AudioMath – using MathML for speaking mathematics

Helder Ferreira, Diamantino Freitas

Laboratório de Sinais e Sistemas, Faculdade de Engenharia Universidade do Porto
R. Dr. Roberto Frias s/n,
4200-465 Porto, Portugal
{hfilipe,dfreitas}@fe.up.pt
http://lpf-esi.fe.up.pt

Abstract. How can blind people surpass the difficulty in reading online documents containing mathematical expressions? Usually they can't. Technical, scientific or even simple documents presented online that involve math expressions, issue a big accessibility problem. One possible solution is to create the contents using mark-up and in the retrieval phase to parse and interpret such contents and convert them into an audio format. This remains a complex, multidisciplinary problem, currently addressed by the AudioMath project, providing conversions from math expressions in W3Cs MathML format into marked-up text for text-to-speech engine understand and read-out. This paper reviews the state-of-the-art in publishing Web scientific documents, introduces the accessibility problems and reviews the problem of speaking mathematics. The main scientific and technical challenges, specifically at the levels of interpretation of the XML content of the math expressions and of conversion into spoken form are considered, as well as the current status of the work.

1 Introduction

Since the beginning of times that mankind feels the need to communicate. Language allows us to structure thinking, transmit feelings, record what we know and communicate with each others. The publication and distribution of scientific papers and articles, as well as e-learning websites, is a desirable way to spread knowledge, promote education and stimulate research and development.

Nowadays this knowledge is not only available as books, magazines or conference proceedings, but also as web pages or digital files on the Internet, reachable by "almost" everyone. The upcoming of Internet improved the availability and access to information, but introduced new problems. A web document can contain: text, images (graphics, diagrams or pictures), mathematical expressions, applets, scripts, multimedia (Flash, Shockwave…) or code blocks. Each one of these elements might present accessibility problems in the reading and interpretation of the document.

Although the existence of the Web Content Accessible Guidelines (WCAG) [1] from W3C, that provide a set of rules which can be applied to resolve these problems, at this time the accessibility of technical and scientific documents, especially on mathematical expressions, is extremely weak.

Most of the web documents containing mathematical materials utilize images (jpg, gif, png) created by TeX related software or Microsoft Word, to display the mathematical formulae. Some distribute the documents in PDF, TeX, Postscript, Word or RTF file types. The use of Java applets or plug-ins is also frequent.

This has lead to accessibility problems in the reading of documents containing mathematical expressions for visually impaired persons, and, in particular, for the blind. Their difficulty rises and accessibility diminishes, as the technical level of documents increases, even if the contents are already in a digital format and properly equipped computers are used [2].

The growing concern about accessibility to mathematical expressions on the Internet has taken research groups such as: W3C [3], OpenMath [4] and AMS [5], to create and develop initiatives in this area, promoting and regulating the accessible publication of mathematical contents over the Internet. One of those initiatives is the creation of the *MathML (Mathematical Mark-up Language)* [6] from W3C, which is used in this project.

The work presented in this paper refers to the problem of online accessibility to mathematical expressions and presents the tool *AudioMath*, which uses MathML and text-to-speech (TTS) technology to create audio versions of the mathematical contents. This technical solution has been adopted taking into consideration two principles: first, the audio medium is accessible and general purpose TTS engines are available; and second, it's assumed that math can be spoken out.

This paper is structured in the following sections: - Section 1 introduces the problem; - Section 2 reviews the current state of art; - Section 3 presents some considerations about MathML audio rendering; - Section 4 makes a brief overview analysis on 2 related tools; - Section 5 describes the tool AudioMath; and Section 6 presents the conclusions and the forthcoming work.

2 State of the art

This section reviews the current state of art in publishing mathematical contents online, and the current related projects in mathematical audio rendering.

2.1 Publishing technical documents in the Internet

The publication of scientific documents containing mathematical formulae is extremely demanding. The appearance of the TeX [7] system developed by Donald Knuth solved the majority of problems with printed documents. Then WYSIWYG (*What You See Is What You Get*) editor's appeared and subsequently mark-up languages for Internet, such as HTML (*Hypertext Mark-up Language*). However, HTML *per se* doesn't allow the use of a mathematical description language directly into the document. Being so, alternatives were developed, such as the use of:

– **(X)HTML** [8] **+ Images (JPG, GIF, PNG, SVG)**: the use of math expressions as images (vector or raster) is non accessible because usually don´t provide a text equivalent.
– **HTML + Symbol Fonts / (X)HTML + CSS** [9]: usually they use tables to structure information. No semantic meaning is provided. Ex: *translator TtH*. [10].
– **Applets**: these are used to generate mathematical expressions; it's a slow and non accessible process. Ex: *WebEQ*. [11].
– **Word / RTF / PDF / Postscript / TeX / Latex**: the HTML documents produced by these materials usually represent math expressions in the form of images. Ex: *TeX4tht* [12] (latest versions can now create MathML).
– **Mathematical Markup Language (MathML)**: one of the most accessible solutions.

MathML is becoming one of the most adopted formats for publishing math online since it is:

– Being developed by W3C and is becoming a standard.
– Rapidly growing and being used by several relevant organizations associated with the teaching and learning of mathematical contents, as well as the increasing involvement of software houses.
– Supported by several editors and applications able to create and manipulate MathML documents.
– The existence of conversion tools for the main publishing formats.
– The fact that it is a mark-up language allows its parsing, interpretation and conversion to other formats, and consequently a higher accessibility, portability and independence – it's XML [13]!

For all this reasons, MathML is one of the best supporting technologies for publishing online, manipulating formulae and speaking mathematics.

2.2 Audio rendering of mathematical expressions

It is known that most people read text letter-by-letter beginning at or near the leftmost symbol in a text line and often engage in backward scans to retrieve information that was previously read. But reading mathematics is very different than reading a plain text, the first one is a lot more complex. Reading math like a normal book doesn't allow us imagine the elements and the relationships between them. Formal structures and rules are required.

To start with, mathematics is a two-dimensional written language. This is quite different from almost all the other ordinary languages that are primarily spoken and later on written, and both in a one-dimensional form (in a clearly defined sequence).

For a non-visual person, understanding a mathematical formula requires a repeated scan and jumping over secondary portions. However, this can be a very complex task to blind people [14]; therefore studies should be done to understand how we should provide a correct access to math expressions. Unfortunately there is an almost complete lack of studies on how to read mathematical expressions, mainly on mathematical prosody [15] [16].

The *MathSpeak Project* [17] is one of the proposed methods, consisting of a group of rules to dictate mathematical contents. However it is not a standard and it is intended to serve blind people that want to transcribe their documents into *Nemeth Code* [18], and later on into Braille.

Another method comes from the work of *T.V. Raman* [19], the most impressive and recognized study in this area (Aster system), consisting of a mathematical notation that takes into consideration several audio dimensions that make up the various pieces of the notation. However this only works for TeX related documents and has limitations, such as no provision for intra-formula navigation in the mathematical expression. Nevertheless, this work points out several generally useful clues for audio rendering of mathematics.

Arthur Karshmer is another influent researcher working in the area that is trying to study how to read mathematical expressions to blind students [20].

Design Science [21] is also working with audio rendering of MathML. Its product MathPlayer version 2.0 [22] is an example of it. However this plug-in has ambiguous rendering in some mathematical expressions. It's therefore a work in progress.

3 Audio rendering of MathML

In section 2.1 it was stated that MathML is one of the best supporting technologies for publishing online, manipulating formulae and speaking mathematics. Therefore, it is the authors' intention to make some considerations about using MathML for mathematical audio rendering.

3.1 Which MathML markup to use?

W3C's MathML Specification [23] states that "*MathML allows authors to encode both the notation which represents a mathematical object and the mathematical structure of the object itself.*" In other words, representation of a math expression is perceived by two distinct but associated concepts:

– *Visual structure or notation* (ex: a/b, a^{-1}) – *Presentation Markup*: since it concerns the notation it is ambiguous on the semantics. Adaptation for audio rendering is possible.
– *Meaning it represents* (ex: a divided by b) – *Content Markup*: since it is used for semantic meaning, it is ambiguous on the notation. Because it encloses the semantics, it is in theory the best to use for audio rendering.

The relationship between notation (Presentation) and meaning (Content) is not univocal: one meaning, more than one notation.

Also, it seems that Content Markup would suit better for audio rendering, however there are 2 major open issues: Content markup only covers basic math; and the majority of published MathML online is using Presentation markup.

Therefore, MathML Presentation Markup is for the moment the best choice. The downside is that requires a relatively much higher effort in the interpretation of given mathematical expressions, since it is prepared for the enclosure of visual information and not of semantics.

3.2 Interpretation of MathML tag set elements and attributes

Not all the elements and attributes need to be processed for audio rendering. For instance, styles and visual attributes not always provide extra information about the expression, and rarely enhance the audio description. However if Presentation markup is used some style attributes might be important to disambiguate meaning. For example, consider the following expression:

$$\left(\frac{2}{3}\right) \neq \binom{2}{3}. \tag{1}$$

The left side of the expression is using the `mfrac` element with the attribute `linethickness` different from zero (becoming a fraction), but the right side of the expression uses `linethickness` equal to zero (changing the meaning of the expression to combinatorial number).

Other elements and attributes are important in the Presentation markup, for example: `mathvariant` (because different types of variants have different meanings, for example: *italic* might indicate a function, *bold* might indicate a vector, *fraktur* might indicate lie algebra) and `mrow` (which is usually used to enclose mathematical sub expressions, giving precious clues to audio rendering).

When using Content markup, almost every elements are useful once they all enclose semantic meaning about the mathematical expression.

MathML also has some special characters that are extremely important for audio rendering: `⁡` (allows an audio render to know it's a function of something) and `⁢` (allows an audio render to know it is a product even if the symbol is not there). For example:

```
xy - <mi>x</mi><mo>&InvisibleTimes;</mo><mi>y</mi>
```

Without the `⁢` it will render "*x y*". With the special character it will render "*x times y*".

4 Mathematical Audio Rendering Tools Overview

This section gives a brief overview of the only tools, besides AudioMath, that are able to produce mathematical audio rendering: ASTER and MathPlayer.

4.1 Audio system for technical readings - ASTER

ASTER stands for *Audio System For Technical Readings* [19] and is an application that accepts TeX notation (LaTeX [24]) as input and produces audio rendering as output. It has been developed by T.V. Raman in 1994 during his PhD and uses an Emacs [25] front-end (Linux platform). ASTER has 3 main components: *Latex parser* that creates an internal representation easier for the program to manipulate; AFL *(Audio Formatting Language)* which is used to render the parsed text using speech and other sounds; and *Browser* used to help the audio rendering.

ASTER was a breakthrough. TV Raman's work it's considered a bible in mathematics audio rendering and it supports a large number of mathematical formulae in Latex. However no math formulae navigation is supported. To suppress this need, ASTER uses a variable substitution process: complex expressions can be divided in sub expressions on user's request.

4.2 MathPlayer 2.0

MathPlayer is a mathematics display engine for Microsoft's Internet Explorer 6.0, developed by Design Science. It uses MathML Presentation markup as input and visual rendering (version 1.0 and 2.0) and audio rendering (only in version 2.0 out in 2004) as output.

Since MathPlayer is too recent, it has a few problems yet to be solved: system of equations and matrices are detected both as tables (it is not capable to detect a more detailed semantic meaning), and there is a lack of some `begin <operator>` and `end <operator>` keywords, making the audio result sometimes ambiguous. Also, no math formulae navigation is provided, so it gets complicated with complex math expressions; and there are no *usermodes* (the way people like to hear something).

5 AudioMath

Since the first appearance of MathML (1999), and the emerging support to it by several browsers, that the future of publishing mathematics online seems to have assured a certain degree of quality (better browsing, presentation, manipulation, flexibility and accessibility). Even the previous works in LateX are now being converted into XHTML+MathML.

For these reasons, the need to build an accessibility tool that would be able to deal with MathML was justified. AudioMath was the first tool, as far as the authors know, to be able to speak MathML contents. Only months later, Design Science included the feature "Speak Math" on MathPlayer 2.0.

This section describes the tool, *AudioMath* [26], that the Laboratory of Signals and Systems from the Faculty Engineering University of Porto is developing to enable MathML audio rendering.

5.1 The AudioMath project

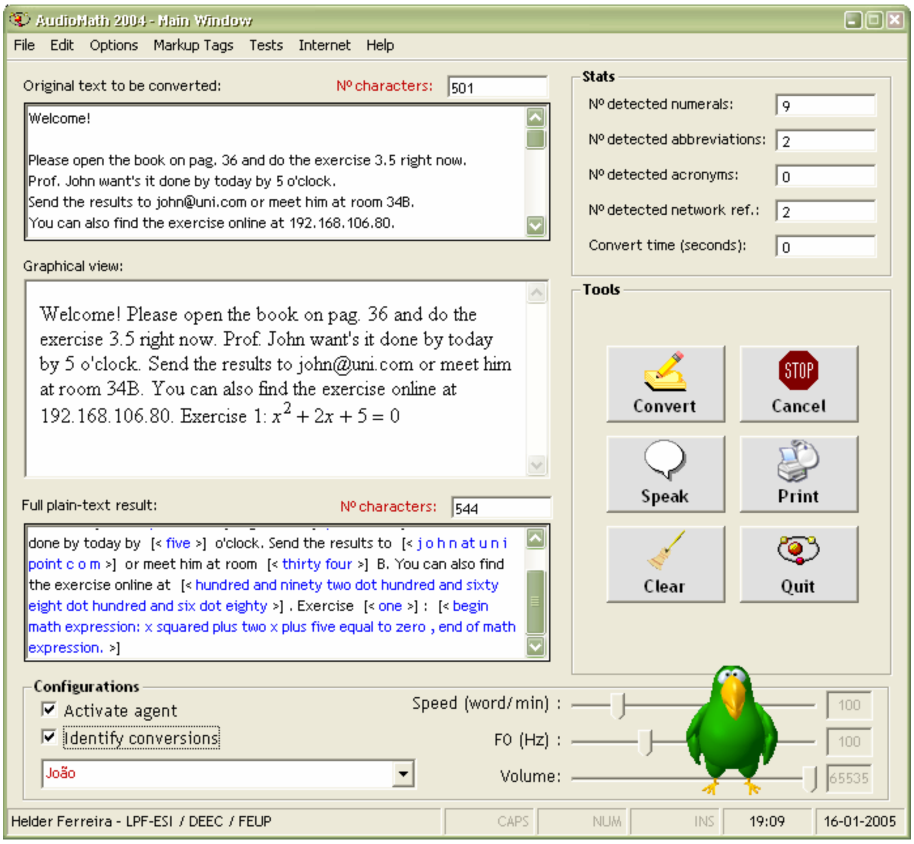

Fig. 1. AudioMath Screenshot – English version.

The AudioMath project aims is to build a tool to operate either as standalone or integrated in a speech interface (TTS text-to-speech), capable of:

– *Mathematics Audio Rendering*: parsing, interpretation and conversion of MathML into plain text format; generation of the appropriate prosodic contour for reading of the formula's text; and development of an intra-formula browsing device (navigation).
– Recognition and conversion of any other text or markup elements not directly "understandable" by the TTS Engine, such as: numerals, abbreviations, acronyms and network references.

AudioMath is an ActiveX dynamic link library (dll) that can be used by any program through internal calls. It is currently developed in PERL [27] and for Windows environment. However its main modules can also be used in Linux/Unix due to PERL's portability.

Its main applications are: reading of technical and scientific documents online in an accessible way; teaching and learning how to read math; and enhancing general accessibility to computer-based applications, when applied to a TTS engine.

5.2 AudioMath architecture

AudioMath has been built in a modular, extensible and configurable architecture. Currently it contains 6 major modules: *Numerals* (conversion of several types of numeric forms), *Abbreviations* (conversion of abbreviations in a text), Acronyms (conversion of acronyms in the document), *Network References* (conversion of IPs, emails and URI/URLs), *Mathematical* (conversion of MathML expressions) and *Auto-Discovery* (the "brain" of the operation that recognizes or identifies elements in the document and calls the respective conversion modules).

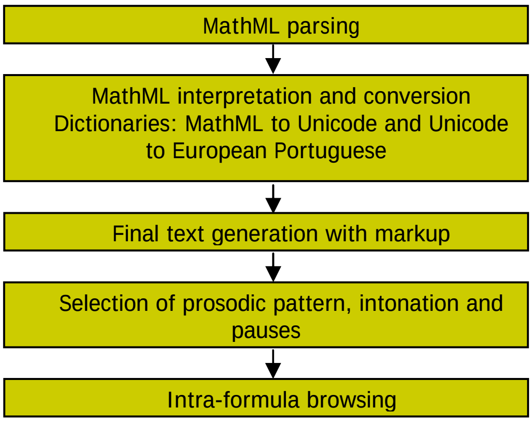

Fig. 2. AudioMath Architecture Overview.

5.3 The AudioMath process

The AudioMath's process can be divided into 3 functional parts: *Text Analysis*, *Parsing and Interpreting MathML* and *Converting and Speaking Mathematical Contents*.

There are several types of text elements in a document (acronyms, abbreviations, numerals, web references, math expressions or even special Unicode characters or math glyphs [28]), justifying the reason for text analysis algorithms.

To speed up the process the document should be divided into blocks of text, splitting MathML markup from the rest of the text. Text processing is strongly based on regular expressions [29] [30] and databases (for acronyms and abbreviations). Dictionaries and databases were included inside the code as hash tables (more speed but less flexibility in updates). All the conversion algorithms support *user modes,* i.e., preferences on audio rendering. For example: 1.25 can be spoken: *one point two five*, or *one point twenty-five*.

Currently, in AudioMath, interpreting Presentation markup is a process of raising flags each time a starting and ending tag is detected (SAX parser – event based), which allows to know the history of the markup and to retrieve information, enabling to understand the structure of the math expression and do its conversion. The identification of sub-expressions and its conversion also allows the composition of a bigger and complex math formula. As the expression is becoming discovered, the conversion process takes place by calling several algorithms as well as Unicode and MathML dictionaries. AudioMath uses 2 kinds of dictionaries: MathML entities to Unicode (`⁢` is converted to `U+02062`) and Unicode to European Portuguese full plaintext form (`U+02062` is converted to `vezes(=times)`). The conversion to the full plaintext form is done according to a database of vocabulary and speech rules.

5.4 Database of vocabulary and speech rules

One of the AudioMath's current tasks is the definition of a database of vocabulary and speech rules, for several subsets of math formulae. These rules and intonation will be implemented at conversion time by tagging the text (using SSML [31]) with prosodic marks, to command the TTS engine in order to produce the required pauses and f0 modulations.

Math corpus is being defined and categorized by different semantic types: basic algebra, integrals, roots, powers, derivatives, complex numbers, matrices, systems of equations, trigonometry, series, theorems and definitions, Greek alphabet, *etc*.

Each type has its structure/concept analysed and defined in a formal plaintext way. Different readings of the same formula are spoken and analyzed for prosody and perceiveness (user evaluation). Pitch patterns and pauses are inferred from speech. For example, consider the following expression:

$$\sqrt{a^2 + b^2}\ . \qquad (2)$$

Can be spoken as: "*square root of (pause) a squared (mini-pause) plus b squared (pause) end of radicand*".

We can see that there are 2 types of pauses: *large pause* - precede lower levels in hierarchy; *short pause* – optional, but used before some operators and between the arguments.

Also, there are rising and falling movements of f0 in the speaker's intonation intended to provide classification of the boundaries introduced by the pauses: - *rising tone*: used when lower hierarchical level is starting (ex: root of); - *falling tone*: used when level is ended (ex: b squared); - *falling and rising tone*: used to clarify the smaller separating pause (ex: a squared); and *emphatic falling tone*: used at the end of the expression (ex: end of radicand).

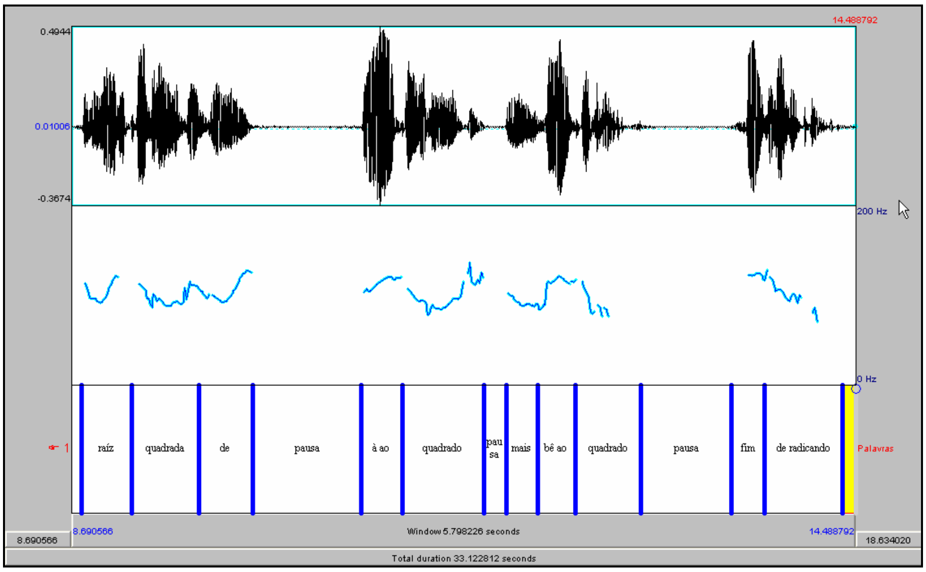

Fig. 3. Speech waveform of the previous mathematical formula example (2).

5.5 Current status and future work

In spite of the vast research this project has done in the field of speaking math using MathML, AudioMath is still a work in progress.

The tool currently supports several tags of MathML Presentation Markup and Unicode, and it is also able to detect and convert several types of numerals (cardinals, ordinals, roman numbers, dates…), abbreviations, acronyms and network references (emails, ip addresses …). Each module, used for detection and conversion, supports user preferences on the MathML audio rendering.

A browser has been built for functionality test and TTS integration. User evaluation is also performed in several iterations of the AudioMath's development, especially in the building of its mathematical corpus and rules for reading math. A few studies on mathematical prosody have also been made.

AudioMath's future work includes adding support for MathML Content Markup; completing the support to the MathML Presentation Markup; further learning and surveying on how to read mathematical formulae; develop modules that support HTML, XHTML, SSML and others; providing mechanisms for navigating inside mathematical formulae (eventually a special audio browser); adding support for new languages (English, French…); and finally, to develop the study on the prosody of reading mathematics.

6 Conclusions

The research developed along this project allowed us to conclude that, reading mathematical contents is very different than reading a text. The full verbal understanding of a math structure and its meaning requires, on the majority of cases, the presence of some kind of navigational mechanism; otherwise users will experience serious difficulties due to the human short-term memory. Also, the use of prosody should be combined with those mechanisms, to ensure a more natural and perceivable experience to the users. However, there is currently lack of studies on math prosody, which means there is yet no standard on how to read math!

The increase of applications and online materials that deal with MathML motivate the use of this markup language on tools that have in mind the audio rendering of math. The choice of the markup set that one should adopt, for MathML audio rendering, is not an easy one. In theory, Content Markup should be better, once it encloses the semantic meaning of the math expression; however Presentation Markup is more widely used, becoming the first choice for the moment. Also, the developer should take into consideration that not all the elements and attributes of the 2 markup sets are needed for the audio rendering (attributes that convey colour information, for instance).

The solution presented here, by AudioMath, concerns a work in progress of an accessibility tool that creates audio versions of the MathML Presentation Markup mathematical contents, with user modes support, i.e., it allows users to configure the MathML rendering to their preferences. AudioMath is built in a modular, extensible and configurable architecture that allows it to be upgraded easily. The use of special dictionaries, one for MathML entities and other for Unicode entities, allows this tool to adapt to several online realities. This project also studies the mathematical prosody and it's committed to the building of a speech rules database.

In conclusion, the almost lack of references, studies and standards on "how to use MathML for speaking mathematics" and "how to read math", prove the need and importance of this project and its work in progress to build the tool AudioMath [32][33][34].

References

1. W3C: Web Content Accessible Guidelines 1.0 (1999) - http://www.w3.org/TR/WCAG10/ .
2. Monaghan, Alex. Fitzpatrick, Donal. *Browsing Technical Documents: Document Modelling And User Interface Design*. BULAG99 Proceedings. 1999.
3. World Wide Web Consortium (W3C) – http://www.w3.org .
4. The OpenMath Group - http://www.openmath.org/cocoon/openmath//index.html .
5. American Mathematical Society (AMS) – http://www.ams.org .
6. Mathematical Markup Language (MathML) – http://www.w3.org/Math .
7. TeX Users Group Homepage – http://www.tug.org .
8. XHTML 1.1 - http://www.w3.org/TR/xhtml11/ .
9. CSS 2.1 - http://www.w3.org/TR/CSS21/ .
10. Translator Tth Homepage – http://hutchinson.belmont.ma.us/tth .

11. Design Science WebEQ Homepage - http://www.dessci.com/en/products/webeq/ .
12. TeX4ht - http://www.cse.ohio-state.edu/~gurari/TeX4ht/mn.html .
13. Extensible Markup Language Specification - http://www.w3.org/TR/REC-xml/ .
14. Stöger Bernhard, et al. *Mathematical Working Environment for the Blind Motivation and Basic Ideas*. ICCHP04 Proceedings. 2004.
15. Fitzpatrick D. and Karshmer A.I. *Multi-modal Mathematics: Conveying Math Using Synthetic Speech and Speech Recognition*. ICCHP04 Proceedings. 2004.
16. Stevens R., *Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People*. PhD thesis. Department of Computer Science, January 1996.
17. MathSpeak - http://www.rit.edu/~easi/easisem/talkmath.htm .
18. Learn Nemeth Code – http://www.tsbvi.edu/math/math-nemeth.htm .
19. T.V. Raman. *Audio System for Technical Readings*. Dissertation to the Faculty of the Graduate School of Cornell University. 1994. Home Page: http://www.cs.cornell.edu/Info/People/raman/raman.html .
20. Karshmer, Arthur. *How Well Can We Read Equations to Blind Mathematic Students?* HCII2003 Proceedings Volume 4. 2003.
21. DesignScience Home Page - http://www.dessci.com/ .
22. MathPlayer 2.0 - http://www.dessci.com/en/products/mathplayer/ .
23. MathML W3C's Recommendation 2.0 - http://www.w3.org/TR/2003/REC-MathML2-20031021/ .
24. LateX - http://www.latex-project.org/ .
25. GNU Emacs - http://www.gnu.org/software/emacs/emacs.html .
26. AudioMath Home Page – http://lpf-esi.fe.up.pt/~audiomath .
27. Practical Extraction and Report Language - http://www.perl.org/ .
28. Unicode characters and math glyphs, MathML, W3C - http://www.w3.org/TR/2003/REC-MathML2-20031021/chapter6.html#chars.mathmlchars .
29. Friedl, Jeffrey. *Mastering Regular Expressions 2nd Edition*. O'Reilly. 2002. ISBN: 0-596-00289-0.
30. Jurafsky, Daniel and Martin, James. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice-Hall. 2000. ISBN: 0-13-095069-6.
31. Speech Synthesis Markup Language (SSML) - http://www.w3.org/TR/speech-synthesis/ .
32. Ferreira, Helder. *AudioMath: Speaking Mathematics with MathML*. 2nd European Workshop on MathML & Scientific e-Contents. 2004.
33. Ferreira, Helder and Freitas, Diamantino. *Audio Rendering of Mathematical Formulae using MathML and AudioMath*. UI4ALL Proceedings. 2004.
34. Ferreira, Helder and Freitas, Diamantino. *Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project*. ICCHP04 Proceedings. 2004.