType, to translate Word Documents into Braille for printing.
- **Math2Braille** (Crombie, Lenoir et al. 2004) is an open-source module to convert MathML 2.0 to the Braille standard used in the Netherlands.

3.2.3. Universal Libraries

The previously described approaches highlight the inherent difficulty arising from the great variety existing in the digital formats (e.g., MathML, OpenMath), typographical formats (e.g., LaTeX) and Braille formats (e.g., Nemeth, Marburg) adopted to describe mathematical content. Furthermore, in many settings (e.g., educational applications) it is interesting to allow translation between arbitrary pairs of formats - e.g.,

- Translation from MathML to Nemeth Braille is required for students accessing mathematical content deployed on the web
- Translation from LaTeX or MathML to Nemeth Braille is required to allow an instructor to distribute homework and notes to visually impaired students
- Translation from Nemeth Braille to LaTeX is required to allow a student, who is typesetting his/her solutions using a Braille Typewriter or a Braille embosser, to submit the material to the instructor.
- Translation of a Mathematical document written in one Braille-based Math code (say Marburg) to another Braille-based Math code (say Nemeth), to facilitate communication between blind scholars, mathematicians and engineers.

The research conducted by the iGroup UMA[2] directly addressed this problem. This work hinges on designing a common intermediate format to which various notations (for sighted as well as non-sighted) can be translated to, back and forth, as the basis of translating one notation to another. As part of this project, Palmer, Pontelli et all (Palmer, Pontelli et al. 2003) proposed OpenMath as a common intermediate format, and to use it as a basis to develop two-directional translation tools between OpenMath and the other relevant formats (e.g., Nemeth Braille, Marburg, LaTeX). The work of iGroup UMA successively evolved in a full-blown and open source *Universal Math Conversion library* (Archambault, Fitzpatrick et al. 2004; Archambault, Batusic et al. 2005). The emphasis in this project is have an open library, easily expandable, that supports multi-modal presentation of mathematical content and which can be transparently used by non-expert users. The library provides a single API for developers of applications necessitating conversions between different mathematical formats. The library relies on a central format, and the format chosen is MathML Presentation. For each additional format, the library provides two modules (the *input* module and the *output* module) aimed at converting to and from MathML. The library maintains, in particular, internal Braille tables describing each National Braille format. The library is highly portable and accessible by both Windows and Linux applications.

3.3. Mathematics-specific Braille Extensions

The use of 6-dot Braille for the encoding of mathematical content has been widely criticized. The 6-dot format can only produce 63 different Braille cells (observe that in 6-dot Braille an unused cell or blank cell is implicitly considered as a space). As a consequence multiple sequences of Braille cells have to be used to encode distinct symbols, and many Braille cells have different meanings depending on the context they appear in. This type of designs have an unfortunate consequence – formats like Nemeth Braille become *context*

[2] http://karshmer.lklnd.usf.edu/~igroupuma/index.html

sensitive languages (Hopcroft, Motwani et al. 2000), which are inherently hard and expensive to parse and translate.

For this reason, there have been many attempts made to expand Braille codes to 8-dot formats. 8-dot allows a larger number of Braille cells (256), simplifying the encoding of a larger set of symbols and create a better consistency with the 8-bit standard ASCII character set used on most computers. A number of official and semi-official general 8-dot Braille codes have been developed, mostly in Europe, but relatively little literature has been reproduced in any 8-dot Braille code. In the context of representation of mathematical content, two relevant proposals targeting 8-dot representation of mathematical content are DotsPlus (Gardner 2003) and LAMBDA (Edwards, McCartney et al. 2006).

DotsPlus Braille gracefully extends 6-dot grade 1 standard Braille, with the exception that most double cell characters of the 6-dot font are single cells in the 8-dot font. Capital letters, for example, are encoded with an extra dot (dot-7 position) on the left side of the row below the bottom of the standard 6-dot lower case letter cell. The novelties of DotsPlus include

- Most punctuation marks are not Braille but small graphic symbols; similarly, most of the characters from complex literature (e.g., mathematical symbols) are encoded as graphic symbols shaped similarly to the corresponding print symbols.
- Numbers are encoded in single cell; the digits from 1 to 9 are encoded as in the literary Braille number mode with an additional dot-6 (the digit 0 has a distinct encoding to avoid conflict with the letter ‘w’).

The outcome is a notation that is easier to learn and remember for the representation of mathematical content. Unusual symbols are encoded in a shape analogous to the print format, allowing visually impaired individuals to follow the same learning/remembering process as a sighted reader, and avoid having to memorize complex sequences of Braille cells. Furthermore, DotsPlus Braille reduces the context dependence of individual symbols in a Braille line, leading to the ability to print math in standard print format (including positional placements of symbols). Figure 1, drawn from (Gardner 2002), shows the print format and the DotsPlus format of a simple equation. The success of DotsPlus is also related to the ability to

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

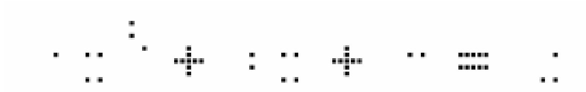


Figure 1: A simple equation in print format and DotsPlus format

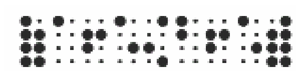
use the Tiger Tactile Graphics and Braille Embosser (Gardner, Ungier et al. 2006), which includes printer drivers for DotsPlus fonts.

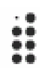
The LAMBDA project introduced a linear math notation, the LAMBDA code (Edwards, McCartney et al. 2006; Fogarolo 2006; Schweikhardt, Bernareggi et al. 2006), largely inspired by the way mathematical content is encoded in MathML. Thus, the LAMBDA code plays the role of a markup language – i.e., it provides markers to denote special types of expressions and representation of common symbols – expressed in an 8-dot Braille notation, to provide linear encoding of mathematical content. The LAMBDA code is meant to be an internal source code for the representation of mathematics, and tools have been investigated to translate LAMBDA code to national Braille formats (e.g., Italian Braille) for output purposes.

For example, the representation of a fraction requires three markers: one denoting the beginning of the fraction, one representing the end of the numerator, and one representing the end of the fraction. For example, the fraction

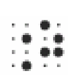
$$\frac{a+1}{b+1}$$

would be represented as <start fraction> a+1 <fraction symbol> b+1 <end fraction>. The three markers have different 8-dot Braille representations depending on the specific country; the above fraction, in UK 8-dot LAMBDA code, would appear as (Fogarolo, Bernareggi et al. 2005):

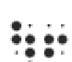


where  represents the <start fraction>,  represents the <end fraction>, and  represents the fraction symbol.

The use of 8-dot Braille code, with its 256 possible combinations of dots, implies that many markers and symbols require multiple Braille cells. The format structure adopted in the LAMBDA code tries to appeal to logical constructions to facilitate the interpretation and recollection; in particular, many symbols are constructed using prefixes, which identify the class the symbol belongs to. For example, Greek symbols are constructed by a fixed prefix followed by the Braille code for the corresponding Latin letter, e.g., the symbol ζ is represented as:



As another example, the symbol $\cup$ (set union) is encoded as



where the first cell is a prefix denoting set operations, and the second cell is the traditional encoding of '+'.

The LAMBDA code is supported by a sophisticated LAMBDA editor, which recognizes and handles the hierarchical structure of mathematical expression, automatically manages the different blocks composing it, and allowing different forms of visualization (e.g., it allows to hid the content of the blocks).

3.4. Other Tactile Approaches

- ViewPlus™ have developed a variety of tools to enhance the mathematical experience of visually impaired individuals. One of the notable products is the *Accessible Graphing Calculator* (AGC) (Walsh, Lundquist et al. 2001; Gardner, Ungier et al. 2006), which offers the traditional features of a graphing calculator but with the ability to produce both aural presentation (as a varying frequency sound) and tactile presentation (using Tiger embosser) of two dimensional graphs.
- A recent improved method for determining significant boundaries in images encoded in Scalable Vector Graphics (SVG) has been investigated (Krufka and Barner 2005) with the goal of producing more effective outputs on the Tiger embosser; this can be effectively used for presentation of mathematical content, thanks to the recent studies of encoding of mathematical formulae using SVG (Stanley, Bledsoe et al. 2004).

4. Dynamic Approaches

4.1. Foundations of Math Presentation

4.1.1. Cognitive Foundations

Relatively little work has been done in investigating the cognitive aspects of human interaction with mathematical content. The most relevant effort in this areas has been reported in (Gillan, Barraza et al. 2004). This work studies the perceptual and cognitive processes used by sighted individuals during equations reading; the ultimate goal is to understand these processes to the extent of being able to develop aural presentation mechanisms that provide visually impaired individuals with equivalent process capabilities. The investigation in (Gillan, Barraza et al. 2004) was conducted using "think aloud" protocols and eye-tracking devices, and it involved separate experiments aimed at measuring

1. The capability of recalling equations – the experiment consisted of exposure to different equations for varying period of times, followed by presentation of "distracting" screens.

2. The impact of knowledge of the structure of the equation – the experiment exposed the subjects to a preview of the structure of the equation, and measured whether this affected the time to solve the equation.
3. The extent to which subjects use a “chunking” approach in reading equation, i.e., they decompose the equation into chunks, defined by expressions within parentheses, solve such sub-expressions and store their outcome in memory.

The outcome of this study can be summarized as follows:

- the reading process is mostly a left-to-right process, moving one element at a time (similarly to the way standard text is read)
- an initial scan is typically performed to acquire the structure of the equation before proceeding to its detailed understanding
- readers backtrack very frequently when understanding/solving an equation
- readers tend to process operators and numbers more deeply than parentheses
- readers chunk together parts of an equation (especially the content of a sub-expression within parentheses) and process the chunk modularly; as a consequence of chunking, the readers tend to solve the equation hierarchically.

4.1.2. Prosody and Speaking of Mathematics

Significant research efforts have been invested in understanding the effectiveness of different approaches to the actual "speaking" of mathematics.

The majority of the approaches to aural presentation of mathematical content rely on two key schemes for denoting the structure within a mathematical expression: lexical and prosody cues.

Use of *lexical indicators*, which explicitly denote structural information through additional spoken components. For example, an expression of the type $\sqrt{x+1}$ would require the use of lexical indicators of the type "begin square root" and "end square root", leading to a possible utterance of the form "begin square root x plus one end square root". Seminal work in the investigation of lexical indicators have been conducted by Chang in (Chang 1983). This approach has been widely adopted in many of the tools for aural presentation of mathematics, and it has been recognized as particularly effective when dealing with the major structure of the expression. On the other hand, various studies (e.g., (Baddeley 1992; Stevens, Edwards et al. 1997)) have highlighted negative aspects of lexical indicators, mostly associated to the overload imposed on the reader's working memory.

Use of *Prosodic indicators*, i.e., modifications of features of the spoken output to capture changes in the presented structure. These may include pauses, modifications of pitch and tempo, rhythm and tone. Seminal work on the use of prosodic cues in the presentation of algebraic content has appeared in (O'Malley, Kloker et al. 1973) - which discovered the correlation between pauses and syntactic boundaries at operators, fractions, parentheses, etc. This investigation was refined to consider pitch, duration, and amplitude by Streeter (Streeter 1978). A richer set of rules for the use of prosody cues in algebraic presentations has been presented in (Stevens 1996). The various studies have also drawn a distinction between speech prosody cues and non-speech ones. The studies of (Blattner, Surnikawa et al. 1989; Brewster, Wright et al. 1994; Stevens 1996) suggest the use of *algebraic earcons* which associate non-speech sounds to different constructs - e.g., a complex fraction is associated to two long notes with constant pitch separated by two silent beats (Stevens, Edwards et al. 1997).

The more recent work of Fitzpatrick (Fitzpatrick 2002; Fitzpatrick 2006) argues on the effectiveness of using only speech prosody; the proposal draws a parallel between the structure of mathematical expressions and the composition in English sentences, and using analogous use of pausing and speaking rate to aurally capture the nesting structure of the expression. Fitzpatrick's effort aims at the design of standardized prosodic effects, instead of ad hoc approaches used by individual human readers. Fitzpatrick also advocates the combined use of prosody and lexical clues to better deal with complex expressions (Fitzpatrick 2006), and to overcome limitations of existing speech synthesizers.

Another interesting overview of some of the issues connected to speaking mathematics has been presented in (Fateman 2006), with considerations relative to the transition between speech synthesis and speech recognition of mathematics.

4.2. Presentation and Navigation Tools

4.2.1. AsTeR

Audio System for Technical Readings (AsTeR) (Raman 1994) is a system to reformat electronic documents, typeset in LaTeX, to produce *audio documents.*

The first step in the AsTeR design originates from the development of a document representation model, aimed at making explicit the logical structure of the document; the model relies on attributed trees. Mathematical content is encoded in a *quasi-prefix* form, where the prefix form of the expression is enriched by *visual* attributes (e.g., left superscript, left subscript, accent, underbar); the objective is to delay the assignment of a semantics to the mathematical content, by preserving those visual components that make possible its (possibly ambiguous) presentation. AsTeR contains a recursive descent parser to extract quasi-prefix forms from the mathematical content of a LaTeX document, enriched with a complete precedence table of mathematical operators and heuristic rules to handle ambiguous notations (e.g., to ensure that an expression like $\sin 2n\pi$ is correctly interpreted as $\sin(2n\pi)$ instead of $\sin(2)\cdot n\cdot\pi$, as the strong precedence of function application would suggest).

AsTeR introduces a rule-based language, called *Audio Formatting Language (AFL)*, used to map the internal document representation to audio. Rules in AFL perform transformations of the *audio state*, which is composed of a speech state, a sound state, and a pronunciation state. The language allows blocks, multi-threads, and synchronization between modifications to different components of the state. Statements from AFL are grouped into *rendering rules* for each object of the document tree, and group of rules can be themselves grouped into a *rendering style* (akin to the notion of style-sheet). Multi-modality is achieved by activating and deactivating different rendering styles.

The rendering style provided for mathematical content is built on the principle of minimizing verboseness, mostly through the use of fleeting and persistent cues, and modifications of intonation and voice inflection. In particular:

- Nesting in the expression is reflected by a change in voice that feels like falling off into the distance, along with pauses around the subexpression (whose length is dependent on the complexity of the subexpression)
- Higher and lower pitch voices are used to reflect superscripts and subscripts
- In absence of additional information, the visual attributes are presented in the order subscript, superscript, underbar, accent, left-subscript, left-superscript

Parenthesized expressions are conveyed by combining the change in tone with a persistent sound cue.

AsTeR also relies on the principles (described in Section 4.1.1) of allowing readers to acquire the top-level structure of a mathematical formula before accessing the subexpressions ("chunking"). This process is realized in AsTeR via *variable substitution*; this is automatically applied when the complexity of the sub-expressions is sufficiently large. For example, given an expression $\frac{e_1}{e_2}$ where e_1 and e_2 are complex expression, this would be rendered as "fraction x over y, where x is ... and y is ...".

AsTeR provides active browsing capabilities. Mathematical content can be navigated as a tree structure, with the ability of moving across siblings, as well as marking nodes and returning to marked nodes. Since the mathematical content is enriched by the visual attributes, the navigation includes specific commands for accessing such attributes. Upon entering a node, an immediate summarization (which includes a summary of the content - typically the main operator) and a contextual description (which recalls the siblings) is presented.

4.2.2. MathGenie

Introduction

The MathGenie is a comprehensive equation reading program that presents the visually impaired student with verbal renderings of the equation under study along with refreshable Nemeth Braille code output (Stanley and Karshmer 2006). By design, the system offers a number of different techniques for browsing equations, each of which present the structure as well as the content of the mathematics. The system was designed to run on virtually any computer running Windows 2000 or later without the aid of any other software package.

For the teacher, no knowledge of Braille is required to generate materials for his/her visually impaired students. Through the use of any equation editor that generates MathML, the teacher can prepare materials for both sighted and visually impaired students. The MathML output of the equation editor is the input to the MathGenie equation browser. Figure 2 shows a snapshot of the MathGenie interface.

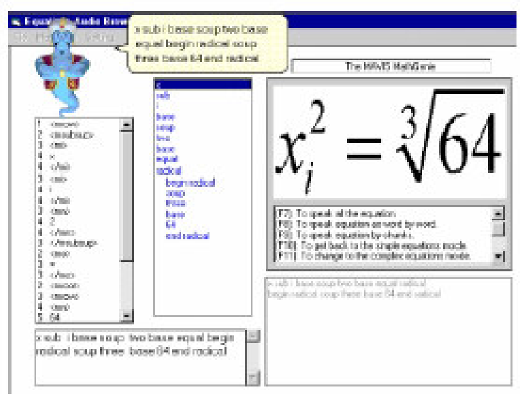


Figure 2: snapshot of the MathGenie user interface

Design Principles and Navigation Strategies

The design of the MathGenie involved research in domains, beyond the obvious computing and HCI issues. For the first time, cognitive psychological research was employed to understand the key issues in reading and understanding mathematical equations (Karshmer and Gillan 2003; Gillan, Barraza et al. 2004). The results of the studies indicated several important issues associated with reading equations to a visually impaired student. Among the findings were 1) casual reading of equations is highly prone to error on the part of the reader and therefore the listener, 2) structural components such as fractions, radicals, summations, parenthetical expressions, integrations, etc are critical components and should be foci in equation reading, 3) the preliminary “glimpse” of an equation seems to offer little in the process of equation cognition and 4) the student must have the ability to navigate equations in a variety of ways. These results became the foundation of the design of the MathGenie.

The MathGenie relies on the use of lexical clues to represent the structure of the mathematical content. The browsing process provides the following key features (Karshmer, Gupta et al. 2002):

- The default reading proceeds left to right along the expression; the access to the components of the expression can be realized either at the level of symbols or at the level of words;
- The expression can be presented in an abstract way, to highlight the hierarchical structure and abstract away small sub-expressions. For example, an expression of the type $a_i = \int_{\sqrt{a-b}}^{\sqrt{a+b}} x^2 dx$ could be abstractly presented as
    - a subscript i equals to
    - limit integral with
        - lower limit square root of something, and
        - upper limit square root of something
    - of something dx

  The depth of navigation to provide an abstract presentation is dependent on the specific symbols used in the equation.
- Users are allowed to evaluate sub-expressions and replace them with the corresponding value;
- The parse tree of the expression is visible, can be traversed in various directions, including the ability to repeatedly navigate the same sub-expressions.

Another important feature of the navigation process supported by the MathGenie is the ability to *voicemark* expressions. A voicemark, similar to a voice anchor (Reddy, Annamalai, et al 2004), is an aural bookmarking of sub-expressions, where the user can vocally assign a bookmark to any sub-expression during the navigation process. Bookmarking sub-expressions can be useful in understanding and internalizing expressions, allowing a step-by-step navigation, and allowing users to jump at will between voicemarked expressions – effectively allowing him/her to implement his/her own navigation strategy and customize the abstraction of expressions. Work is in progress to introduce voicemarking in the MathGenie infrastructure.

MathGenie's Architecture

Figure 3 shows the overall architecture of the MathGenie (Karshmer, Bledsoe et al. 2004). Input is provided in the form of (Presentation) MathML, and it offers synchronized display and aural presentation; the MathGenie includes the possibility of providing speech input to control navigation, and the ability to control a refreshable Braille display.

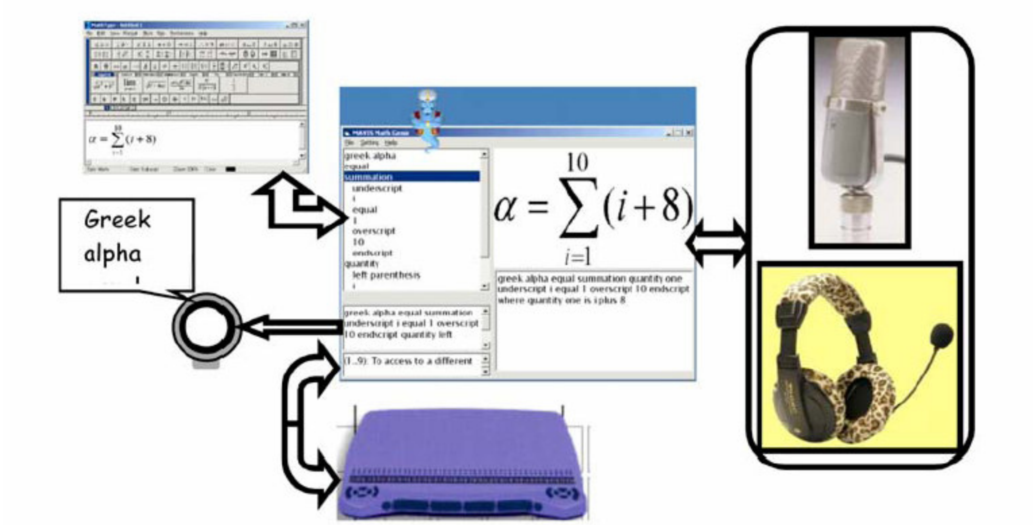


Figure 3: External architecture of the MathGenie

To increase the usability of the MathGenie, the browser uses a table-driven speech technique. This means that before any component is spoken, it is first found in the table for the local national language, its correct pronunciation is then sent to the speech engine. This again permits the use of inexpensive hardware and software tools, since the least expensive speech engine can be employed and no special voice engines are required. To date the MathGenie supports English, French and Spanish. The system includes editing tools to allow local language support to be added.

The MathGenie offers synchronized aural and graphical presentation of the equation components being accessed. The graphical rendering is aimed at enhance accessibility for individuals with low vision (by providing magnification of the equation been presented) and dyslexia (by providing color-contrasted highlighting). The visual rendering is achieved by mapping MathML to Scalable Vector Graphics (SVG).

Accessibility is further enhanced by connection to an on-line dictionary of mathematical terms, accessible through a simple keyboard shortcut during navigation of a mathematical expression.

4.2.3. Browsing Mathematics via VoiceXML

Another approach to interactively listening to and navigating Mathematics has been proposed by Gupta et al. (Reddy, Gupta et al. 2005). This approach is based on translating an expression coded in MathML to VoiceXML (Oshry, Auburn et al. 2006). They extend

VoiceXML (Reddy, Annamalai et al. 2004) to make it dynamically navigable via audio. A user can give navigational commands through voice, thus no keyboard is involved in the interaction.

VoiceXML is an XML based markup language designed for creating audio/voice based documents. Just as a web browser renders HTML documents visually, a VoiceXML interpreter renders VoiceXML documents aurally. The VoiceXML technology has been widely adopted in the telecom industry for automated handling of calls.

In Gupta and Reddy's approach, mathematical expressions coded in MathML are mechanically translated to VoiceXML via XSLT, a transformation language for XML. The VoiceXML document can then be heard with a voice browser by a visually impaired individual. However, merely hearing the VoiceXML document is not enough, since VoiceXML provides very little navigational control to a user while listening to a document. In fact, navigational control in present versions of VoiceXML is completely controlled by the author of the VoiceXML page (Reddy, Annamalai et al. 2004). To make navigation more interactive and under listener's control, they further enhance the VoiceXML document with *voice anchors*. VoiceXML documents enhanced with voice anchors allow listeners to place speech labels on sub-expressions of a mathematical expression. These speech labels can be used by listeners to move around the various sub-expressions via simple voice utterances and voice commands. Thus, the mathematical formula can be browsed much more effectively. Note that in Reddy et al.'s approach, in contrast to other browsers for Math formulas (such as MathGenie), the listener interacts with the documents and aurally browses math expressions through voice input rather than through keyboard input.

4.2.4. MathPlayer

MathPlayer (Soiffer 2005) has been proposed by Design Science Inc. MathPlayer originates as a browser plug-in for the visualization of MathML content; in its successive versions, MathPlayer includes speech rules based on lexical cues (making use of explicit start/stop markers of complex operators), as well as interfaces with screen readers (e.g., Jaws). In a style similar to MathGenie, MathPlayer allows the user to navigate the more complex mathematical expressions. MathPlayer supports two types of navigation:

- Text-based navigation: it allows the user to move linearly through the textual representation of the expression, by navigating words that are spoken.
- Tree-based navigation: it allows the navigation of the tree structure of the expression, in a fashion analogous to MathGenie.

A rule library is provided to minimize the number of words used to produce an unambiguous presentation of an expression; e.g., an operator applied to *single token* operands is spoken without the need of markers denoting the start and end of the expression - an expression like a/b is spoken simply as "a over b" instead of "begin fraction a fraction symbol b end fraction".

The navigation is synchronized with highlighting of the parts of the expression being accessed - a feature that is important for students with certain types of learning disabilities.

4.2.5. Other Systems

The AudioMath project (Ferreira and Freitas 2004) proposes a tool to provide aural presentation of mathematical content encoded in MathML Presentation. The AudioMath prototype relies on a database of rules for conversion of MathML to text (to be spoken) and the use of prosodic marks to provide pauses and modulations required to reduce ambiguity.

REMathEx (Gaura 2002) is another sophisticated math reader based on MathML Presentation. The program views expressions according to their tree structure and the user can navigate the tree structure. The navigation process relies on the alternation of different types of activities

- *Global expression preview:* this activity leads to the complete aural presentation of an expression up to a user-determined depth of its expression tree;
- *Selection of current node:* the user can position himself/herself on any node of the frontier of the expression tree visited during an expression preview; this will cause the presentation of the sub-expression rooted at such node;
- *Local expression preview:* this is analogous to the Global expression preview, except that it is applied to the sub-expression represented by the node the user is currently positioned at.

The aural presentation presents an expression up to a given depth; the elements below the set depth limit are presented using a sub-expression substitution mechanism (e.g., if the sub-expression below the depth limit is a/b then the system will replace it with the word *"fraction"*). The substitution is avoided if the "complexity" of the underlying expression is very small. Time delays are also used in the aural presentation to enhance the separation between components of the expression.

MathTalk (McClellan 2007) is another tool that employs voice to facilitate learning of Mathematics. It uses sophisticated voice recognition software to recognize mathematical expressions spoken by a blind user. In conjunction with Scientific Notebook, these expressions can be recorded in Latex, and then converted to Nemeth code.

5. Beyond Reading

5.1. Creating and Editing Accessible Mathematics

5.1.1. The Infty Project

The Infty project is mainly aimed at the enhancing the accessibility of mathematical content existing in printed form. The Infty system is composed of different tools, interacting via exchange of content in XML format. The key components are the following:

- InftyReader (Fukuda, Ohtake et al. 2000): this tool makes use of advanced OCR technology to recognize the structure of mathematical formulae from printed text; the recognition process proceeds by first creating a network linking the symbols in the expression by labelled edges (where each label denotes the type of relation between the symbols, e.g., subscript, superscript, etc.), and then using an algorithm to detect the spanning tree of the network representing the structure of the expression.
- InftyEditor (Suzuki, Kanahori et al. 2004): this tool is a typesetting tool for scientific documents, typically used to edit the output of InftyReader - indeed, the tool maintains in a window the image of the original printed document for comparison purposes. Mathematical content can be edited by reasoning about it in terms of its LaTeX representation. The editor includes the facility to produce an aural presentation of the expression (based on simple insertion of markers to unambiguously represent the structure of the expression).
- Infty coverters (Suzuki, Kanahori et al. 2004): the output of InftyEditor is stored in an XML format (called IML). The Infty project has led to the development of converters from IML to various output formats, such as LaTeX, MathML, Unified Braille Codes and Japanese Braille codes.
- ChattyInfty (Suzuki 2005; Komada, Yamaguchi et al. 2006): this tool is an advanced component for aural presentation, along the lines of the MathGenie System (Stanley and Karshmer 2006).

5.1.2. WinTriangle

WinTriangle (Gardner 2005) is a RTF scientific word processor, usable by both sighed and visually impaired individuals, capable of both displaying and voicing text and symbols, and providing access to a collection of markup symbols (Triangle fonts) that can be used to linearly encode virtually any scientific expression; the notation makes use of markers for the non-linear components of the expression - e.g.,

$\frac{a^2}{b}$ is encoded as the sequence of symbols [𝔽 a ↗ (2) ⟨/⟩ b 𝔽]

WinTriangle allows the editing and aural presentation of documents. In addition, the WinTriangle project has been enriched with an additional component, called LateX2Tri (Thompson 2005), which converts LaTeX to RTF with Triangle fonts.

5.1.3. Other Proposals

- REMathEx (Gaura 2002) is a system developed to provide the ability to read and edit mathematical expressions, encoded using MathML Presentation format. Expressions are viewed according to their parse-tree structure. The user can access the expression in its integral format or in *preview* mode, where the tree is navigated only up to a certain depth. The editing capabilities are limited to the substitution of the current subexpression - i.e., the node of the parse tree the user is currently positioned at - with a new expression. The editing is guided by an interactive dialogue, which provides aural menus for the choice of operators to insert in the expression.
- the BlindMath project (Pepino, Freda et al. 2005) integrates Scientific Work Place[3] and Jaws[4] to provide accessibility of document preparation using LaTeX to express mathematical content.

5.2. Working with Mathematics

The use of mathematical assistive technologies in the educational context raises the problem of moving from simple access to mathematical content to the actual *manipulation* of mathematics - i.e., doing mathematics, carrying out calculations and solving exercises. This prompted researchers to explore the development of assistive environments for the manipulation of mathematics. The most advanced proposal in this area is the work of Stöger et al. (Stöger, Miesenberger et al. 2004; Stöger, Batušić et al. 2006). Manipulation of mathematics involves tasks well beyond the simple understanding of a mathematical formula; it requires ability to segment the formula, copy and transforms part of it, and to maintain referential access to distinct parts of a formula. The Mathematical Working Environment (MaWEn) is designed to enhance formula navigation with the following features

- Ability to identify sub-expressions according to the semantic structure of the mathematical content - e.g., given the beginning of a parenthesized expression, automatically detect the complete sub-expression enclosed by parenthesis

---

[3] ™ http://www.mackichan.com/
[4] ™ http://www.freedomscientific.com/fs_products/software_jaws.asp

- Ability to mark&recall arbitrary sub-formulae as well as to conceal/reveal their presence on request
- Ability to replace sub-formulae with new expressions (e.g., simplify a sub-expression to its value)
- Support for "scratch" paper, where the user can temporarily copy parts of expressions to be manipulated, and ability to call a calculator
- Provide libraries to apply global transformations to complete expressions - e.g., allow to change a sign globally in a parenthesize expression instead of manually copying and negate each individual term
- Provide intelligent replacement strategies; for example, when evaluating an equation, each unknown has to be consistently substituted with the corresponding value.

6. Open Problems and Perspectives for the Future

6.1. Localization Problems

The previous efforts on translation between mathematical formats for mathematics have already addressed the presence of significant differences in notations across national boundaries.

The differences go beyond the specific notations – as distinct approaches, traditions, and methodologies affect the way mathematics is seen and understood. This implies that approaches to presentation of mathematics should include localization components and customize the delivery to the specific national standards.

Relatively limited work has been conducted to enable presentation methodologies to apply localization to its output. The Universal Math Conversion library (Archambault, Fitzpatrick et al. 2004) has been developed with the specific intent of facilitating the inter-operation between Braille mathematical documents expressed using different national math Braille notations – including the development of a canonical subset of MathML specifically designed to facilitate the inter-conversion process (Archambault and Moço 2006). The LAMBDA project (Edwards, McCartney et al. 2006) lays the foundations of the design of its 8-dot code on understanding the peculiarities of the different national codes used in different European countries. LAMBDA defines an open-separator-close tag structure to linearly represent mathematical structures. In order to be actually usable, the tag structures have to be concretized according to the national conventions (this was done for each country participating in the LAMBDA consortium). The instantiation process requires the following components:

- The specification of all dot configurations for the symbols which are represented on one Braille cell (e.g. lower-/upper-case Latin letters etc.).
- The definition of the notations to be described. All the mathematical notations used in an educational curriculum have to be linearly described.
- The assignment to each tag of a name to be displayed in the list of tags in the mathematical editor. The name depends on the national languages.
- The assignment to each tag of at least two names to be used in the preparation of the speech output. They are necessary to process the string to be read by the speech synthesizer.

Each localization of LAMBDA is encoded in an XML structure, to enable the LAMBDA-related tools (e.g., the mathematical editor) to easily retrieve information and set the local working environment.

The MathGenie (Karshmer, Bledsoe et al. 2004) has also been designed with language localization capabilities – its design isolates the vocabulary as a user accessible table, and the current distribution includes both an English and a Spanish table.

Nevertheless, true accessibility requires adaptation of presentation based not only on language and notation, but also based on learning styles and teaching methodologies. A few studies have been conducted concerning country-specific issues in math accessibility (Kobolkova and Lecky 2002; Meyer and Jung 2002), but clearly more work is required to fill this gap.

6.2. The Importance of Context

The majority of the research on math accessibility conducted so far concentrates on enabling accessibility of math at the *formula* level. This implies that the presentation is based on analyzing exclusively the individual formula currently under consideration. This approach has inherent limitations; most formulae can only be properly interpreted and presented according to the context they appear in. For example, a formula like $\pi(x+1)$ could be interpreted as

- a permutation of the numbers from 0 to x+1
- the product of π (=3.14159…) and x+1
- the application of the function π to the value x+1

Relatively limited work has been conducted to incorporate *context* in the interpretation and presentation of mathematics. A first step in this direction has been discussed in (Palmer, Pontelli et al. 2003), which addresses the accessibility of OpenMath (Caprotti, Carlisle et al. 2000). Contrary to other digital formats for the representation of mathematical content (e.g., MathML and LaTeX), OpenMath draws the operators used in the construction of mathematical formulae from *Content Dictionaries (CDs)*, which are topic-specific collections of operator definitions – and this allows the analysis and presentation of formulae to have a clear reference to the defining CDs, which provide the appropriate semantics of each operator.

Additional precision in the translation process, and a better connection between *semantics* and *presentation*, can be gained by taking advantage of the *context* in which the formula is used (Pontelli and Palmer 2004). This opens the problem of:

- Determining ways to represent relevant components of the document containing the formulae; for example, formulae appear as part of definitions, theorems, proofs, examples, etc.
- Determining ways to relate relevant document components to the corresponding formulae.
- Determining ways to relate documents to *classes* of documents. For example, documents should be related to classes of documents discussing the same topic – e.g., chapters of a back should be related to the topic of the book (e.g., a book on foundations of statistic).

A natural answer to part of this problem comes from the recent work on OMDoc (Kohlhase 2001; Kohlhase and Franke 2001), which provides a natural markup framework for the encoding of mathematical concepts (e.g., lemmas, propositions, statements, proofs). The current directions investigated are:

- Identification of the relevant document components within presentation formats (e.g., a Braille+Nemeth document) and their explicit representation within OMDoc
- Use of the knowledge encoded in the OMDoc documents to facilitate the translation process – the OMDoc information can be used to provide missing components of the semantics of the formula (e.g., the context provided by OMDoc might determine that the formula is a mathematical logic statement, thus clearly identifying the set of CDs to be used to generate an OpenMath version of the formula).

Observe that the use of OMDoc allows us also to draw knowledge from the existing rich on-line formalizations of mathematics (e.g., as in MBase (Kohlhase and Franke 2001)). On the other hand, the capabilities of OMDoc are limited to the representation of a single document or part of a document. It lacks the capability to describe global properties of a document – e.g., the topic of a document – or global relationships between components of the document. Even worse, the format does not address the problem of relating between and across different documents. Approaches like those presented in MBase, aimed at the creation of a repository of mathematical facts, are note adequate to the needs of accessibility of mathematics, since they are mostly meant to help automated theorem proving systems and not to directly assist human documents. This state of things suggests not only the need of better analysis algorithms, but also the need for a high-level markup language that builds on OMDoc and extends its expressive capabilities to the encoding of whole mathematical documents and organized collections of documents.

6.3. Integration of Components

The previous discussion highlights the complexity of providing non-visual access to mathematical content. In particular, it is clear that a number of dependent issues have to be effectively addressed, ranging from inter-operation between representation formats, adoption of different assistive devices, and modalities of presentation/navigation. These issues are strongly dependent on each others, forcing researchers to address the whole spectrum of options and issues. This has led in recent years to the development of wide-breadth projects that span a variety of aspects of accessibility of mathematical content in an integrated fashion. Two notable efforts in this direction are represented by the LAMBDA Project[5] and the International Universal Mathematics Accessibility group (iGroup UMA)[6].

The LAMBDA project was funded by the European Union with the objective of creating an integrated system for both writing and reading mathematical content for the benefit of blind students. The strategic structure of the project relies on the development of an editor, to write mathematical expressions in a linear way, capable of interacting with different assistive devices, and of a linear mathematical code, designed to be inter-operable with various existing formats for mathematics (including formats used by advanced systems for the manipulation of mathematics, e.g., Mathematica). The international collaboration involves teams from the U.K., Italy, Germany, Portugal, Greece, Spain, and France.

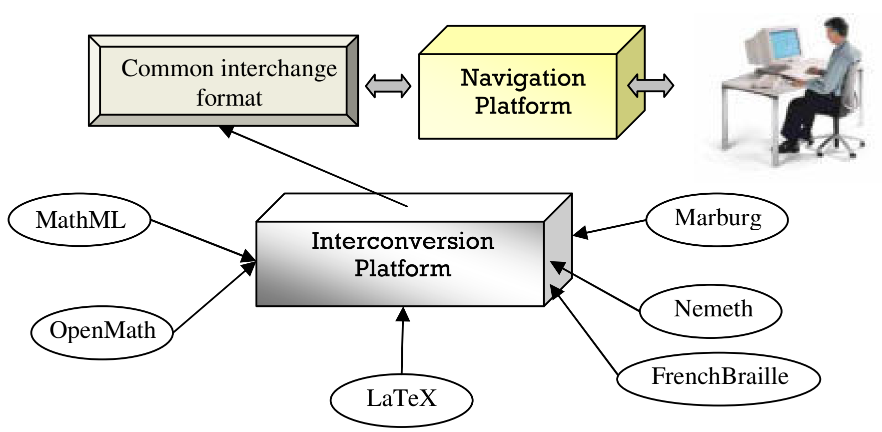


Figure 4: System Organization

[5] http://www.lambdaproject.org
[6] http://karshmer.lklnd.usf.edu/~igroupuma/index.html

The iGroup UMA (Karshmer, Gupta et al. 2004) is an international cooperation, involving researchers from Ireland, Austria, France, Japan, and the U.S.A. The goal of the cooperation is to develop tools to enhance accessibility of mathematical content across different representation formats (e.g., Braille and digital formats), across National styles and conventions, and across different levels of visual capabilities. Figure 4 shows the overall organization of the project. The project provides a uniform inter-conversion platform which allows two-directional translation between a wide range of formats for the representation of mathematics. The inter-conversion platform relies on the use of a common interchange format (MathML in the case of the current version of the project) which is used as a bridge between any pair of formats considered. The project provides also tools for the rendering and navigation of mathematical content – currently this component is realized within the previously described MathGenie.

7. Conclusion

The problems associated with teaching mathematics to the visually impaired are long standing and difficult problems. With the advent of the reasonably inexpensive computer, many partial solutions have been put forward, as described in the body of this work. Each has added knowledge required to make incremental advances in the state of the art.

Additionally, the problems associated with math and blindness are now being studied from an interdisciplinary perspective, adding needed fundamental understanding to the design and development process. The data generated by these efforts have been invaluable to the understanding of the problem's nature and potential solution.

Finally, and most importantly, is the advent of real international cooperation in understanding the dimensions of the problem and their potential solution. As most governmental units do not view the problem as one of great import, and the obvious lack of potential wealth to be derived from solving the problem, the international research community has banded together to make essential breakthroughs in the past decade. This effort is to be applauded and encouraged.

The future looks brighter than ever before. With new research in basic life sciences and understanding of the brain and nervous system, we should see dramatic advances in the near future. We look to a bright future in which our colleagues in scientific disciplines will be playing on a level field regardless of disabilities.

Acknowledgements

The authors wish to acknowledge the collaboration of A. Annamalai, D. Archambault, M. Batusic, C. Bledsoe, D. Fitzpatrick, D. Gillan, D. Gopal, H. Guo, K. Miesenberger, V. Moço, B. Palmer, H. Reddy, P. Stanley, and B. Stöger.

References

Annamalai, A., D. Gopal, et al. (2003). INSIGHT: a comprehensive system for converting Braille based mathematical documents to LaTeX. Universal Access in HCI, LEA, pp. 1245-1249.

Archambault, D., M. Batusic, et al. (2005). The universal maths conversion library: an attempt to build an open software library to convert mathematical contents in various formats. Universal Access in Human-Computer Interaction, Las Vegas, pp. CD-ROM.

Archambault, D., D. Fitzpatrick, et al. (2004). Towards a universal maths conversion library. International Conference on Computers Helping People (ICCHP), Paris, Springer Verlag, pp. 664-669.

Archambault, D. and V. Moço (2006). Canonical MathML to simplify conversion of MathML to Braille mathematical notations. International Conference on Computers Helping People (ICCHP), Linz, Springer Verlag, pp. 1191-1198.

Baddeley, A. D. (1992). Your memory: a user's guide, Penguin Books.

Batusic, M., K. Miesenberger, et al. (1998). Labradoor: a contribution to making mathematics accessible for the blind. International Conference on Computers Helping People (ICCHP), Wien, Springer Verlag, pp.

Batusic, M., K. Miesenberger, et al. (2003). Parser for the Marburg Mathematical Braille Notation NIDRR Project: Universal Math Converter. Human-Computer Interaction, 10th International Conference, Crete, Greece, L. Erlbaum, pp.

Biblioteca Italiana per Ciechi "Regina Margherita" O.N.L.U.S. (1998). Codice Braille Italiano.

Blattner, M., D. Surnikawa, et al. (1989). "Earcons and icons: their structure and common design principles." Human Computer Interaction **4**(1): 11-44.

Braille Authority of the United Kingdom (1987). Braille Mathematics Notation, Mathematics Committee.

Braille, L. (1829). Method of Writing Words, Music, and Plain Songs by Means of Dots, for Use by the Blind and Arranged for Them (in French)

Brewster, S. A., P. Wright, et al. (1994). A detailed investigation into the effectiveness of earcons. International Conference on Auditory Displays, Addison Wesley, pp. 471-498.

Caprotti, O., D. P. Carlisle, et al. (2000). The OpenMath Standard, The OpenMath Esprit Consortium.

Chang, L. A. (1983). Handbook for Spoken Mathematics (Larry's Speakeasy), Lawrence Livermore Laboratory, The Regents of the University of California.

Commission Evolution du Braille Francais (2001). Notation Mathematique Braille, mise a jour de la notation mathematique de 1971.

Crombie, D., R. Lenoir, et al. (2004). math2braille: Opening Access to Mathematics. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 670-677.

Duxbury Systems (2000). MegaMath Translator for MegaDots.

Edwards, A. D. N., H. McCartney, et al. (2006). *Lambda:* a multimodal approach to making mathematics accessible to blind students. International Conference on Computers and Accessibility, Portland, OR, ACM Press, pp.

Epheser, H., D. Pograniczna, et al. (1992). Internationale Mathematikschrift für Blinde. Marburg (Lahn), Deutsche Blindenstudienanstalt.

Fateman, R. (2006). How can we speak math? The evolution of mathematical communication in the age of digital libraries, University of Minnesota, pp.

Ferreira, H. and D. Freitas (2004). Enhancing the Accessibility of Mathematics for Blind People: the AudioMath Project. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 678-685.

Fitzpatrick, D. (2002). Speaking technical documents: using prosody to convey textual and mathematical material. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 494-501.

Fitzpatrick, D. (2006). Mathematics: how and what to speak. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 1199-1206.

Fogarolo, F. (2006). Maths and blind students: the LAMBDA project, CSA Vicenza.

Fogarolo, F., C. Bernareggi, et al. (2005). Handbook for the LAMBDA Maths Editor vs. 3.35.

Fukuda, R., N. Ohtake, et al. (2000). Optical recognition and Braille transcription of mathematical documents. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 711-718.

Gardner, J. (2003). DotsPlus Braille Tutorial: simplifying communication between sighted and blind people. CSUN, pp.

Gardner, J. (2005). from http://dots.physics.orst.edu/dotsplus.html.

Gardner, J. (2005). WinTriangle: a scientific word processor for the blind.

Gardner, J., L. Ungier, et al. (2006). Braille Math Made Easy with the Tiger Formatter. International Conference on Computers Helping People (ICCHP), Linz, Springer Verlag, pp. 1215-1222.

Gaura, P. (2002). REMathEx - reader and editor of the mathematical expressions for blind students. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 486-493.

Gill, J. (2006). Information Resources for People Working in the Field of Visual Disabilities, Royal National Institute for the Blind.

Gillan, D., P. Barraza, et al. (2004). A cognitive analysis of equation reading applied to the development of assistive technology for visually-impaired students. HFES 48th Annual Meeting, New Orleans, pp.

Gillan, D. J., P. Barraza, et al. (2004). Cognitive analysis of equation readings: application to the develompent of the MathGenie. International Conference on Computers Helping People (ICCHP), Paris, France, Springer Verlag, pp.

Gopal, D., Q. Wang, et al. (2007). Towards Completely Automatic Backtranslation of Nemeth Braille Code. HCI International, L. Erlbaum, pp.

Hopcroft, J. E., R. Motwani, et al. (2000). Introduction to Automata Theory, Languages, and Computation, Addison Wesley.

Karshmer, A. I., C. Bledsoe, et al. (2004). The architecture of a comprehensive equation browser for the print impaired. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 614-619.

Karshmer, A. I. and D. Gillan (2003). How well can we read equations to blind mathematics students: some answers from psychology. Human Computer Interface International Conference, Crete, pp.

Karshmer, A. I., G. Gupta, et al. (1998). Reading and writing mathematics: the MAVIS project. International Conference on Assistive Technologies (ASSETS), Marina del Rey, CA, ACM Press, pp. 136-143.

Karshmer, A. I., G. Gupta, et al. (2002). Architecting an auditory browser for navigating mathematical expressions. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 477-485.

Karshmer, A. I., G. Gupta, et al. (2004). UMA: a system for universal mathematics accessibility. Conference on Computers and Accessibility, ACM Press, pp. 55-62.

Kobolkova, M. and P. Lecky (2002). Experience with access to mathematics for the blind students in Slovakia. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 510-511.

Kohlhase, M. (2001). OMDOC: towards an Internet standard for the administration, distribution and teaching of mathematical knowledge. Artificial Intelligence and Symbolic Computation, Springer Verlag, pp. 32-42.

Kohlhase, M. and A. Franke (2001). "MBase: representing knowledge and context for the integration of mathematical software systems." Journal of Symbolic Computation **32**(4): 365-402.

Komada, T., K. Yamaguchi, et al. (2006). New environment for visually disabled students to access scientific information by combining speech interface and tactile graphics. International Conference on Computers Helping People, Springer Verlag, pp. 1183-1190.

Krufka, S. E. and K. E. Barner (2005). Automatic Production of Tactile Graphics from Scalable Vector Graphics. International Conference on Computers and Accessibility (ASSETS), Baltimore, ACM Press, pp. 166-172.

Lamport, L. (1985). LaTeX: a Document Preparation System, Addison Wesley.

Lee, C. (2005). "Filling the math/science teacher void." UCLA Today **26**(1).

McClellan, N. (2007). "The MathTalk System." 2007, from http://www.metroplexvoice.com/tech_notes.htm.

Meyer, E. and M. Jung (2002). LaTeX at the University of Applied Sciences Giessen-Friedberg - experiences at the Institute for Visually Impaired Students. International Conference on Computers Helping People, Springer Verlag, pp. 508-509.

Moço, V. and D. Archambault (2003). Automatic translator for mathematical Braille. Universal Access in HCI, LEA, pp. 1335-1339.

National Library of Service (2000). Braille: Into the Next Millenium, Library of Congress.

Nemeth, A. (1972). The Nemeth Braille Code for Mathematics and Science Notation, 1972 Revision, American Printing House for the Blind.

O'Malley, M. M., D. Kloker, et al. (1973). "Recovering parentheses from spoken algebraic expressions." IEEE Transactions on Audio and Electroacoustics **AU-21**: 217-220.

Oshry, M., R. J. Auburn, et al. (2006). Voice Extensible Markup Language. W3C Working Draft.

Palmer, B., E. Pontelli, et al. (2003). Experiments in Translating and Navigating Digital Formats for Mathematics. HCI International, LEA, pp.

Pepino, A., C. Freda, et al. (2005). BlindMath: an Innovative Scientific Editor. Southwest Conference on Disability, Albuquerque, NM, pp.

Pontelli, E. and B. Palmer (2004). Translating between formats for mathematics: current approach and an agenda for future developments. International Conference on Computers Helping People (ICCHP), Paris, Springer Verlag, pp. 620-625.

Raman, T. V. (1994). Audio Systems for Technical Reading, Cornell University. **Ph.D.**

Reddy, H., N. Annamalai, et al. (2004). Dynamic Navigation of VoiceXML Documents. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 337-354.

Reddy, H., G. Gupta, et al. (2005). Dynamic Aural Browsing of MathML Documents with VoiceXML. Human-Computer Interaction, Las Vegas, L. Erlbaum, pp.

Scadden, L. (1996). Making Mathematics and Science Accessible to Blind Students Through Technology. RESNA, pp.

Schwebel, F. (2004). "BraMaNet: logiciel de traduction des mathématiques en Braille." from http://handy.univ-lyon1.fr/projects/bramanet.

Schweikhardt, W., C. Bernareggi, et al. (2006). LAMBDA: a European system to access mathematics with Braille and audio synthesis. International Conference on Computer Helping People (ICCHP), Springer Verlag, pp. 1223-1230.

Schweikjardt, W. (1998). Stuttgarter mathematikschrift für blinde, Universität Stuttgart, Institut für Informatik.

Soiffer, N. (2005). Advances in Accessible Web-based Mathematics. CSUN, California State University Northridge, pp.

Stanley, P. and A. I. Karshmer (2006). Translatinve MathML into Nemeth Braille code. International Conference on Computers Helping People (ICCHP), Linz, Austria, Springer Verlag, pp.

Stanley, P. B., C. Bledsoe, et al. (2004). Utilizing Scalable Vector Graphics in the Instruction of Mathematics to the Print Impaired Student. International Conference on Computers Helping People (ICCHP), Paris, France, Springer Verlag, pp. 626-629.

Stevens, R. D. (1996). Principles for the design of auditory interfaces to present complex information to blind people, Universty of York. **Ph.D.**

Stevens, R. D., A. D. N. Edwards, et al. (1997). "Access to mathematics for visually disabled students through multi-modal interaction." Human-Computer Interaction **12**(1-2): 47-92.

Stöger, B., M. Batušić, et al. (2006). Supporting blind students in navigation and manipulation of mathematical expressions: basic requirements and strategies. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 1235-1242.

Stöger, B., K. Miesenberger, et al. (2004). Mathematical working environment for the blind: motivation and basic ideas. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 658-663.

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

Suzuki, M. (2005). "Infty Project: about ChattyInfty." from http://www.inftyproject.org/download/AboutChattyInftyE.txt.

Suzuki, M., T. Kanahori, et al. (2004). An integrated OCR software for mathematical document and its output with accessibility. International Conference on Computers Helping People (ICCHP), Springer Verlag, pp. 648-655.

Thompson, D. M. (2005). LaTeX2Tri: physics and mathematics for the blind or visually impaired. CSUN Conference, California State University Northridge, pp.

Walsh, P., R. Lundquist, et al. (2001) The Audio-accessible Graphing Calculator. CSUN **Volume**, DOI: http://www.csun.edu/cod/conf/2001/proceedings/csun01.htm