Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project

Helder Ferreira[1] and Diamantino Freitas[1]

[1] LSS - Laboratory of Signals and Systems
Faculdade de Engenharia da Universidade do Porto, Portugal
{hfilipe, dfreitas}@fe.up.pt
http://lpf-esi.fe.up.pt

Abstract. "*How can a blind person surpass the difficulty in reading an on-line document's mathematical expressions? Why wasn't this completely solved yet? Is it not necessary? Is not easy?*" – These questions are only the top of the iceberg of a big problem with accessibility in the Internet. This concerns technical, scientific or even simple documents presented on-line that involve mathematical expressions. Addressing these issues the authors developed the *AudioMath* [1] project at LSS. It can be connected to a text-to-speech engine (TTS), providing speech rendering of the W3C's MathML [2][3] coded mathematical expressions. The paper intends to present the project methodology as well as the results already obtained. With *AudioMath* we intend to increase the accessibility of, not only, e-learning websites that use MathML, but also general websites. Therefore, *AudioMath* is an accessibility tool that can bring great benefits for visual impaired persons, but not only.

1 Introduction: Understanding the need and the difficulties

The publication and distribution of scientific papers and articles, as well as e-learning websites, are a desirable way to spread knowledge, promote education and stimulate research and development. The language chosen for these communications is often the **Mathematical Language**, because it is universal and not ambiguous.

However, the mathematical language can be split into 2 parts: the *meaning* and the *notation* [4], i.e., the mathematical concept exists no matter the representation it uses. Nevertheless, a good graphical representation of a mathematical expression leads to a better and fast interpretation of its meaning. Most of the web documents containing mathematical material use 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 common. This has lead to accessibility problems in the reading of documents containing mathematical expressions for visually impaired persons, and, in particular, for blind persons.

It is also consistent with the idea that the difficulty for blind persons rises and accessibility diminishes as the technical level of documents increase, even if they are already in digital format and using properly equipped computers [5]. The growing concern about accessibility of mathematical expressions on the Internet has taken research groups such as: W3C, OpenMath [6] and AMS [7], 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 Mark-up Language** [3] from W3C, which is used in this project.

2 The AudioMath Project

The practical interpretation and reading of mathematical formulae is not a simple task. Several challenges have to be overcome, such as:

- Lack of existing knowledge, as well as studies and research on how to read a mathematical expression [8][9][10].
- Weak existing implementations in current browsers in support of mark-up languages for mathematical expressions.
- Rather complex parsing and interpretation of mathematical formulae.
- Automatic identification of lexical and non-lexical elements in a text, needing a conversion into plain text. Use of glyphs, characters and fonts from other culture or language in a mathematical expression.
- Hhigh computational performance needed from the conversion module when associated to a TTS engine, in order to provide fast read-out of expressions without too large time lag in the speech output.
- The need of mechanisms to navigate inside mathematical formulae, allowing repetitions or reading of its internal contents in a personalized way.

The main aim of this project is to provide a tool, to work either standalone or integrated with a TTS engine, that converts expressions presently not “understandable” by a regular TTS engine, both in text and in mark-up elements (MathML). The tool is under continuous development and does the parsing, interpretation and conversion into plaintext form and readable by anyone, it is called **AudioMath**. It speaks the mathematical expressions supplied in the *MathML Presentation Mark-up* format.

Applications are many, such as: -teaching or learning how to read mathematical formulae [11]; -reading of technical and scientific documents online for everyone with particular benefit for vision impaired persons; -enhancing general accessibility to computer-based applications, when applied to a TTS engine.

2.1 MathML - Mathematical Mark-up Language

MathML is an XML application developed by W3C (versions on 1998 and 2001). This mark-up language has a simple and concise syntax that currently codes either the notation (*MathML Presentation Mark-up*) or the meaning (*MathML Content Mark-up*) of a mathematical expression. The **AudioMath Project** has chosen MathML for the following reasons:

- It’s a mark-up language developed by W3C (organization that issues recommendations about web technologies, that later on become standards through organizations like ISO). 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 platform independence.

Growing development of MathML (already in version 2.0), and the involvement of several relevant organizations associated with the teaching and learning of mathematical contents, such as the *American Mathematical Society* [15] and *The OpenMath Group* [16], and the involvement of software houses like *Design Science*, *Wolfram Research*, *Microsoft*, *IBM* and *HP*.

Emergence of editors and applications that create and manipulate MathML documents. For example: *Scientific Word*, *Mathematica* and *MathType*.

Existence of conversion tools between the main publishing formats: TeX/LateX to MathML and vice-versa.

2.2. MathML Presentation Mark-up vs. MathML Content Markup

The representation of mathematical formulae and other mathematical contents is perceived by two distinct concepts: the visual structure or notation of the mathematical expression, and the concept or meaning that it represents [4]. For example: the same concept of "division of a by b" can have the notations:

$$a/b\,,\ \frac{a}{b}\ \text{or}\ ab^{-1}. \qquad (\mathbf{1})$$

The inverse is also possible; there can be different meanings for the same notation. For instance, the notation "He" can have one of the following meanings: "product between variables H and e" or "chemical symbol of Helium". Therefore the W3C created two sets of MathML mark-up:

- MathML Presentation Mark-up: -Aimed to the visual presentation of a mathematical expression; -Doesn't differentiate the meaning of the mathematical expression; -Not intended to (but possible) audio rendering of mathematics; -The conversion of Presentation Mark-up into Content Mark-up is not advised.
- MathML Content Mark-up: -Aimed to represent the meaning of a mathematical expression; -Used to transfer MathML between applications; -Doesn't differentiate the mathematical notation; -Ideal to audio rendering of mathematical representations; -Can be converted into Presentation Mark-up; -Limited only by the mathematical operators and functions it supports.

Although Content Mark-up is recommended to be used in audio rendering of mathematical expressions [17] (since it preserves the meaning of the formulae) it has some limitations concerning the list of mathematical operators and functions it supports. The OpenMath Group, for example, offers a lot bigger dictionary. Moreover, current editors are WYSIWYG and code the mathematical contents into Presentation Mark-up instead of Content Mark-up. Therefore the online documents that contain MathML expressions are coded in Presentation Mark-up. For these reasons, AudioMath uses MathML Presentation Mark-up. The price to pay is that this introduces several difficulties to the interpretation of the formulae.

For instance, consider expression 2. In MathML Presentation Mark-up the indexes "2" and "3" will be coded as "subscript" and "superscript".

$$A_2^3. \tag{2}$$

This gives no information if the expression refers to the “cubic power of element A_2”, or “the permutations of 3 elements taken 2 at a time”. In conclusion, the use of MathML Presentation Mark-up requires a much bigger effort in the interpretation of mathematical expressions.

2.3. The AudioMath Process

2.3.1. Introduction

AudioMath has been built in a modular, extensible and configurable architecture, in Perl. Therefore, the support for new languages, the update of dictionaries and the updating of algorithms can be done easily. Currently only the European Portuguese language is supported.

AudioMath contains 6 major conversion modules: Numerals (conversion of several types of numeric forms), Abbreviations (conversion of abbreviations in a text), Acronyms (conversion of acronyms in the document), Network (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). The modules are packed together in the form of one ActiveX DLL.

2.3.2. Text Analysis

A technical document can probably come with special characters in Unicode and math glyphs. Therefore the first step is to clean up the text, converting Unicode to Latin1 (in this case to support European Portuguese characters). Next step is an auto-discovery process that recognizes certain types of elements and makes calls to the modules that convert them into a full text form. For example: if “det.” is detected, it should be converted into “determinant (of)”. To speed up the process the document should be divided into blocks of text, splitting the MathML Mark-up from the rest of the text.

2.3.3. Parsing MathML

The MathML Markup is parsed using the Perl module: XML::Parser [18] which acts as a SAX parser type (event-based), supporting Encoding ISO-8859-1 and discarding XML Namespaces.

2.3.4. MathML Interpretation and Conversion

Since AudioMath uses MathML Presentation Mark-up, a big effort and computation is needed in the interpretation of the mathematical expression. As the expression is becoming discovered, the conversion process takes place by calling several algorithms as well as Unicode and MathML dictionaries.

AudioMath uses two kinds of dictionaries: MathML entities to Unicode and Unicode to European Portuguese full plaintext form. For example: if the “⁢” MathML operator is used, then it is converted into “U+02062” Unicode element and then into the word “multiplies”. The conversion to text form is done according to a database of rules that was built based on a collection of material written by experienced professors. Note that the bigger the expression the more complex the interpretation and conversion process becomes. A solution would be to introduce navigational mechanisms to browse inside the expression.

2.3.5. Speaking Mathematical Contents

As already referred in this paper, one of the challenges this project faces is the scarcity of studies or research on how should we read a mathematical expression [8][9][10]. The *MathSpeak Project* [9] 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* [12], and later into Braille.

Another method comes from the work of *T.V. Raman* [10] the most impressive and recognized study in this area, 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. Nevertheless, this work points out several generally useful cues for audio rendering of mathematics.

The objective of automatically speaking mathematical contents has to deal with the non-trivial issues of text generation and phrasing as well as the generation of the prosody to superimpose over the synthetic speech.

Starting with the first one, the main problem is that there are several more or less “common” ways to read a mathematical expression. These “common” ways are used by speakers when the visual form is also present in any way or is known before-hand. We will illustrate the un-definiteness by an example. For instance, consider the simple mathematical expression 3.

$$\sqrt{a^2+b^2}\ . \qquad (\mathbf{3})$$

This could be rendered more or less ambiguously as:

Square root of a squared plus b squared, end of radicand.

Square root of a taken to power two, plus, b taken to power two, end of radicand.

Square root of power base a exponent two, end of power, plus power base b exponent two, end of radicand.

Which of these forms is more correct, not ambiguous and efficient, if any?

Now consider expressions 4 a, b and c. Taking in account the text forms presented before, how should we read these expressions?

One interesting experience is speaking the texts monotonically to persons that have no access to the graphical expression and ask for a written version after the dictation.

The solution of this problem must pass by the adoption of formal ways of text generation that keep the right structure information of the formula. On the other hand this trend considerably complicates the resulting text, by introducing a big overhead of structural words and therefore contributes to decrease the overall efficiency of the spoken communication.

$$\text{(a) } \sqrt{a^{2+b^2}}\text{, (b) } \sqrt{(a^{2+b})^2} \text{ and (c) } \sqrt{(a^2+b)^2}\ . \qquad (4)$$

How do things happen in spoken communication then? Are there specific cues introduced by the speaker in the form of prosodic marks intended to supply the listener boundary and structure information that the text itself does not possess? This topic was investigated and interesting issues resulted as described in the following.

Let's go back to the first example of text rendering of the first expression and consider the sample spoken version, whose waveform, f0 contour (intonation) and text labeling are depicted in figure 1, from top to bottom (the picture was obtained with the PRAAT [19] software).

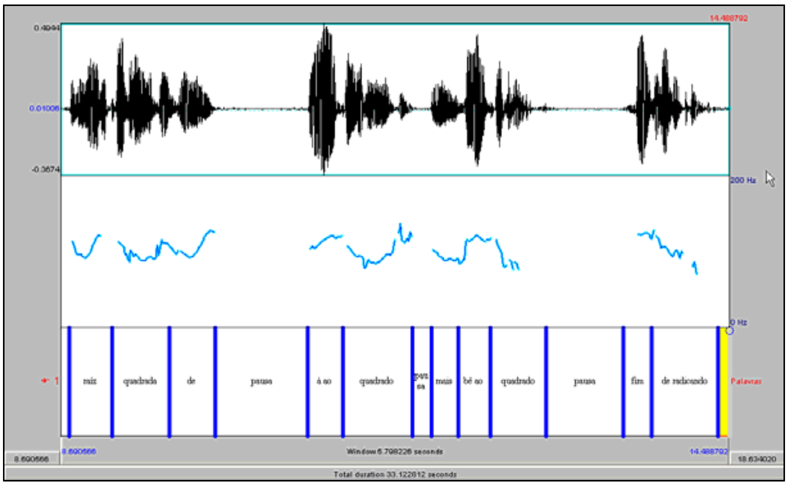

Fig. 1. Utterance "Square root of a square plus b square, end of radicand" spoken in European Portuguese.

As can be seen, there are two distinct pauses: one after "Square root of" and another before "End of Radicand". There is also a smaller optional pause between "á ao quadrado" (a squared) and "mais bê ao quadrado" (plus b squared).

Another immediately apparent aspect is the critical use of rising and falling movements of f0 by the speaker to provide classification of the separations introduced by the pauses. A rising tone is used when a lower hierarchical level is starting (see at the end of "quadrada de") and a falling tone is used when this level is ended (see at the end of "bê ao quadrado") while a rising tone associated to a down town are used to classify the smaller separating pause between identical elements, so indicating a continuation. Finally the emphatic falling f0 movement at "fim de radicando" (end of radicand) signals the end of the expression.

The rules already defined in the present development phase of the project are implemented at conversion time by tagging the text with prosodic marks, to command the TTS device in order to produce the required pauses and f0 modulations.

The **AudioMath Project** is continuously studying the prosody of speaking mathematical expressions and is currently working to deepen the knowledge of the use of those common prosodic delimiters and other characteristics inside more complex expressions and their varieties. Although the research is being carried out for European Portuguese, the authors believe that it can be substantially extrapolated to many other languages.

2.4 The AudioMath Prototype

AudioMath is capable of analysing, parsing, identifying and interpreting several types of mathematical expressions in MathML Presentation format. An especially important issue is the prosody control that is needed as input to the TTS in order for the speech output to be as intelligible and the comprehension as high as possible. Extensive studies of the prosody factors are continuously under way.

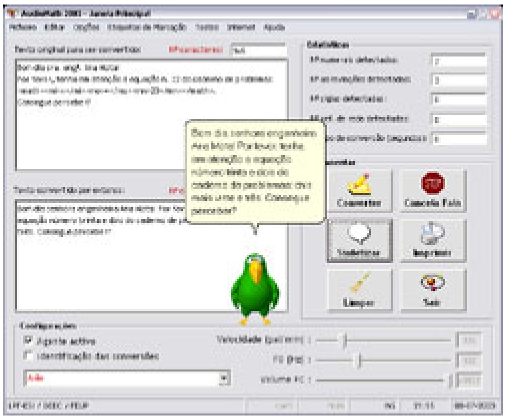

Fig. 2. Screenshot from the AudioMath Test and Development Application.

The existing prototype was connected to a European Portuguese TTS engine. However, since it was built in a modular and flexible structure, it is possible to add and change modules that allow conversion and speech output in other languages.

To develop and test the *AudioMath* features, an application has been developed. Through the easy interface (figure 2), text to be read from files or can be entered, or an Internet web page be opened for reading.

3 Conclusions and Future Trends of Work

Providing mathematical contents on-line, allows a greater access to information than before. However, problems do exist for blind persons in reading those contents.

AudioMath is, without doubt for the authors, a valuable contribution to the increase of accessibility in reading on-line documents with mathematical contents.

However it is a work in progress and plenty still needs to be done, such as: -Complete the support to *MathML Presentation Mark-up*; -Add support to *MathML Content Mark-up* [3]; -Further learning how to read mathematical formulae; -Develop modules that support HTML, XHTML, XML [13] and SSML [14]; -Provide mechanisms for navigating inside mathematical formulae; -Adding support for new languages; -Continue to develop the study on the prosody of reading mathematical formulae.

References

1. AudioMath for European Portuguese – http://lpf-esi.fe.up.pt/~audiomath .
2. W3C – World Wide Web Consortium – http://www.w3c.org .
3. MathML – Mathematical Mark-up Language – http://www.w3.org/Math .
4. Pierce, John. *An Introduction to Information Theory. Symbols, Signals and Noise.* Dover Publications Inc. New York. 1961.
5. Monaghan, Alex. Fitzpatrick, Donal. *Browsing Technical Documents: Document Modelling And User Interface Design*. BULAG 1999.
6. The OpenMath Group – http://www.openmath.org/coccon/openmath/index.html .
7. AMS – American Mathematical Society – http://www.ams.org .
8. Karshmer, Arthur. *How Well Can We Read Equations to Blind Mathematic Students?* HCII2003 Proceedings. Volume 4.
9. MathSpeak - http://www.rit.edu/~easi/easisem/talkmath.htm .
10. T.V. Raman. *Audio System For Technical Readings*. Dissertation to the Faculty of the Graduate School of Cornell University. 1994.
11. Freitas, Diamantino. Ferreira, Helder. et al. *A prototype application for helping to teach how to read numbers*. HCII2003 Proceedings. Volume I.
12. Resources to Learn Nemeth Code – http://www.tsbvi.edu/math/math-nemeth.htm
13. Mark-up languages from W3C used to publishing documents in Internet.
14. SSML - Speech Synthesis Mark-up Language – http://www.w3.org/TR/speech-synthesis .
15. AMS - American Mathematical Society - http://www.ams.org .
16. The OpenMath Group – http://www.openmath.org/coccon/openmath/index.html .
17. Sandhu, Pavi. *The MathML Handbook.* Charles River Media. 2002.
18. XML::Parser - http://search.cpan.org/dist/XML-Parser/Parser.pm .
19. PRAAT: Doing Phonetics by computer, http://www.fon.hum.uva.nl/praat/.