

# Access to Mathematical Expressions in MathML for the Blind

*Martin Rotard, Klaus Bosse, Waltraud Schweikhardt and Thomas Ertl*

Visualization and Interactive Systems Group  
University of Stuttgart, Germany

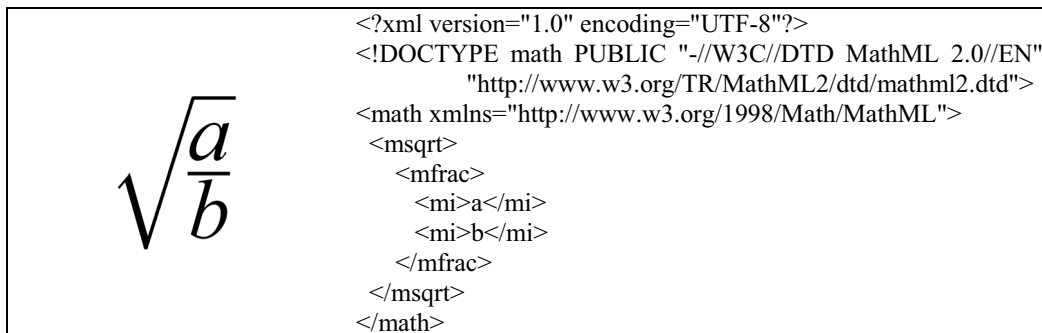
{rotard, schweikh, ertl}@informatik.uni-stuttgart.de, klaus.bosse@web.de

## Abstract

For visually handicapped people virtual learning environments are a chance to participate in scientific education, because they can access the written content. However mathematical expressions rendered as raster graphics are not accessible to them. MathML, the web-standard for mathematical notation, provides the possibility to describe mathematical expressions in a textual form. However, it is not comfortable for this purpose because of its length and complexity. In this paper we present a transformation from MathML expressions into a compact mathematical notation for the blind (SMFB - Stuttgart Mathematical Notation For the Blind). Finally, we show how we automate this transformation by integrating it into a virtual learning environment.

## 1 Introduction

More and more documents become available in virtual learning environments. This increases the chances for blind people to get access to textual content of a course. However, it is common that mathematical notations are integrated into the content as bitmap graphics. Therefore, they are not available to blind people. This paper presents an approach on how to transform the content of a course with integrated MathML [MathML] into a mathematical notation for the blind and how to automate this transformation.



**Figure 1:** A mathematical expression and the related source code in MathML

## 2 Mathematical Expressions in MathML

MathML (Mathematical Markup Language) is a W3C standard (World Wide Web Consortium) [W3C] for mathematical notations. It allows to integrate mathematical expressions into a web document and to use it as an interchange format for mathematical software. As an example the

source code of the square root of a fraction is presented in figure 1. It can be displayed natively by a graphical web browser [Mozilla] or by using a browser plugin.

The source code of mathematical expressions in MathML is legible for blind people due to its linear notation. However it is not comfortable for this purpose because of its length and complexity. Additionally, the limitation of braille displays to a maximum of 80 characters makes it absolutely necessary to transform mathematical expressions from MathML into a compact mathematical notation for blind people.

### 3 Transformation from MathML into SMFB

For this transformation we use an XSLT stylesheet (Extensible Stylesheet Language Transformations) [XSLT]. A stylesheet can process valid XML documents like XHTML documents (Extensible HyperText Markup Language) [XHTML]. As an example we choose SMFB (Stuttgart Mathematical Notation For the Blind) as a target mathematical notation for the blind [Schweikhardt 1980].

#### 3.1 SMFB

SMFB has been developed in our group at the University of Stuttgart since 1980. It is based on the Marburg Notation [Scheid et.al. 1930] and is encoded in 8-dot-Braille. It has been always our goal to promote the integration of blind children, students and adults into regular learning and working environments. Since 1999, SMFB expressions can be transformed into traditional mathematical notation (figure 1 - left) [Schweikhardt 2000]. The number of characters in a mathematical expression in SMFB matches the one of the representation in the corresponding version for sighted readers (figure 2). This made it possible to develop a working environment to use Microsoft Word to write and print mathematical papers [Schweikhardt 1998].



Figure 2: 8-dot-Braille and corresponding visual representation of a square root of a fraction

#### 3.2 Transformation

During the transformation with our XSLT stylesheet [Bosse 2002] the abstract notational structure of a mathematical expression is translated from MathML into a combination of SMFB and XHTML. The characters used for SMFB are represented in the Unicode character set [Unicode].

In the transformation process every MathML element is replaced by a suitable XHTML element. In most cases an element without special meaning is sufficient. Such elements are `<span>` or `<div>` which are allowed in every position where a `<math>` element can occur. They are the representatives of the inline and block-level elements, which divide up all XHTML elements. In XHTML it is not allowed to use block-level elements subordinate in inline elements. Especially an XHTML `<table>` that we want to use as a substitute for a MathML `<mtable>` is only allowed in block-level elements and forbidden in inline elements. Thus, there has to be a test, which indicates the type of the parent element of the initiatory `<math>` element. If this type is 'inline' the contained MathML elements have to be all replaced by inline elements – in particular `<mtable>` must not be replaced by `<table>`, but has to be translated into an SMFB-table. This procedure is necessary to get a valid (DTD-conforming) XHTML-document at the end of the transformation.

There is a template for the transformation of every element of the MathML presentation markup in the XSLT stylesheet (Figure 3). In these templates the XHTML elements for replacement, the order of SMFB special characters and the processing of child elements are chosen. The template for the initiatory  $\langle\text{math}\rangle$  element additionally contains the query for the type of its parent XHTML element. This is given by a parameter ('IsSMfBInlineMode') to all applied templates.

```

<xsl:template match="mathalias:msqrt">
  <xsl:param name="IsSMfBInlineMode" />
  <xsl:element name="{ $stdtagsdict[@isinline=string($IsSMfBInlineMode)]/@tag}"
    namespace="http://www.w3.org/1999/xhtml">
    <!-- comment: chooses element name 'span' or 'div' from a dictionary -->
    <xsl:value-of select="$BeginRadicalBase" /> <!-- comment: BeginRadicalBase=U+221A -->
    <xsl:apply-templates>
      <xsl:with-param name="IsSMfBInlineMode" select="$IsSMfBInlineMode"/>
    </xsl:apply-templates>
    <xsl:value-of select="$EndRadicalBase" /> <!-- comment: EndRadicalBase=U+301B -->
  </xsl:element>
</xsl:template>

```

**Figure 3:** XSLT template for a MathML square root (msqrt) element

In addition, a transformation takes place on the character-level where some Unicode characters, which are not part of the limited SMFB character set, are replaced by special combination of SMFB characters. This is realized by a dictionary, which is placed in an XSLT variable and a template for all MathML text nodes which uses this dictionary.

The result of the transformation is a document combining SMFB and XHTML. A screen reader has to be able to read XHTML, recognize the contained Unicode character set and convert SMFB characters into their braille counterpart. There is a possibility to transform MathML directly into SMFB without the detour to XHTML markup. But our indirect way preserves the XML structure and therefore minimizes the loss of information. Additionally the navigation support of the screen reader for XHTML tables can be used in matrices.

A mathematical expression can be defined in MathML either in presentation or content markup. Elements of the presentation markup describe the abstract notational structure used for rendering. The content markup encodes the mathematical structure, i.e. the meaning of an expression. Our XSLT stylesheet transforms mathematical expressions of the presentation markup. However, there are other stylesheets (e.g. proposed by [ORCCA]), which transform content into presentation markup. By combining both stylesheets mathematical expressions belonging to the content markup can be transformed, too.

## 4 Automatic Transformations

XSLT stylesheets are XML documents, therefore they are not directly executable. To process these stylesheets it is necessary to use an XSLT processor like Xalan, Saxon, etc. [Xalan, Saxon]. As a consequence the users have to transform the web pages manually in the command line. This manual and time-consuming transformation can be made much easier in the following cases:

- Future versions of screen readers will be able to integrate XSLT stylesheets. Then, mathematical expressions in MathML will be processed directly. The mathematical output notation will depend on the configuration of the screen reader.
- In future versions, it may be possible to integrate XSLT stylesheets into web browsers or plugins for web browsers. Then, it will be possible to configure the processing of mathematical expressions in this software.
- XSLT stylesheets can be integrated into virtual learning environments residing on the server. XHTML documents can then be generated easily with different mathematical notations for the blind. The first time a blind user logs on the environment the mathematical output notation for mathematical expressions can be specified. We implemented this approach in a prototype, which is described in the following paragraph.

## 5 Presentation Engine

In the project ITO (Information Technology online) [ITO] that is initiated by the German Federal Ministry of Education and Research to exchange lecture materials, we developed a presentation engine for a virtual learning environment. We used Apache Cocoon and JetSpeed for the Tomcat web server [Cocoon, JetSpeed, Tomcat]. As a repository for the components of a course we defined an XML-intermediate format [Finsterle et al. 2003]. An XSLT stylesheet is used to transform the XML content into XHTML in our presentation engine. Our presentation engine can display documents in XML, XHTML, HTML, etc. Additionally we integrated another XSLT stylesheet, which presents the XML-content in a special version for the blind. In the XHTML output the paragraphs are labeled with the paragraph type instead of using special style. E.g. a definition paragraph is labeled with “Definition:” instead of a surrounding box.

$$\sqrt{\frac{a}{b}}$$

**Figure 4:** Representation of a square root of a fraction in LaTeX

For the processing of documents Cocoon supports a pipeline concept. This makes it possible to process more than one XSLT stylesheet in sequence. As a result the users can configure their preferred mathematical output notation. For the transformation of mathematical expressions we integrated two XSLT stylesheets in this engine: Our stylesheet for the transformation of mathematical expressions from MathML into SMFB and a stylesheet to transform from MathML into LaTeX (figure 4) [Yaroshevich], since LaTeX becomes more and more important as an alternative mathematical notation for the blind.

## 6 Concluding Remarks

In this paper we have shown how to transform expressions from MathML into SMFB and how to automate this transformation by integrating it in a virtual learning environment. The main aspect of our paper is not limited to SMFB. Part of the functionality of our stylesheet can be easily used in other stylesheets for transforming MathML into other mathematical notations for the blind.

In the future, it will be necessary that blind users can configure the transformation for the mathematical notation they are familiar with in all applications they are working with. To learn a new mathematical notation is an additional effort for them. MathML is a good basis to support more than one mathematical notation for the blind in virtual learning environments.

## Acknowledgement

*This work was funded by the project ITO of the German Federal Ministry of Education and Research. We would like to thank J. Diepstraten, L. Finsterle, B. Josefy, M. Ressel, A. Hub and M. A. Schaal for the help to finish this paper and for the fruitful discussions.*

## References

- [Bosse 2002] Bosse, K.: *Transformation von mathematischen Dokumenten, die Teile in MathML enthalten, in eine in Stuttgarter Mathematikschrift für Blinde (SMFB) formulierte Darstellung*, Diplomarbeit, Insitute for Computer Science, University of Stuttgart, 2002
- [Cocoon] The Apache XML Project: *Cocoon*, <http://xml.apache.org/cocoon/>, 20.01.2003
- [Finsterle et al. 2003] Finsterle, L.; Rotard, M.: *Mit konventionellen Autorensystemen zum E-Learning Portal*. In: K.P. Jantke, W.S. Wittig, J. Herrmann (ed.), *Tagungsband 10. Leipziger Informatik Tage*, pp. 111-121, Akademische Verlagsgesellschaft Aka GmbH, 2002
- [ITO] ITO Projekt: *ITO – Information Technology Online*, <http://www.ias.uni-stuttgart.de/ito/>, 20.01.2003
- [JetSpeed] The Jakarta Project: *JetSpeed*, [jakarta.apache.org/jetspeed/](http://jakarta.apache.org/jetspeed/), 20.01.2003
- [Mozilla] The Mozilla Organization: *Mozilla.org*, <http://www.mozilla.org/>, 20.01.2003
- [MathML] W3C: *MathML - Math Home*, <http://www.w3.org/Math/>, 21.01.2003
- [ORCCA] ORCCA: *MathML Content to MathML Presentation*, <http://www.orcca.on.ca/MathML/software.html>, 22.01.2003
- [Saxon] Kay, M. H.: *Saxon -the XSLT Processor*, <http://saxon.sourceforge.net/>, 20.01.2003
- [Scheid et.al. 1930] Scheid, F. M.; Windau, W.; Zehme; G., *System der Mathematik- und Chemieschrift für Blinde*, Deutsche Blindenstudienanstalt, Marburg a.d. Lahn, 1930.
- [Schweikhardt 1980] Schweikhardt, W.: *A Computer Based Education System for the Blind*, in: Lavington, S. H. (ed.), *Information Processing 80*, pp. 951-954, North Holland Publishing Company, 1980
- [Schweikhardt 1998] Schweikhardt, W.: *8-Dot-Braille for Writing, Reading and Printing Texts which Include Mathematical Characters*, In: A.D.N. Edwards, A. Arato, W.L. Zagler: *Computers and Assistive Technology, ICCHP98, Proceedings of the XV. IFIP World Computer Congress*, 1998
- [Schweikhardt 2000] Schweikhardt, W.: *Requirements on a Mathematical Notation for the Blind*. In: R. Vollmar, R. Wagner (eds), *Computers Helping Peoples with Special Needs ICCHP 2000*, pp. 663-670, July, 17-21, 2000
- [Tomcat] The Jakarta Project: *Tomcat*, <http://jakarta.apache.org/tomcat/>, 20.01.2003
- [Unicode] The Unicode Consortium: *Unicode Home Page*, <http://www.unicode.org/>, 21.01.2003
- [W3C] W3C: *World Wide Web Consortium*, <http://www.w3.org/>, 21.01.2003
- [Xalan] The Apache XML Project: *Xalan-Java*, <http://xml.apache.org/xalan-j/>, 20.01.2003
- [XHTML] W3C: *The Extensible HyperText Markup Language*, <http://www.w3.org/TR/xhtml1/>; 20.01.2003
- [XSLT] W3C: *XSL Transformations*, <http://www.w3.org/TR/xslt>, 21.01.2003
- [Yaroshevich] Yaroshevich, V.: *XSLT MathML Library*, <http://www.raleigh.ru/MathML/mmltex/>, 21.01.2003