DELIVERABLE

Project Acronym:	**EuDML**
Grant Agreement number:	**250503**
Project Title:	**The European Digital Mathematics Library**

D10.1: State of the Art of Accessibility Tools

Revision: 1.0 as of 28th February 2011

Authors:

Volker Sorge	University of Birmingham
Mark Lee	University of Birmingham
Petr Sojka	Masaryk University
Alan P. Sexton	University of Birmingham

Contributors:

Martin Jarmar	Masaryk University

Project co-funded by the European Comission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	✓
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
0.1	15th October 2010	Volker Sorge	UB	Just a content outline.
0.2	28th January 2011	Volker Sorge	UB	Section 2 completed.
0.3	4th February 2011	Petr Sojka	MU	MU experience part added.
0.4	13th Feburary 2011	Volker Sorge	UB	Sections 3,4,5 completed.
0.5	14th February 2011	VS+PS	UB+MU	Development version.
0.6	14th February 2011	Alan P. Sexton	UB	Typo corrections and polishing
0.7	19th February 2011	Mark Lee	UB	Addition of conclusions on language translation
1.0	28th Feburary 2011	Volker Sorge	UB	Made changes suggested by reviewer and final editing.

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Abstract
The purpose of this report is to present a state of the art in accessibility tools that can be used to provide access to mathematical literature to visually impaired users as well as print impaired users (i.e., people with specific learning disabilities like dyslexia) as well as an overview of current automated translation technology. We draw some conclusions for the EuDML project by suggesting what support EuDML can provide for accessibility and translation as well as sketch a road map to the first prototype implementation of the accessibility tool set.

Contents

1 Introduction

The goal of work package 10 is to enhance EuDML by providing a set of tools that enable accessibility to mathematical documents for special needs user groups, in particular, visual impaired users but also users with specific learning difficulties such as dyslexia, dyscalculia, etc. We will refer to these learning difficulties generally by the common concept of print impairments. Furthermore, in particular for multilingual collections the adequate automatic or semi-automatic translation of content, especially titles, abstracts and keywords is desirable.

The tool set aims to provide some added value to EuDML that can not be found in other digital libraries or repositories for scientific literature. This will make the EuDML portal not only attractive for users with visual or print impairments but can also cater for a wider audience by providing tools that allow accessibility of contents by alternative means that can complement the traditional paper or electronic version and could prove to be useful and convenient.

The purpose of this report is to evaluate existing software tools and techniques that can provide accessibility to documents in general and scientific literature in particular. We will therefore focus on the following topics:

Speech Synthesis is the ability to automatically transform written text into speech output to have documents read out. Within EuDML we are particularly interested in tools that not only allow speech synthesis for plain text but also for mathematical text speech as well as enabling multilingual access to documents.

Mathematical Accessibility Formats and Tools allow access to mathematical literature by readers with special needs. We will examine standards and reformatting tools for mathematical formulae or graphics that enable access for visually and print impaired readers such as via Braille, large print or the DAISY format.

Language Translation is a critical issue in EuDML given our concept to serve as a portal for different national content providers that have collections of mathematical literature in a variety of different languages. As a consequence, tools that enable the translation between languages will not only provide additional accessibility for users but also support search and cross referencing facilities within EuDML itself.

1.1 Relationship to EuDML as a whole

Due to its explicit aim to provide added value to EuDML, work package 10 is an add-on and not a prerequisite for any of the other work packages within the project. However, its success depends on a more basic form of accessibility, namely the ability to obtain access to the actual text and formulae of mathematical documents. This is especially crucial for documents that only exist in retro-digitised form or where the sources for a digitally born document are not available. In other words, one prerequisite for the more advanced accessibility techniques we would like to offer are the basic retrieval of content by techniques like optical character recognition, formula recognition etc. These technologies overlap with those employed in work package 7 and they have been discussed in detail in the state of the art report of deliverable D7.1. We therefore omit a repetition of their discussion in this report and rather refer to D7.1 [28] for details.

1.2 Structure of the Report

The structure of the report follows closely the three main topics of the state of the art: In section 2 we first review assistive technology tools by presenting the state of the art for general screen readers, special purpose speech synthesis tools and general reading aids which aim to support printing impaired readers. Section 3 gives an overview of different formats that enable accessibility and tools to reformat documents. Section 4 concentrates on technology for multi-lingual translation, giving the different perspectives on automatic and computer-assisted translation methods as well as online and offline tools. We conclude in section 6 with a discussion of possible tools that could be integrated into EuDML and a short road map towards the first prototype.

2 Assistive Technology Tools

We first survey some of the assistive technologies that are currently available in order to support users with special needs, such as visual or print impairments. As there exists a plethora of tools, making a comprehensive survey infeasible, we focus on those tools that are directly relevant to the aims of the EuDML accessibility support. These are twofold:

Our primary goal is to make documents accessible in the sense that they can be easily exploited by generic assistive technologies such as screen readers or accessibility software but also so they can be easily found by offering adequate search aids and input correction for print impaired users. A secondary goal is to provide support for particular accessibility tools that can be downloaded freely and installed by users of EuDML in order to exploit advanced accessibility features. We therefore focus on a survey of generic screen readers, bespoke speech synthesis tools and more general electronic reading aids.

2.1 Screen readers

A screen reader is a form of assistive technology that aim to identify and interpret what is displayed on a computer screen in order to reformat it to an output format suitable for blind or visually impaired users. Screen readers can produce a variety of output formats, e.g. translating into sound using text-to-speech engines, or into formats suitable for Braille output devices such as refreshable Braille displays or embossers. There are a variety of different types of screen readers (for a relatively comprehensive list see `http://en.wikipedia.org/wiki/Comparison_of_screen_readers`). For our purpose we distinguish three different types of screen readers:

1. tools working at the desktop level
2. tools embedded into specialised document readers, and
3. readers specialising in web access.

2.1.1 Desktop Readers

The purpose of screen readers at the desktop level is both to interpret control features, such as cursor position, desktop icons, menu items, and content displayed in different applications. We will examine some of the most common screen readers for different platforms with an emphasis on their usability for multilingual and mathematical documents.

Jaws is a commercial screen reader developed by Freedom Scientific for the Windows operating system (`http://www.freedomscientific.com`). It is one of the most widely used accessibility software tools providing support for many commonly used desktop applications. It has specific support for text to speech translation of PDF documents, in particular it can work with tagged PDF. This feature can especially be exploited in combination with PDF viewers like Acrobat Reader. Jaws can produce both speech output as well as output for Braille readers. In addition it has multi-language support in that one can install multiple voice languages as additional modules.

Jaws provides a proprietary scripting language that allows software developers to fine-tune their applications for the screen reader. While this allows for very precise accessibility for particular applications, it has the drawback that Jaws output for plug-ins in programs is often less effective then that in dedicated software. For example, screen reading a PDF file within a web browser using a PDF viewer plugin can lead to inferior results than when viewing the same file with the same viewer in stand-alone mode. This is because the Jaws scripts invoked from within the web browser are optimised for web browsing and not for PDF reading in a plugin.

While there have been some studies by Freedom Scientific on support for mathematics in their accessibility software suite — notably, attempts were made to integrate support for Nemeth Code in 2005 under U.S. Department of Education Grant no. H327A020024 — there is currently no special support for mathematical notation or scientific symbols in Jaws. However, Jaws can work to some extend with MathML.

Non Visual Desktop Access (NVDA) (`http://www.nvda-project.org/`) is an open source project with the aim to provide a freely available screen reader for the Windows operating system. NVDA uses Python as a script manager that can be employed to adapt NVDA for particular applications. It already has support for a number of common desktop applications, and, while it has very limited native support for PDF, it does have explicit support for Acrobat PDF Reader, and can read highlighted text. NVDA uses the open source text to speech engine eSpeak (see 2.2) giving NVDA multilingual support for 20 languages. In addition NVDA provides support for Braille output.

NVDA has no specific support for mathematical text. However, there are several ongoing projects that aim to enhance NVDA in the future. Of particular interest for EuDML are the current projects that aim to provide Mathematics accessibility and Diagram accessibility.

Orca (`http://live.gnome.org/Orca`) is a freely available, open source screen reader primarily developed for Linux with support from Sun Microsystems for Solaris platforms. It supports output in both speech and Braille as well as magnification support for screens. Orca provides access to graphical desktops primarily based on Gnome (there is very limited support for KDE and even less for older windows managers) as well as to compliant applications. Orca can be extended and adapted to applications using Python scripts.

Orca uses Gnome Speech as an interface and can consequently interface to a variety of speech engine supported by Gnome Speech. In particular, it can exploit speech engines like eSpeak and Festival (see Sec 2.2) and their multi language support.

While there is a project to implement Python scripts to adapt Acrobat Reader for Orca there is currently no special support for PDF or any other PDF readers. Likewise there is no particular support for mathematical notation or MathML, SVG, etc. in Orca.

VoiceOver is a screen reader integrated with the operating system of the Apple MacIntosh (`http://www.apple.com/accessibility/voiceover/`). It provides multilingual speech support, magnification as well as translation into Braille of several languages and displays. As VoiceOver is developed by Apple, most software available for MacIntosh has adequate accessibility support. While VoiceOver provides a Java API for developers it does not have easy scripting support for previously developed applications.

VoiceOver can read PDF via a highlighting or a page by page mode. In particular, the screen reader can pick up alternative text embedded into tagged PDF documents, e.g., natively or after OCR of the original document. VoiceOver has limited support for mathematical notation and no support for MathML or SVG.

2.1.2 Document Viewers

As EuDML is primarily interested in providing accessibility to mathematical documents we briefly review some dedicated document readers with respect to their accessibility features.

Adobe Acrobat Reader The vast majority of our articles will be provided in some version of Adobe's Portable Document Format (PDF). Consequently accessibility to PDF documents will be of great interest for EuDML. There are a number of PDF readers available for all major platforms, some of them provide accessibility support primarily in the form of magnification. However, to our knowledge the only viewer with direct speech support is Adobe's Acrobat Reader that has a built-in *read-out-loud* function. Acrobat Reader provides this functionality in all three major operating systems, Windows, Mac, and Linux, however, not for other flavours of Unix. In addition, under the Windows operating system Acrobat Reader can cooperate with Jaws for advanced support such as different voices etc. Similar integration is not provided for other screen readers or operating systems.

Read-out-loud is primarily tuned to work with regular text files, in particular, using a line by line reading approach. As a consequence, it is not suitable for non-standard text items such as tables, which can be difficult to comprehend. For mathematical expressions this approach leads to even less desirable interpretations, e.g., when sub- or superscripts are interpreted as separate lines and read out in a disconnected fashion. *Read-out-loud* does not distinguish different fonts, for instance reading both R and $\mathcal{R}$ simply as 'R'. However, these subtle distinctions can be crucial for the understanding of mathematical expressions. Other mathematical symbols are interpreted incorrectly, e.g. $\neq$ is read as 'equal' or can not be interpreted at all. While most of the time this means that symbols are simply interpreted and often read out as blank, sometimes mathematical symbols can interrupt *read-out-load* altogether. For instance, when asked to read an entire page, *read-out-loud* can fail to continue when reaching a complex mathematical expression.

Read-out-loud can also not deal with any graphics or image files embedded into PDF. However, the advantage of *read-out-loud* is that it can work with tagged PDF files. In particular, it can pick up alternative text provided in tags. This can be useful when mathematical expressions, tables, diagrams, etc. are tagged with the appropriate expressions that should be read out.

Emacspeak [25] aims to provide a fully accessible working environment based on Emacs alone. Emacspeak is a free, open source software that is available for all operating systems to which Emacs has been ported but has best support for Unix flavours including MacOSX. Emacspeak can work with a variety of speech engines, such as eSpeak, and take advantage of their features such as multiple voices and multilingual support.

The goal of Emacspeak is to provide a comprehensive computing environment that is fully accessible for the blind without the need for a graphical desktop or windows manager. As a consequence it supports tools such as programming environments, web browser, document creation, calculator etc. in Emacs. This also means that Emacspeak supports ASCII documents, only, and can not deal with PDF documents, diagrams etc. Emacspeak does not support MathML. However, Emacspeak has excellent support for LaTeX documents and in particular, using the Aster [26] audio system, it can read out mathematical formulae written in standard LaTeX correctly.

2.1.3 Web Browsing

Some technology has been developed that aims exclusively to provide accessibility support for web browsing. In the context of EuDML we examine some of the available tools with respect to their ability to deal with mathematics markup such as MathML or LaTeX. First, however, we briefly review a standard for accessibility of web pages.

WAI-ARIA Accessible Rich Internet Applications Suite is a standard proposed by W3C's web accessibility initiative with the goal of supporting accessibility of web pages for disabled users (cf. `http://www.w3.org/WAI/intro/aria.php`). It not only focuses on support for screen readers but also on providing accessibility to features of web pages for users who can not use graphical input devices like mice. It especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.

BrowseAloud is a browser companion that reads out the words on a web site, highlighting the words as they are being read aloud. It is designed for Windows and MacOS operating systems, but can be made to run in Linux as well. BrowseAloud aims to be a tool for people with literacy problems and/or learning disabilities such as dyslexia. It is not necessarily useful for people with visual impairments or blind people. It is particular effective on web pages designed with respect to the W3C accessibility guidelines. BrowseAloud can deal with MathML via plug-ins, like MathPlayer (see section 3.3).

Spoken Web is a browser plug-in for the Internet Explorer running exclusively under Windows. It adds text to speech functionality to Internet Explorer and employs MathPlayer as a plug-in as well.

Fire Vox is a freely available, open source extension for the Firefox browser that runs on all major platforms. It not only gives a speech extension for regular text but also has support for MathML directly built-in.

WebAnywhere is a web-based screen reader being developed by the WebInSight group at the University of Washington (`http://webanywhere.cs.washington.edu/`). It requires no special software to be installed on the client machine and can be used on any computer and system that has a sound card. To date there is no support for MathML.

2.2 Speech Synthesis Tools

Speech synthesis is the production of speech artificially by computer. It could be a production of speech using prerecorded speech segments or speech reproduction based on information extracted from the original speech signal or source by mimicking a human speech articulation model. Many of the screen readers and related tools we have surveyed in the previous section contain a speech synthesis module or make use of a third party speech synthesis tool.

Apart from these more generic tools there are a number of dedicated speech synthesis tools under development, primarily in academia, that aim at providing experimental and advanced features, such as modelling of emotions, multilingual text, etc. These tools generally function as text-to-speech engines, abbreviated TTS, whereby input text can be raw text, as in general reading text or tagged text, which may already be annotated with hints for the speech synthesis process, such as phonetic and/or prosodic estimation values. TTS engines transform regular input text into speech by a sequence of processing steps, which, roughly outlined, proceed as follows:

Preprocessing normalises input text generally by (1) tokenising input around either word units or sentence units, (2) removing unwanted characters that do not contribute to the spelling of a word, such as punctuation or end of line characters, (3) storing separated units into appropriate data structures for the latter steps of the synthesis.

Morphological Analysis determines the linguistic units contained within a word. For example, entities such as compound words and website URLs contain a number of different lexical units, which are extracted, examined and processed separately.

Syntactic Analysis establishes the part of speech which should be assigned to each word or linguistic unit.

Grapheme-to-phonemes transcription has the task of assigning a phonetic transcription to each word by mapping sequences of graphemes — the fundamental units of written language — to sequences of phonemes — the fundamental units of spoken language. A phonetic transcription can either be applied by using a dictionary look-up process, or by use of a grapheme-to-phoneme conversion unit. Dictionary look-up requires a list of words that are to be synthesised in the system, along with their phonetic transcriptions. This is sufficient for a synthesiser within a fixed domain. However, for use in an unlimited domain, where words such as names and proper nouns are included, a transcription dictionary will not suffice. This is where grapheme-to-phoneme rules are applied to calculate the transcription of a given word.

Prosody is the final step of text-to-speech synthesis and adds rhythm, stress, and intonation for creating the actual acoustic waveform for the final output.

Within the context of the EuDML project we are particularly interested in how one could employ existing TTS tools in order to enhance speech generation for mathematical documents. This could be achieved by supplying specialist support for some of the above steps of the TTS process. Clearly it is outside the remit of this project to actually work on a dedicated TTS procedure for mathematics. However, we envision a scenario in which EuDML could provide support for an existing TTS tool, e.g., by providing a specialist preprocessing of mathematical expressions or adding a dedicated dictionary for grapheme-to-phonemes transcription. A user could then install the supported tool together with the library offered by us in order to get greater support in accessing documents.

Consequently, we will survey TTS tools with respect to ease of modification for our purposes as well as their support for multilingual speech synthesis, which has the goal of creating synthetic speech for multiple languages, as this will be of particular benefit for EuDML given the multi-lingual aspect of our collections.

MBROLA is a freely Available Multilingual Speech Synthesizer from the TCTS labs at the University of Monsi [12]. MBROLA, which stands for Multi-band Re-synthesis Overlap Add, is a signal modification algorithm used to modify the prosody of the speech which originated in PSOLA (Pitch Synchronous Overlap and Add). MBROLA uses a speech synthesis engine and a collection of diphone speech corpora of 72 speech corpora for 37 languages [17], which has been encrypted and formatted in such a way that it is only readable for the MBROLA synthesiser. The only work required to make the MBROLA fully run as a complete TTS system is by providing preprocessing of the text such as providing the conversion from grapheme-to-phoneme for the desired language, determine the template for prosody assignment and build the code from preprocessing up to the passing of parameters to the MBROLA synthesiser.

Festival Festival is a speech synthesis framework implemented in Lisp by the Centre for Speech Technology Research, University of Edinburgh [7, 9]. It provides complete text-to-speech libraries which can be use as in a plug and play manner for developers to create a complete concatenative synthesiser system. The libraries include preprocessing tasks, development support for diphone speech databases and a platform to allow communication between the corpus and the developed system. While the libraries are language independent, there are some prerequisite documents which need to be prepared before one can create a TTS system using Festival. These are, for example, speech recording, annotation of diphones, prosody alignment and assignment, etc. The flexibility of the Festival framework enables a developer to customise the TTS process at every major step and to provide new bespoke modules if desired. In particular it is also possible to use different synthesisers such as MBROLA. Although Festival is still an ongoing research project it is fairly mature for American and British English and also works for Welsh and Spanish. Furthermore, there exist a number of languages provided by third-party projects for Festival.

FestVox is a project that tries to support the systematic development of synthetic voices to enable the easy construction of new voices by non-specialists [6]. FestVox is part of the Sphinx project for speech recognition at the Carnegie Mellon University. It aims to be tool within the Festival environment and provides specific scripts to build new voices in different languages as well as databases of basic utterances that can be composed for novel voices.

Demosthenes is the name of at least two separate speech synthesisers. The first system of this name is a speech synthesiser for the Czech language developed at the Faculty of Informatics, Masaryk University. It achieves particularly good results when fast reading is needed due to its prosody model being based on a composed syllable segments approach [23]. The idea of composed syllable segments is to reduce the number of elementary segments that have to be extracted from samples to create the speech segment database. Experiments show that the quality of the synthesised speech that is obtained by using composed syllable segments is very good even in comparison to the speech synthesis that uses the full syllable segment database [20]. Demosthenes is no longer being actively developed.

A second system called Demosthenes is a multilingual TTS composer from the University of Athens [33]. It provides a highly flexible framework that can be further developed and customised for specific applications. Similarly to Festival, it can be used with different synthesisers like MBrola. One particular aim of Demosthenes is to provide TTS facilities for electronic text to exploit meta information, e.g. by accessing HTML tags, etc. [32] Demosthenes also provides support for multiple languages with a particular focus on multi-lingual documents. However, at the moment it only supports English and Greek. Nevertheless in the context of EuDML, having accurate pronunciation of Greek letters is highly desirable. But in its current form Demosthenes primarily supports interspersed natural language, e.g., mixed English and Greek text and has little support for single letters.

eSpeak is a compact open source software speech synthesiser, that is available for all major operating systems, including Windows, Linux, MacOS, and Solaris. It provides voices for a large number of different languages, including different variations of English, German, Polish, French, Greek. eSpeak can also be used as a front-end to the MBROLA algorithm and can make use of voices by this tool as well. eSpeak is available as a choice of synthesiser for many open source screen and document readers, such as Orca or Emacspeak.

FreeTTS is a TTS system developed in Java. It provides a Java library and API for the further development of TTS engines. It includes support for English voices but also has the capability of incorporating voices from other system such as Festival and FestVox. It also provides an interface for Emacspeak to enable the use of FreeTTS as an alternative synthesiser.

2.3 Reading Aids

Reading aids are more general tools to assist reading and text understanding for people with learning disabilities such as dyslexia or dyscalculia, but also people that can neither read nor write or those who are non-native speakers of the language they work in. All of the tools previously discussed can be understood as reading aids for this target group. However, more specialist tools aimed at this user group — especially commercial tools such as Read and Write from TextHelp or ClaroRead from Claro Software — are primarily integrated software solutions that include a number of tools for (a) text input, such as voice input, text prediction, spelling dictionaries and correction, (b) text output via speech synthesis or font separation for better legibility, and (c) paper text conversion with scanning and OCR software. While some tools also offer simple support for mathematics, such as talking calculators, there is generally little to no support for advanced formula input, output or OCR. As support for these advanced software packages is certainly beyond the scope of a digital library we have not researched them in detail. However, from personal communication with dyslexic mathematicians we have learned that input correction and text prediction with respect to a specialist mathematical vernacular is highly desirable but currently not offered by any of the available tools or search engines for existing electronic mathematical libraries.

Another common form of reading aid is using magnification of documents or of parts or all of the display. Software magnifiers come in several guises, depending on their usage, and we can distinguish the following:

Desktop Magnifiers are tools that can magnify anything displayed on a desktop, including most content within application programs. There are two approaches to magnification: Zoom magnification that magnifies the entire screen by some percentage and cursor magnification that only magnifies a portion of the screen under a cursor or mouse pointer. Often screen readers come bundled with a magnification function, such as Orca in Linux, or with a magnification counterpart such as MAGic for Jaws in Windows. MacOSX comes both with zoom and cursor magnification function to complement VoiceOver. There is also a number of separate tools for different operating systems such as Virtual Magnifying Glass, GMag, or KZoom. Many of these tools offer also optional contrast change as additional functionality to increase legibility.

Document Magnification is provided by most document readers usually via a general zoom function for the entire document, and in some cases also cursor based magnification is offered (for example in DVI previewers for LaTeX documents). In most document viewers, e.g. for PDF or PostScript, as well as word processing system, zooming is achieved by scaling the rendered document. In some programs, in particular, many ASCII text editors, magnification can often only be achieved by changing the font size.

Browser Magnification is similar to document magnification but is handled within web browsers. Zooming is usually achieved by changing font sizes and scaling up embedded graphics. As a consequence the zooming is usually discrete and there is a limited upper bound on the size that can be achieved.

Since magnification depends mostly on rendering on a display, there is no significant difference between the magnification of normal text and mathematical expressions. For instance there is no difference in quality between text or mathematics for magnification via desktop magnifiers or document readers. The only exception are programs that depend on changing font sizes to achieve magnification. This can particularly affect MathML viewing, where some browsers can experience rendering problems for MathML expressions at very high zoom factors.

Magnification works less reliably on graphics whether standalone or embedded in documents. While scalable vector graphics (SVG) can be rendered without loss of legibility in any magnification by the majority of tools, graphics format with fixed resolution suffer from pixelisation and aliasing artifacts. Many scanned documents in EuDML will be in a relative high resolution format (e.g., 600 DPI Tiff graphics), which remains fairly legible even with great magnification. However, low resolution graphics content can become quickly illegible under magnification. In the context of mathematics this particularly occurs with web content created by `latex2html` or by `MathTran` (`http://www.mathtran.org/`) where the mathematical content is transformed into bitmap graphics.

3 Specialist Accessibility Tools for Mathematics

In this section we present an overview on accessibility tools that specialise on mathematics. One necessary requirement for such tools is to transform mathematics into a linear form. A number of specialist formats have been invented for this purpose, which we will survey. We will also provide an overview of tools that convert into these formats and tools that translates mathematics into speech or Braille. Other challenges are given by translating tables, figures or graphics into accessible output such as Braille or another tactile form, and we will briefly mention project working on these problems as well.

3.1 Math Accessibility Formats

Besides the conventional formats that linearises mathematics like LaTeX and MathML, there are a number of specialist formats for providing accessibility for blind people to documents in general and mathematics in particular.

Math Braille has the goal of providing tactile access to mathematical expressions by linearising them into a Braille format. There are numerous national variants of Braille for mathematics and even different variants for the same language, most based on six dot Braille cells, but some also on eight dots. See `http://chezdom.net/blog/?page_id=51` for an overview.

DAISY XML Many accessibility initiatives are driven by the Daisy consortium, an international association that develops, maintains and promotes international standards for Digital Accessible Information System (`http://www.daisy.org`). As a consequence the now globally accepted standard for accessible documents is DAISY XML a markup language that adds accessibility tags to electronic documents and is considered the standard format for digital talking books. Furthermore, DAISY adopts as standards MathML for mathematics, SVG for figures, and XHTML for tables.

The DAISY Pipeline (`http://www.daisy.org/pipeline`) is used to convert documents in traditional markup to DAISY XML. While it can translate XHTML to DAISY XML it does not work on MathML. The Infty project (`http://www.infty.org`) is currently developing a conversion tool from MathML to DAISY XML. Likewise there is a project on developing an SVG analyser to convert figures into tactile SVG graphics in collaboration between Infty and ViewPlus (`http://www.viewplus.com`).

Canonical MathML The central representation supported in UMCL (see Section 3.2) and systems based on it is, in fact, a subset of MathML called *Canonical MathML* [1]. Canonical MathML is an attempt to unify MathML structures in a deterministic way as to simplify transcription into Braille, by requiring mathematical expressions to be coded in a unique way. Nevertheless, Canonical MathML is valid MathML and can therefore be used by any tool that can handle MathML.

A UMCL input module for MathML is available and allows conversion of any MathML document to Canonical MathML. When processing a MathML formula for conversion into Braille, about 80% of the processing time is needed for canonicalisation.

The Braille characters are depicted according to a specific Braille dots representation on 2 bytes. The first byte corresponds to a binary coding of the first column (dots 1-2-3-7) and the second byte to the second column (dots 4-5-6-8). This means that both 6-dot and 8-dot Braille notations are supported. [2]

HRTeX is an extension of regular LaTeX with the idea of establishing a human readable LaTeX format. It's major aim is to simplify LaTeX code for direct reading, therefore making LaTeX documents immediately comprehensible when read by arbitrary screen readers. To our knowledge HRTeX does not have a wide user base.

3.2 Linearising Mathematical Expressions

MathDaisy (`http://www.dessci.com/en/products/mathdaisy/`) is a commercial system developed at Design Sciences. It works with Microsoft Word's Save As DAISY add-in, which saves the document as a DAISY Digital Talking Book ready to be used in an eBook reader. MathDaisy enhances the Word-to-DAISY conversion process, converting the equations in the document to MathML as required by the DAISY format. MathDaisy also works with *MathType* (`http://www.dessci.com/en/products/MathType/`), another Design Sciences tool , that lets a user enter mathematical expressions via a graphical interface, handwriting recognition, or by typing TeX or LaTeX.

Triangle and Wintriangle are a set of tools for displaying and voicing conventional text and the symbols commonly used in math and scientific expressions. They were developed by John Gardener and the Science Access Project at the Oregon State University. WinTriangle is now an open source project at `http://wintriangle.sourceforge.net/`. The idea of Triangle is to provide a format and editor for linearised mathematics. This has resulted in the Triangle.ttf font that contains markup symbols permitting virtually any math or scientific expression to be expressed in a linear form.

LaTeX2Tri (`http://www.ideal-group.org/latex2triangle/latex2tri.htm`) is an open source tool that translates LaTeX files into the WinTriangle format, the RTF word-processing language . It is particularly suited to work with the Jaws screen reader on Windows platforms. LaTeX2Tri has been used extensively at the Harvard University, for the conversion of physics textbooks, arXiv papers, journal articles, class notes, and problem sets. This effort works on PDF files only, using the Infty system for preprocessing.

MathSpeak (`http://www.gh-mathspeak.com/`) is a project initiated by the American mathematician Abraham Nemeth, who invented the Nemeth math Braille code. Its goal is to provide a standard for how mathematical expressions should be linearised when spoken. MathSpeak is defined via a set of grammar rules that tries to define a standard semantics for how certain mathematical expressions should be read. The MathSpeak grammar rules are employed by some screen readers to read aloud embedded mathematical expressions given, for example, in MathML.

UMCL Universal Math Conversion Library (UMCL) `http://umcl.sourceforge.net/` is an open-source portable programming library encapsulating various converters for different Braille codes which enables developers of applications designed for visually impaired people to easily support multiple Braille mathematical notations.

The library can be used just as well with transcription tools (from mainstream notations to Braille and vice versa) as with software that needs real time conversions (e.g. formula browsers or editors). It also makes converting a document from one national Braille notation to another possible.

UMCL was developed in standard portable C with wrappers to various other programming languages (like Java or Python).

An architecture based on a MathML as the central representation of the formula was developed and is quite simple. It includes a main module and as many input and output modules as there are math codes available. These input and output modules are independent and can be installed later on. Also, interfaces for these modules have been published so they can be developed in any language [3]. The main module detects the available input and output modules and calls the modules needed to perform the conversions according to requests.

3.3 Speech or Tactile Access

AsTeR is an Audio System For Technical Readings designed by T.V. Raman at Cornell University [27]. AsTeR renders technical documents in audio, in particular it can take original LaTeX input and translate it into speech. AsTeR can, in principle, be used in conjunction with Emacspeak to render entire LaTeX documents into speech. However, AsTeR was built in order to produce speech via Dectalk, a hardware synthesizer and the current state of support for AsTeR is unclear.

TechRead [13] was a project which, similar to AsTeR, was designed to convert both TeX and LaTeX input into speech as well as into Braille output. TechRead is no longer available as a tool.

MathPlayer (`http://www.dessci.com/en/products/mathplayer/`) is an Internet Explorer plug-in for viewing of MathML. While it is freely available, it is proprietary software produced by DesignScience and therefore not open source. As it works exclusively for Internet Explorer, it is consequently only available for Windows platforms. MathPlayer provides accessibility by performing text to speech translation and by supporting zooming of MathML expressions. MathPlayer can deal with presentation MathML and claims to work on content MathML. However, the latter could not be tested successfully.

Infty and ChattyInfty Infty is a mathematical document analysis system that provides support for many activities related to analysing and accessing mathematical documents. Infty can handle multiple input format for its analysis, such as PDF or TIFF files but concentrates in particular on retro-digitised mathematical documents by means of OCR, formula recognition and ground truthing. There is a variety of peripheral software tools developed within the Infty project, one of which is ChattyInfty that provides speech output for formulae recognised by the Infty system.

Labradoor is an open source tool developed at the University of Linz, Austria, that converts LaTeX to Braille [4].

Lambda is a commercial product `http://www.lambdaproject.org/` that aims at providing support for secondary-school and university students with visually impairments. It is an integrated environment for the visualisation, writing and manipulation of mathematics in a linear format. This linear format was designed to be used directly with Braille peripherals as well as for speech synthesis. The format derives from MathML and conversion between the two is flawless. As a consequence Lambda can work with different type of input and output syntax, such as LaTeX, MathType, Mathematica. Infty2Lambda is a project within Lambda that provides an automated conversion of math texts into Braille, through the Infty OCR engine.

MathGenie is a commercial product developed at Logical Soft for a similar target group as Lambda and for teaching assistance (`http://www.logicalsoft.net/Math.html`). It converts MathML expressions into a multilevel text statements navigable with the cursor keys. The project has the explicit goal of building “an assistive software application that enables blind users to navigate an unambiguous verbal rendering of nontrivial math statements". MathGenie translates a MathML expression into a multi-layer mathematics statement to enable their exploration by users on different levels, thereby trying to exploit as much semantic information as possible to provide a fully disambiguated reading of the expression. While this might be possible for secondary school mathematics, it is unlikely to be effective for more advanced mathematics.

ReMathEx is a Renderer and Editor of Mathematical Expressions for blind students developed at the Faculty of Informatics, Masaryk University [14]. Similar to MathGenie, it allows blind students to explore complex mathematical expressions on multiple levels. The system uses a combination of Braille display and speech synthesis outputs to provide the user with information concerning the mathematical expressions being studied.

ReMathEx system uses MathML input and stores mathematical expressions internally in a parse tree structure. In a first step the whole mathematical expression is previewed by transcribing the tree structure using speech synthesis. The user can then navigate through the expression by walking the parse tree structure. The active part of the expression, i.e., the sub-expression under the node where the user is positioned, is sent to a Braille display using mathematical Braille code.

Math2Braille is a system developed at the University of Amsterdam that provides translation from MathML to Math Braille as it is used in the Netherlands [10]. Math2-Braille supports adding new Braille notation for unknown symbols and provides several XML based schemas and descriptions to allow different protocols to be added for different countries/regions.

M2B *M2B* is a Java application for conversion of MathML documents into Braille [8]. It was developed at Masaryk University and is based on XSLT transformation. A preprocessing step deals with text processing that is impossible during XSLT transformation or that would make the transformation confusing. These steps include:

- Modification of fenced elements to resolve separation between sub-elements.
- Transcription of numbers in text nodes into Braille.
- Conversion of upper-case characters to lower-case ones with appropriate prefixes.
- Processing of a table of replacements defined in a configuration file.

The table of replacements is a way to deal with geographical differences of Braille notation for math. The same operator might be presented in different ways depending on local habit.

An XSLT transformation is the core of the application. The XSL template deals mostly with context elements of MathML. Presentation elements are resolved during preprocessing. The transformation supports multiple Braille notations, defined in a configuration file, depending on language attributes. The result of the transformation is a string without line breaks. These are added during a post-processing step.

BraMaNet is an XSL Style Sheet that translates MathML (Presentation tags only) into French Mathematical Braille. It has a user-friendly Visual Basic interface and can be used together with MathType to translate Word Documents into Braille for printing.

gh ReadHear (`http://gh-accessibility.com/`) is a commercial DAISY Player for reading digital talking books. It supports the reading of embedded MathML by offering both reading of full expression as well as character by character navigation in formulae. One advanced feature of gh ReadHear is a more comfortable navigation of tables, which, for example, allows the user to change reading order or exit tables at any point. However, this navigation only works well for regular tables, i.e., it breaks down for tables containing spanning cells. Also gh ReadHear also has support for SVG graphics, which currently is primarily oriented towards lossless representation of graphics at arbitrary zoom levels.

GATE is a system developed at Masaryk University, aimed at making computer graphics accessible by means of dialogues. The system provides a framework that allows access to graphics in annotated SVG format, where annotations contain information that categorises the corresponding graphical objects into associated binary representations [21]. GATE also supports annotating regular graphics by providing an ontology to eliminate multilingual problems or problems with the annotation and consistency of the annotation process.

Entering an annotated picture, the user is made familiar with short basic information about the picture and some technical data. Then the user gets to a tree structure, into which the data describing the picture are organised. The user can navigate through the tree structure by using a dialogue in “What-Where language”.

To enhance the efficiency of the system, a method of sonification of complex graphical objects, such as colour photographs, based on a hybrid approach combining sound and speech communication was developed. This approach is supported by a special colour model, called the semantic colour model, which assigns sounds to colours. A blind user can browse freely within a graphic while a sound representing the colour of the selected region is played. The boundary of the image is signalled by a special sound. For the annotated SVG format, the user’s orientation is supported by the annotation provided by the format. [22]

4 Automatic Language Translation

Machine Translation (MT) or the automatic translation of text in one natural language into another was one of the first major challenges attempted by computer science with early prototype systems appearing in the 1950s (e.g. [11]). Many good literature surveys exist for Machine Translation, e.g. [18, 31]).

The main historical application of such MT systems was to provide adequate glossing of foreign language articles to allow an educated reader to ascertain the topic and relevance of a document. For example, much early work was done by American nuclear scientists in developing methods to automatically gloss Russian academic papers. If a paper was found to be relevant, it would be translated by a human translator.

During the 1960s, many optimistic claims were made for the potential of fully automatic Machine Translation and many approaches suggested. However, these claims were premature due to the lack of usable data (for instance, fully electronic dictionaries, large sized bilingual corpora) and lack of computational processing power. Finally, in

1966, the ALPAC report (Automatic Language Processing Advisory Committee) (see [24]) resulted in a funding cut for this research which effectively curtailed research in this area till the 1980s. The re-emergence of Machine Translation research was due to electronic publishing providing new sources of linguistic (and bilingual) data and the increase in computational processing power.

Some research projects did continue. Most notably, SYSTRAN, a large MT project with origins in the automatic glossing of academic papers, continued development and has been continuously improved for the past 50 years [29]. To this day, SYSTRAN remains one of the most popular off the shelf MT platforms and underlies the translation capabilities of various Internet MT applications such as Yahoo! Babelfish and AOL and is used extensively by the US Department of Defence and the European Commission. (SYSTRAN was also the underlying technology behind Google language tools until 2007 when it was replaced by an in-house statistically-based MT service.)

Traditionally, Machine Translation Systems can be classified as either direct, transfer or interlingua based systems. In direct translation, the system proceeds word by word through each document, replacing each word with its target language equivalent. After this first pass, post-processing may be used to improve readability by phrase re-ordering. Direct MT approaches are often too inflexible to produce high quality translation. Transfer-based systems try to improve the translation by first analysing the source text and then transforming it so it more closely resembles the target language. After these two stages, translation proceeds in a manner similar to direct translation.

Both direct and transfer approaches have a major limitation in that translation is often between a pair of languages and is uni-directional (e.g. English into French). Therefore translation into a third language (or in the other direction) requires considerable work since it effectively requires building a new MT system. Interlingua-based approaches aim to solve this limitation by translating the source text into a language-neutral semantic representation which then can be used to generate translated documents in a range of different languages. In addition, interlingua-based MT typically allows bi-directional translation. Despite these advantages, interlingua systems still require considerable effort in development and are typically restricted to particular domains. For example, the Verbmobil system [30] provided high quality real time speech to speech Machine Translation to and from English, Japanese and German but was restricted to a hotel-room booking domain. In addition, the high development costs meant that, for most applications and domains, high quality bespoke Machine Translation systems are unrealistic.

However, since the 1990s there has been much promising work on developing purely statistically-based Machine Translation. For example, in the late 1980s and 1990s, techniques were developed for automatically aligning pairs of sentences and phrases from bilingual collections of text. In the 1990s, IBM adapted well-established algorithms from speech recognition, and developed the Candide system (see [5]). Candide was a commercial MT system based on purely statistical methods and bilingual data which bidirectionally translated English and French. Candide relied on the Canadian Hansard, a printed transcript of debates in the Canadian parliamentary, which is published in both English and French with each sentence aligned with its other language counterpart. The quality of Candide was only slightly worse than SYSTRAN yet was a fraction of the development cost.

Candide (and subsequent releases by IBM) have been highly influential on the direction of current research to the point that the majority of Machine Translation research during the last decade has been statistically-based. Progress has also been helped by the release of several publicly available tool kits for Machine Translation, in particular the EQYPT toolkit which provides implementations of Candide and its five subsequent releases (see [15]).

The most recent open source tool kit is based on the Moses system — a phrase-based MT system [19]. This tool kit allows for the development of MT systems which are comparable with commercial statistically based Machine Translation (such as Google's language services), however, all systems require adequate amounts of bilingual training data. This requirement means that for most domains and applications, statistics-based Machine Translation can require the same development costs as traditional approaches.

5 Experiences of Partners

Some of the partners in the EuDML project have had previous experiences with accessibility tools in their individual projects and we summarise these in this section.

5.1 MU's Accessibility Centre, *Teiresiás*

The Teiresiás Centre (Support Centre for Students with Special Needs) has been a part of the Masaryk University since 2000. The Centre's primary task is to provide maximum accessibility to all accredited degree programmes at the University for the visually impaired, the deaf and hearing impaired, people with limited or restricted mobility and people with other types of disabilities. The Centre provides services both to teachers and students by supplying specialist software and hardware and in particularly by making academic texts accessible in different formats. For this the content of academic texts has to be prepared for electronic accessibility.

The creation of an electronic document is not fully automated. First of all, the original source of the text is acquired. If there is not any then the document is scanned and OCRed. The next stage involves manual corrections and modifications to both correct OCR errors, particular for mathematical symbols, and to include information on the document format, page layout, etc. The resulting document consists of simple text in Microsoft Word format to be read with a screen reader.

Math formulae cause problems because there is not any widely accepted standard of coding mathematical symbols in plain text. The Teiresiás Centre deals with it in two ways, depending on how much math is included in a document.

1. Simple math is put in the resulting document as natural language description.
2. Alternatively six-dot Braille is used following the Handbook for transcription by Goncúrová [16], which includes recommendations for encoding of math, physics and chemistry in Braille. This allows for the mathematics to be included and accessed as tactile text instead.

The preparation of tactile texts and graphics is also done mostly manually. The manually corrected *Microsoft Word 2003* document is transformed into Braille Universal Format (BUF) using Visual Basic macros developed at the Teiresiás Centre processing each part of the text depending on its language. BUF allows viewing a document layout in the same way it is printed on a Braille printer. It provides insertion of Braille prefixes (upper-case prefix, number prefix, highlighting etc.) and also processes mathematical and some other special symbols.

To provide accessibility to scientific documents the Teiresiás Centre has access to a number of tools that have been developed at Masaryk University, to provide access to specialist academic texts. These include the Demosthenes TTS engine, the M2B math to Braille converter, the REMathEx renderer and editor for mathematical expressions for blind students, and the GATE system to create tactile graphics.

5.2 Festival Driver for PDF Extractor

The University of Birmingham has developed a Festival driver for the PDF extractor that enables TTS for mathematical formulae extracted from PDF documents. The transcription for a formula is based on the same parse tree structure that is also used to generate other forms of output such as LaTeX and MathML. Currently the driver only uses the basic Festival functionality available via the *SayText* function and none of the advanced features on voice modulation, intonation etc.

The advantages of the current driver are that one gets a faithful speech translation of mathematical expressions. In particular, characters omitted by other screen readers or TTS systems, such as parentheses, can be meaningfully transcribed. There is also no problem with respect to the two dimensional nature of mathematical expressions, as they can be properly linearised from the given parse tree. In particular super- or subscripts are identified and adequately transcribed.

A disadvantage is that the current linearisation of the parse trees can lead to misinterpretations of expressions. The semantics of the actual mathematical expressions are not taken into account. That is, currently mathematical symbols and expressions have one fixed interpretation when being transcribed for Festival, which might not necessarily be the right interpretation in the context of the mathematical document they originate from. Currently there is no context information passed to the PDF extractor, however, one could imagine a library of interpretations for mathematical expressions, that is specific to particular mathematical fields, that could serve for this purpose.

5.3 WAI-ARIA within Webinterfaces

WAI-ARIA support is routinely provided by Made Media Ltd. in user interface design for web applications they produce for commercial customers. The experience is that including WAI-ARIA support does not significantly add to the development process. As a consequence WAI-ARIA can be easily included in the interface design for EuDML thus increasing the accessibility of the EuDML web page.

6 Conclusions

There is currently a plethora of tools available that aim to provide accessibility support for visually or print impaired users. We have only surveyed some of the more important ones. However, one conclusion that can be drawn is that the support for advanced mathematics in accessibility tools is rather limited. It can also be stated that many tools work less well with mathematical expressions than promised. Generic screen readers, document readers and voice based browsers have very limited ability to work with math. For example Jaws or Fire Vox provide some limited native support for MathML. But in general in order to cooperate with generic screen readers it is preferable to have proper transcription of formulae into natural language. This is particular important in the light of the fact that at the moment most speech synthesis tools discard mathematical notation as noise. However, dedicated modules could change this.

There are a number of dedicated tools for mathematics that work fairly reliably with MathML. Systems like MathPlayer, gh ReadHear, and others work well with MathML, but often have limited platform support, e.g., MathPlayer is only available as an Internet Explorer plug-in under Windows. Furthermore, many of these players give mathematical formulae an interpretation that is not necessarily suitable for advanced mathematics, which depends much more on the semantic of the mathematical field and context in which the formula occurs. While many systems work with MathML it should nevertheless be noted that LaTeX is still an important format for some systems. For example Emacspeak together with Aster works well for LaTeX documents. However, the problem here might be that support for these systems is rather limited. Other systems use LaTeX as the base format for translation into other accessible formats such as Braille.

In terms of automated translation tools, there are many commercially available tools available. SYSTRAN, for example, is already widely used and has already been adopted by the European Commission. However, though such systems are capable of producing high quality machine translation output for general texts, they require considerable fine tuning and further development to produce high quality translation of domain specific text such as mathematical literature. An alternative approach is to develop a statistically-based machine translation system which can be trained on existing data to quickly produce a system capable of producing high quality output. Several tool kits exist (e.g. Moses). However, such systems require large amounts of pre-translated data (i.e. example translated documents or aligned sentences) on which to train. As far as we are aware no such source of data exists in the amounts we require. The third option is to recall the origins of research on automated translation and view it as a useful tool for glossing the general topic of a foreign language document. We believe that this may often be sufficient. If the document is then judged to be relevant, it can be translated professionally. However, it is essential to manage user's expectations and to expect a multi-lingual glossing and summarisation service rather than the production of full, high quality translation.

6.1 Suggested Approach within EuDML

We suggest the following approach for providing enhanced accessibility to documents in EuDML. We aim to

- support existing (commercial) tools as much as possible, and
- implement some experimental, specialist services for open source tools to offer directly from the EuDML website.

As a consequence it is paramount that we enhance documents available through EuDML by providing metadata for PDF documents in the form of

1. transcribing mathematical expressions so they can be picked up by generic screen readers,
2. inserting MathML for MathML-aware readers,
3. inserting LaTeX code to work with appropriate translation as well as Emacspeak, AsTeR etc.

In addition WAI-ARIA support will be implemented by Made Media in the user interface.

From personal communication with dyslexic mathematicians, it has also become clear that an important service EuDML should offer is query correction in a mathematical context, e.g. offering possible alternatives to what might be detected as faulty input keywords. In order not to lose performance on general search this should probably be made available as a specialist search option only.

As a second step we attempt to derive and supply some semantic markup (e.g., Content MathML) which can be used by systems like Mathplayer but more importantly can be exploited to provide formula transcription in the correct context. Furthermore, if time in the project permits, we aim to develop a specialist preprocessing and Grapheme-to-phonemes transcription units for Festival, possibly in conjunction with eSpeak as it is widely available and easy to access.

Since automated translation on a major scale is out of the scope of EuDML, automated translation will be primarily keyword based using glossing techniques. Since the mathematical language is rather formulaic one could attempt to offer more by mapping standard phrases. EuDML partners should contribute to a collection for a multi-lingual dictionary for mathematical keywords. However, it will be necessary to manage user expectations, i.e., that one can not expect flawless translation of mathematical documents from EuDML.

6.2 Road Map to Prototype

As a consequence of our suggestions, we are currently planning to provide the first prototype of the accessibility tool set by combining some of the tools developed mostly by partners in EuDML. In particular we want to use the UMCL library which is mature and flexible enough to be useful for EuDML, possibly with some code from the systems M2B and ReMathEx that have been developed at the Masaryk University.

Another part of the tool set will be translation of mathematical expressions embedded in PDF files into accessible expressions using Birmingham's PDF extractor. Currently this module can produce high quality LaTeX and MathML markup as well as some output for Festival. Furthermore the Birmingham partern is currently improving their recognition algorithm for the PDF extractor to take certain basic semantic information into account for the mathematical formula recognition process. These should lead to more meaningful TTS of formulae as well.

References

[1] Dominique Archambault and Victor Moço. Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations. In *Proc. of ICCHP 2006*, volume 4061 of *LNCS*, pages 1191–1198. Springer, 2006. http://dx.doi.org/10.1007/11788713_172.

[2] Dominique Archambault, Bernhard Stöger, Mario Batušić, Claudia Fahrengruber, and Klaus Miesenberger. A software model to support collaborative mathematical work between Braille and sighted users. In *Proc. of the ASSETS 2007*, pages 115–122. ACM, October 2007. http://portal.acm.org/ft_gateway.cfm?id=1296864&type=pdf.

[3] Dominique Archambault, Bernhard Stöger, Dónal Fitzpatrick, and Klaus Miesenberger. Access to Scientific Content by Visually Impaired People. *Upgrade, Digital journal of CEPIS*, 8(2):29–42, April 2007. http://www.cepis.org/upgrade/files/full-II-2007.pdf.

[4] Mario Batušic, Klaus Miesenberger, and Bernhard Stöger. Labradoor, a contribution to making mathematics accessible for the blind. In *Proc. ICCHP98*, 1998.

[5] Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Robert L. Mercer, Harry Printz, and Luboš Ureš. The Candide system for machine translation. In *Proc. of the workshop on Human Language Technology*, HLT '94, pages 157–162. Association for Computational Linguistics, 1994.

[6] Alan W. Black and Kevin A. Lenzo. Building Synthetic Voices. http://festvox.org/bsv/.

[7] Alan W. Black and Paul A. Taylor. The Festival Speech Synthesis System: System documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, 1997. http://www.cstr.ed.ac.uk/projects/festival.html.

[8] Josef Cacek. Konverze technických textů z MathML do Braille (in Czech, Conversion of Mathematical Texts to Braille). Master's thesis, Masaryk University, Brno, Faculty of Informatics, February 2004. https://is.muni.cz/th/4034/fi_m/?lang=en.

[9] Robert Clark, Korin Richmond, and Simon King. Festival 2: build your own general purpose unit selection speech synthesiser. In *Proc. 5th ISCA workshop on speech synthesis*, 2004.

[10] David Crombie, Roger Lenoir, Neil McKenzie, and Alison Barker. math2braille: Opening Access to Mathematics. In *Proc. of ICCHP 2004*, volume 3118 of *LNCS*, pages 627–627. Springer, 2004. http://dx.doi.org/10.1007/978-3-540-27817-7_100.

[11] Lèon Dostert. The Georgetown-I.B.M. experiment. In *Machine Translation of Languages: Fourteen Essays*, pages 124–135. MIT Press, 1955.

[12] Thierry Dutoit, Vincent Pagel, Nicolas Pierret, Francois Bataille, and Olivier Van der Vrecken. The MBROLA project: Towards a Set of High Quality Speech Synthesizers Free of Use for Non Commercial Purposes. In *Proc. ICSLP '96*, volume 3, pages 1393–1396, 1996.

[13] Donal Fitzpatrick and Alex Monaghan. TechRead: A system for deriving braille and spoken output from LaTeX documents. In *Proc. of ICCHP98*, pages 316–323, 1998.

[14] Pavel Gaura. REMathEx: Reader and Editor of the Mathematical Expressions for Blind Students. In *Proc. of ICCHP 2002*, pages 486–493, Linz, Austria, 2002. Springer. http://www.springerlink.com/content/rqrc3rj4qk0523kp/fulltext.pdf.

[15] Ulrich Germann. Building a statistical machine translation system from scratch: how much bang for the buck can we expect? In *Proc. of the workshop on Data-driven methods in machine translation—Volume 14*, DMMT '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[16] Wanda Gonzúrová. *Handbook on Print Transcription Using the Dot Font Norm*. K. E. Macan's Library and Publishing House for the Blind, Prague, Czech Republic, 1997. available (in Czech) on http://www.teiresias.muni.cz/czbraille/.

[17] MBROLA Group. The MBROLA projects — Towards a Freely Available Multilingual Speech Synthesizer. Technical report, Mons Belgium, Théorie des Circuits et Traitement du Signal, 2005. http://tcts.fpms.ac.be/synthesis/mbrola.html.

[18] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

[19] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL demonstration session*, pages 177–180, 2007.

[20] Ivan Kopeček. Speech Synthesis Based on the Composed Syllable Segments. In *Proc. of TSD1998*, pages 259–265, 1998. http://www.fi.muni.cz/~kopecek/TSD.PS.

[21] Ivan Kopeček and Radek Ošlejšek. GATE to Accessibility of Computer Graphics. In *Proceedings of ICCHP 2008*, volume 5051 of *LNCS*, pages 295–302, Linz, Austria, 2008. Springer. http://www.springerlink.com/content/305827771rl12208/fulltext.pdf.

[22] Ivan Kopeček and Radek Ošlejšek. Hybrid Approach to Sonification of Color Images. In *Proc. of Convergent and Hybrid Information Technologies 2008*, pages 722–727, 2008. http://ieeexplore.ieee.org/ielx5/4681983/4682195/04682330.pdf.

[23] Ivan Kopeček and Karel Pala. Prosody Modelling for Syllable-Based Speech Synthesis. In *Proc. of the International Conference on Artificial Intelligence and Soft Computing*, pages 134–137, Cancun, Mexico, May 1998. http://www.fi.muni.cz/~kopecek/kopepala.ps.

[24] John R. Pierce, John B. Carroll, Eric P. Hamp, David G. Hays, Charles F. Hockett, Anthony G. Oettinger, and Alan Perlis. Language and machines computers in translation and linguistics. Technical report, Automatic Language Processing Committee, National Academy of Sciences, National Research Council, 1966.

[25] T. V. Raman. Emacspeak – a speech interface. In *CHI '96 Conference on Human Factors in Computing Systems*, pages 66–71, Vancouver, BC, Canada, April 13-18 1996. ACM.

[26] T. V. Raman. Emacspeak – direct speech access. In *International ACM Conference on Assistive Technologies*, pages 32–36, 1996.

[27] T. V. Raman. ASTER — Towards Modality-Independent Electronic Documents. *Multimedia Tools Appl.*, 6(2):141–151, 1998.

[28] Petr Sojka, Josef Baker, Alan Sexton, and Volker Sorge. A State of the Art Report on Augmenting Metadata Techniques and Technology, December 2010. Deliverable D7.1 of EU CIP-ICT-PSP project 250503 EuDML: The European Digital Mathematics Library, http://eudml.eu/.

[29] Peter Toma. Systran as a multilingual machine translation system. In *Proc. of the 3rd Europ. Congress on Information Systems and Networks, Overcoming the language barrier*, 1977.

[30] Wolfgang Wahlster. *Verbmobil: foundations of speech-to-speech translation*. Springer, 2000.

[31] Yorick Wilks. *Machine Translation: Its Scope and Limits*. Springer, 2008.

[32] Gerasimos Xydas and Georgios Kouroupetroglou. Text-to-Speech Scripting Interface for Appropriate Vocalisation of e-Texts. In *Proc. of the 7th European Conference on Speech Communication and Technology, EuroSpeech 2001*, pages 2247–2250, 2001.

[33] Gerasimos Xydas and Georgios Kouroupetroglou. The Demosthenes Speech Composer. In *Proc. of the 4th ISCA Tutorial and Workshop on Speech Synthesis*, pages 167–172, 2001.