A Project Summary

Automatic Translation of Scientific Literature to Braille

This project develops transcribing software to make scientific and technical literature freely and almost instantaneously available to braille readers through fully-automated translation of both MathML and \LaTeX to Nemeth braille. At the current going rate, it takes over six months to translate a single textbook at a cost of $5000 and more. The software will be the first braille transcribing utility capable of simultaneous and automated processing of mathematical formulas, scientific text, spatial arrangements and XML-described formats.

MathML is an emerging low-level standard markup language for archiving mathematical content, and for communicating and processing mathematics by machines. \LaTeX is a high-level language for writing mathematical documents. Scientific word processors can export documents in \LaTeX, and many authors also use it directly with text editors.

A prototype, which transforms MathML to Nemeth braille mathematics, was built on top of the PI's TeX4ht, a powerful tool for automatic conversion of \LaTeX documents to MathML mathematics plus XML text. This two-layer approach avoided the need to re-address numerous difficulties, already resolved in TeX4ht, which had plagued earlier attempts to convert legacy LATEX to braille and also answers the call of the National Federation of the Blind to develop tools for direct translation from MathML into Nemeth braille.

The software will be highly portable across numerous platforms, and will be disseminated for no fee as open source in the public domain, as is already the case for \LaTeX and TeX4ht. The ultimate outcome of the project should be increased productivity for braille-reading technical persons, and increased integration of braille readers into technical professions.

The subject areas of machine translation of MathML code to other formats and of automatic generation of Nemeth braille from XML-tagged source have received little attention. The work will identify problem areas in both fields, provide solutions with respect to the braille transcribing system, and identify significant topics for further investigation.
Contents

A Project Summary 1

C Project Description 1
  C.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
  C.2 The Translation System . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
    C.2.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
    C.2.2 The Base System . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
    C.2.3 The Prototype System . . . . . . . . . . . . . . . . . . . . . . . . . . 6
    C.2.4 The System Architecture . . . . . . . . . . . . . . . . . . . . . . . . . 9
    C.2.5 Back End Improvements . . . . . . . . . . . . . . . . . . . . . . . . . 10
    C.2.6 Front End Improvements . . . . . . . . . . . . . . . . . . . . . . . . . 11
    C.2.7 The Limits . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
  C.3 The Key Personnel . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
  C.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

D References 1

E Biographical Sketches 1

F Proposal Budget 1

G Current and Pending Support 1

H Facilities, Equipment and Other Resources 1

I Special Information and Supplementary Documentation 1

J Appendices 1
C  Project Description

C.1  Introduction

The U.S. Congress recognizes [1] the right of every individual to “Live independently; enjoy self-determination; make choices; contribute to society; pursue meaningful careers; and enjoy full inclusion and integration in the economic, political, social, cultural, and educational mainstream of American society.” To reach that end, the legislators enacted laws to support the special needs of people with disabilities. The first of these laws, called *The Act to Promote the Education of the Blind*, got enacted in 1879 [2]. That landmark law identifies the central role of access to information for fulfilling the above right, and the special barriers standing before blind people to reach such an access.

Recent advances in computerized technologies made information considerably more accessible to the general sighted and blind population. However, for mathematicians, scientists, and engineers, the case is different. The sighted ones enjoy a readily available online access to a rapidly expanding electronic database of knowledge, while the blind ones to a large degree are left behind without suitable tools to review the available resources.

For most blind professionals in science, mathematics, and engineering, braille is the medium of choice. For blind people with hearing problems, braille is generally the only option. To quote Curtis Chong [3], the Director of the National Federation of the Blind (NFB) Technology Department:

> Although today I make full use of computers with speech output, I find that there is no substitute for being able to put my hands on a complicated piece of code or a document whose language requires careful crafting. Proficiency in literary Braille, the Nemeth Code, and the Computer Braille Code allows me the freedom to proofread, in extreme detail, the most complicated material… Among blind Americans of working age (18–55), there is a staggering 70 percent unemployment rate. Of the remaining 30 percent who are employed, 80 percent or more use Braille. I submit that this is no accident. There is a direct correlation between success in employment and proficiency in Braille.

From an economic perspective, it should be also noted that a jobless blind person costs
the nation $916,000 in life time support and unpaid taxes [4].

Unfortunately, as the MacArthur award winner blind professor Geerat Vermeij noticed “When you go into science, there’s essentially nothing in Braille” [5]. It doesn’t take much to verify this observation. For instance, a search of the library of Congress catalog for publications in 23 categories of mathematics, sciences, and engineering, provided more than 160,000 articles, but with only 55 of them available in braille [6].

No solution to the problem is currently available even for more basic content. The American Printing House for the Blind estimated that, due to a severe shortage of transcribers, only 78 out of the 3000 general textbook published in 1999 were available in braille in January 2000 [7]. The Texas Education Agency estimated that it would take all the available math transcribers in the nation (150–200 [8]) to braille just the new math adoptions in Texas. Transcribing a single textbook can take more than six months [9], and may cost up to $9,500 [10]. Such costs place a heavy burden on institutions and, in most cases, individuals who need material cannot afford the long waits.

When it comes to mathematical content, \[\LaTeX\] [11] is generally agreed to be by far the best authoring language. It is the authoring language recommended by the American Mathematical Society, and it is widely in use. For instance, the arXiv archive holds over 218,000 reports on Physics, Mathematics, and Computer Science, the majority of which are provided with \[\LaTeX\] sources [12]. On the other hand, scientific word processors export documents also in \[\LaTeX\] format. LyX [13], MathType [14], Publicon [15], and Scientific Notebook [16] are such examples.

This proposal asks support for the development of a system for translating general \[\LaTeX\] sources into formatted Nemeth braille, and for making the system easily configurable for other braille codes as the need arises. Nemeth braille [17] is the standard used in North America, and is the one endorsed by the National Federation of the Blind [18]. The work will investigate different designs and algorithms for the implementation, and will consider extensions of braille code which can facilitate proofreading [19].

The system will allow blind scientists, mathematicians, and engineers to tap into the
vast and ever growing amount of documentation available in \TeX to sighted colleagues in their fields. In addition, it will offer to people who are not familiar with Nemeth braille the option of authoring content in this format indirectly through \TeX.

The system could also become a useful tool for preparing math-based educational material for students at all levels of schooling. Considering the current hardship that blind people encounter in accessing mathematical-based content, it is likely that many who would otherwise choose to practice professions requiring a mathematical background currently do not enter such professions. The proposed development thus has the potential to increase the number of braille readers who would choose to pursue such careers.

It is expected that approaches and code developed for the current project will be incorporated also into the gnome braille translator [20], and will answer the call of the National Federation of the Blind to “developers of Braille translation software to develop software that provides Braille translation of MathML documents into the Nemeth Braille Code” [9]. MathML [21] is an international XML standard for publishing mathematics on the Web, endorsed by the National Federation of the Blind [18].

The proposal is to have the new system built on top of our TeX4ht [22] utility for placing \TeX documents on the Web. A prototype, based on the available TeX4ht configurations of XHTML [23] for standard prose and of MathML for math formulas, is already available. As is the case for \TeX and TeX4ht, the system will not be tied up to any specific dialect of \TeX, will be highly portable across numerous platforms, and will be placed as open source in the public domain to all parties for no fee.

C.2  The Translation System

C.2.1  Background

\TeX is a high-level structural markup language originally designed for typesetting mathematics. It encourages logical representation for content, and it employs linear character-based notation. In many senses, a \TeX source looks like regular prose seeded by meta information describing different parts in the source. Consequently, unprocessed \TeX source
files can serve as a crude medium for exchanging information [24],[25],[26]. Nonetheless, the processing of \LaTeX\ source files greatly facilitates access to their content. For sighted readers, the standard processing is to visual views as specified by the \TeX\ compiler and style files. For blind readers, the ideal presentation would be in terms of the Nemeth braille code.

Just as visual views use common visual layout conventions to implicitly express the logical information explicitly specified in \LaTeX, the conventions of Nemeth code eliminate the need to explicitly specify the \LaTeX\ instructions in braille. Hence, in both cases the processed representations are typically more concise than their corresponding \LaTeX\ source code fragments.

Although the original use for \LaTeX\ was high-quality typesetting for mathematical text, other output formats have also been targeted in recent years.

The MAVIS project [27] aimed at producing Nemeth braille from a \LaTeX\ dialect exported by Scientific Notebook, the LABRADOOR project [28] considered a German math counterpart, and the TechRead project [29] dealt with general technical braille. The AsTeR system [30] provides auditory presentations. Semantical representation in term of Lisp for computer algebra applications has also gotten consideration [31]. As to other visual views, dozens of applications have been conceived for translating \LaTeX\ into hypertext, with the most prominent products being \LaTeX2html [32], TeX4ht, and TTH [33].

The application of \LaTeX\ for communicating mathematics with blind students is as old as \LaTeX\ itself [34]. An early attempt to obtain automated translations to Nemeth braille also took place [35].

According to the National Federation of the Blind, “Braille translation software has not achieved the sophistication that Braille advocates want” [18]. Similarly, the Texas Partnership for Increasing Braille Production stated that “Translation software does not currently work with math-based subjects” [7]. It is of no surprise that in effect there are no tools available for translating technical literature into braille. The publishers of elementary and secondary school textbooks consider the job to be daunting already at that
level of material [36].

### C.2.2 The Base System

TeX4ht is a highly configurable \TeX-based [37] system for producing hypertext. It interacts with \TeX-based applications through style files and postprocessors while leaving the processing of the source files to the native \TeX compiler. Consequently, TeX4ht can handle the features of general \TeX-based systems.

So far, no other conversion system for \LaTeX enjoys similar capabilities as TeX4ht. There are two main factors for the restricted power of the alternative systems. With the exception of Omega [38], they all embarked on building interpreters of their own from scratch (see, for instance, [39]). This is a painful undertaking considering both the complexity of the native \TeX compiler and the fact that the compiler is generally taken to be among the most complete and admired programs ever written. The second factor is that most of the alternative systems hard-code many features of \LaTeX and leave little room to reconfigure them for the variations defined in the numerous style files and fonts of the language.

The MathML component of TeX4ht, which is of special interest for us here, has recently gotten increased attention because MathML itself picked up momentum. Specifically, Netscape 7.0 adopted the MathML rendering facilities of Mozilla [40], Design Science released a free MathML display engine for Internet Explorer named MathPlayer [41], and universal style sheets have been offered for displaying MathML by different browsers [42]. Consequently, more people started translating their documents into MathML with TeX4ht, and the documents themselves grew in size and complexity (for instance [43],[44]). With the increase in use came also an increase in valuable feedback. That feedback resulted in an added support in TeX4ht for the special needs of MathPlayer and the style sheets, and improved quality of the general output. This work is to be discussed in an invited presentation [45].
C.2.3 The Prototype System

The inspiration for targeting TeX4ht to produce braille code came from the BraMaNet utility [46]. That utility relies on XSL style sheets [47] for translating MathML to French braille mathematics. These style sheets work in conjunction with the MathML output of MathType [14] to translate the mathematical portions of Word documents into braille. In scope, however, these style sheets are quite minimal, reflecting on the French mathematical braille code having only a small subset of the functionality of the Nemeth braille code.

Following the BraMaNet lead, we used XSL to code the Nemeth rules for the major mathematical constructs, including quite a few of the exceptions specified for these rules. For the purpose of testing, we expressed over 1700 examples from the Nemeth manual book in \LaTeX, compiled the examples with TeX4ht for XHTML and MathML, and filtered the output with our new XSL style sheets. To our great satisfaction, with the exception of a very few examples for which the style sheets are still incomplete, the outcome turned out to be perfect.

The prototype is already in its third version, after stretching to more than 1,000 mega bytes of XSL code and subsequently being reduced to 30% of that size through normalizations. Nonetheless, major challenges for reaching a useful prototype still lie ahead, with the following being just a few examples.

Rewriting of Code for Text Translation  Quite a few components of the prototype need to be reimplemented from scratch, using yet undetermined alternative approaches. For instance, such is the case for contractions, where traditional approaches are modeled after finite state transducers. The latter approaches not only do not integrate well into XSL style sheets, they are far from being ideal as they rely on ad hoc look-up tables to handle exceptions, due to the unavailability of general purpose algorithms. Considering that finite state transducers allow linear one-way scanning of strings, while XSL-based algorithms permit exhaustive search for contractions through recursion, it might be the case that algorithms of the latter nature can offer superior
solutions for the problem. However, the effort involved in researching such algorithms, and in establishing corresponding look-up tables if the need arises, does not seem to be justified before other issues are addressed.

**Implementing Missing Rules** The prototype system currently deals with a mathematical core rules of the Nemeth code, and a core of the English braille rules [48]. So far it provides no support for the Braille Format rules [49] and the Chemical Notation [50]. Much attention is still required by the 195 Nemeth rules, the many exceptions to the rules, and the relationships between the rules, including the complex problems of spatial arrangements (rules §178–§184) and formats (rules §185–§195). For instance, in typesetting a simple structure like a matrix one needs to take into consideration that braille pages are restricted to widths of 40 braille characters and 25 rows.

To overcome the width constraints, the entries in a matrix might require folding into short lines (rule §183), where the foldings are allowed to take place only at some specific locations. A folding at a single entry is likely to call for reconsidering the typesetting of the other entries across the row and column that contain the entry, to preserve horizontal and vertical alignments and to minimize the presentation size. In addition, it might also add special indicator symbols into the entry. Without appropriate safe guards, the process may enter an infinite loop.

To overcome height constraints, and possibly also help dealing with width constraints, the content of cells might need to be moved elsewhere and be replaced by pointers to the content (rule §187).

Additional complication arises because the braille code consists of only 63 symbols (plus space). The math symbols that occupy just a single location in print are likely to claim several locations in braille. For instance, the Greek character of alpha requires two locations, an approximately equal (dot over equal) sign occupies six locations, and a parallel up-down double-arrow sign claims eleven locations.
We believe that some problems are theoretically intractable, in the sense that they are NP-hard [51]. It is beyond the scope of this project to conduct such an in depth investigation. Instead, we will concentrate on finding heuristic approaches that prove to offer good results.

**Normalization** Normalization is another issue requiring a major investment of effort. This issue has received only scant attention to date in order to keep the code under manageable complexity.

An example of the problem exists even at the primitive level of characters. For instance, a mathematical bold digit 0 may be encountered in the Unicode-based form `<mn>&#x1D7CE;</mn>` or in the attribute-based form `<mn mathvariant="bold">0</mn>` ([21], Section 6.2.3). The two forms are equivalent, and so they need to be treated as such.

The translation of each digit depends on its own value and the value of its predecessor (rule §32b). Consequently, for a bold math digit 0 that is preceded by another bold math digit 0, the translation needs to take care of four possible combinations of the two variants of the digit representations. On the other hand, replacing each `<mi>&#x1D7CE;</mi>` with `<mi mathvariant="bold">0</mi>` reduces the number of combinations to one.

The possible savings is generally much greater. Consider the number of combinations that can arise from ten digits instead of just the single digit of 0. Add to that double representations for lower and upper case bold math Latin characters, and for bold math Greek letters. Then consider double representations not just for the bold math style, but also for 12 other math styles such as the sans-serif, double-struck, and fraktur variants ([21], Section 6.3.6). And, on top of all of that, add also the special treatment of subscript digits when they follow a letter (rule §32b), and the attention the different letters should receive when they are accented and modified (rule §86).

Transformations motivated by multi-equivalent representations are not restricted just
to characters. For example, the simple math expression ‘(x)’ can be represented by both `<mfenced> <mi>x</mi> </mfenced>` and `<mrow> <mo>(</mo> <mi>x</mi> <mo>)</mo> </mrow>` ([21], Section 3.3.8). Similarly, extra mrow tags may appear on many blocks of code without altering the meaning, but not over all blocks (an extra mrow over the second argument of an munder, for instance, can be harmful).

**Interpretation of Code** Improvement in the understanding the meaning of code is another major challenge. That is obviously the case when dealing with content authored by authors who do not play fully according to the rules, but it also may result from shortcomings of representations.

For instance, the code fragment `<msub> <mi>a</mi> <mi>b</mi> </msub>` stands for the expression ‘a subscript b’. On the other hand, the following code fragment of similar structure

```xml
<msub> <mi>a</mi> <mrow> <menclose notation='actuarial'> <mi>n</mi> </menclose> <mo>&it;</mo> <mi>i</mi> </mrow> </msub>
```

says that ‘a is the result of dividing n by i’ ([21], Section 3.3.9.3). The first code fragment faithfully reflects on the intended interpretation of `<msub>`. The second fragment offers a misleading interpretation.

**C.2.4 The System Architecture**

The architecture of the braille system consists of two main layers: a front end layer made up of our TeX4ht for translating MathX into an intermediate format and a new back end layer for translating the intermediate format into Nemeth braille. The effort to polish the prototype system into a tool for regular use will require changes to the back end layer as well as further development of the front end layer.
In the remaking of the prototype system into a production tool, it is our intention to consider potential reuse of code by other applications. Consequently, we will try to employ a modular architecture with maximum use of general-purpose modules. For instance, for the XSL transformations, we will write general style sheet files which take no special clues from TeX4ht and supplement them with specialized style sheet files which take such clues. The general style sheets will then be usable in other projects and by other teams. The specialized ones will allow us to take advantage of features which are tailored specifically for our use.

C.2.5 Back End Improvements

The TeX4ht system is routinely being expanded and modified, in response to the evolving W3C specifications [47] and requests from users for new features. Nonetheless, the major components of the system are in place and running smoothly. Consequently, establishing a prototype subsystem for the back end layer and refining its modules are our main objectives. In particular, we expect the following issues to require special attention.

Performance Speed is definitely a problem for the prototype, and it is too slow for users interested in translating large documents while waiting online. We suspect a solution to the problem might require replacing the XSL style sheets with special purpose C/C++ modules tailored to take advantage of the specific features of the system.

MathML and Beyond MathML seems to be the best available standard for marking up the intermediate representations of the mathematic formulas under translation. However, it is by no means a perfect choice for the job.

For instance, the content of a standard mathematical text element \(<mtext>\) cannot be further marked for words and characters, yet the identification of such subelements is needed for dealing with contractions (rules §55–§56) and for handling capitalization (rules §20–§22). Similarly, the \(<mo>\) elements may hold symbols of grouping
(rules §120–§128), operation (rules §129–§138), or comparison (rules §139–§151) without identifying the different categories within the attributes of the elements.

Consequently, a variant of MathML, enriched with extra tags and attributes, might turn out to be beneficial for the translation of $\LaTeX$ to braille. Much of the extra information is already known to the TeX4ht machinery, and can be requested within a special purpose configuration file, to be used just in the translations into braille.

**XHTML** When it comes to prose, XHTML seems to be a good choice for the job because it is concerned only with a few types of logical constructs and allows for some presentation markup. These features to a large degree fit the needs of the grade two braille, as modified by the rules of the Nemeth code.

However, the default configuration file provided by TeX4ht for XHTML needs some adjustments to better serve the translation into braille. For instance, it sends font information to a Cascading Style Sheets (CSS) [52] file which makes it impossible to retrieve the information through XSL transformations. As is the case for the MathML markup, the needed information can be requested within a special purpose configuration file for use only in the translations into braille.

### C.2.6 Front End Improvements

All the information to be channeled to the back end layer is collected by TeX4ht. The more information TeX4ht collects, the better understanding of the $\LaTeX$ source files the system has. A better understanding, in turn, can be reflected in the XML code sent from the front end layer to the back end layer. An improved XML code allows the back end to make more informed decisions as it determines the braille code to be produced.

Consequently, the front end layer sets the limits to how good the translation into braille can be. The following list is concerned with pending issues of TeX4ht which have implications for Nemeth code production.

**Extensions** $\LaTeX$ is an ever changing and growing environment, and TeX4ht must evolve
accordingly. Hooks are introduced and configured for additional style files in response to users’ requests. In addition, the underlying building blocks and architecture of the system are still being modified and enhanced in order to support a growing array of demands from users and to adjust for developments in Web technologies [47]. Such modifications have the potential of benefiting the brailling project.

For instance, \LaTeX provides support for typesetting chemical formulas [53]. \TeX4ht does not yet have configurations dedicated for the typesetting of such formulas, and so, as a default, the formulas are transformed into pictures. Any work to configure the style files to support alternative transformations should turn to be useful for braille as well as for CML [54]. The brailling of formulas must obey a special standard [50] which extends the Nemeth code conventions.

**Low-Level Features** Currently, \TeX4ht has no access to the internals of the native compiler of \TeX which makes it difficult or even impossible for the utility to properly deal with some files.

For instance, \TeX is likely to fail on sources that use category codes to dynamically alter the parsing rules of the superscript and subscript operations of \LaTeX, use these operations in a manner invisible to \TeX4ht, or introduce empty bases for these operations. Sources of this kind are not very common and, when no changes in category codes are involved, the problems are in general easy to detect and fix with human intervention. Human intervention, however, is rarely an option for consumers who download \LaTeX sources, due to lack of knowledge of the language and time to fix the problems.

Unfortunately, modifying the native compiler is a major task, and the return is relatively small. The potential return is limited because most of what the compiler is capable to offer is already reflected in a processed form within the (DVI) code it produces.
Nonetheless, we would like to get access to some internals of the native $\TeX$ compiler in order to capture semantics which is difficult or impossible to reach in other ways. For example, lack of such capabilities may deprive the system the ability to collect the information needed for handling left superscripts and subscripts (rule §75).

**Error Detection and Correction** The XML outputs of $\TeX4ht$ include comments showing the line numbers in the source $\LaTeX$ files which are the origin of the corresponding XML elements. On the other hand, the error messages issued by the validators use pointers into the XML files to show where the problems are. A trained user can without much effort relate the two locators so as to determine the origin of problems in the $\LaTeX$ sources as necessary for identifying their cause and fixing them. However, this approach is likely to discourage the casual user.

A not too complex utility, with a friendly user interface, could make these error reports much easier to understand. Moreover, such a utility might also be able to automatically correct some kinds of errors.

### C.2.7 The Limits

With the exception of specially tailored $\LaTeX$ dialects (for instance, [55] and [56]), it is unrealistic to expect the final tool to produce perfect output for all $\LaTeX$ sources (see, for instance, [35] and [31]). This well recognized problem is due to the inherent ambiguity involved in mathematical discourse, features of $\LaTeX$ (inherited from $\TeX$) which allow the typesetting of fragments of mathematical expressions without specifying their semantics, and authoring mistakes.

However, we believe that from a practical point of view the tool has the potential of becoming an almost perfect utility. Heuristics can be introduced to confront peculiar cases, many of these cases are rarely if at all used, and when they are present they typically affect only a few small isolated spots of the documents. In fact, a similar attitude for dealing with exceptions is already in place for standard algorithms that deal with contractions in
literary braille (see, for instance, [57]). Specifically, these algorithms assume user-supplied configuration tables containing instructions on how to deal with known exceptions.

C.3 The Key Personnel

Eitan M. Gurari is the developer of the TeX4ht system, and is currently also the sole developer of the extension of TeX4ht for producing braille. It is hoped, however, that graduate students will have the opportunity to conduct the research and development of some advanced features for the system.

Susan J. Jolly-Woodruff is self-educated in braille, she has contacts with numerous braille educators, individual transcribers and transcribing services, and is the author and maintainer of a large online site about braille [19]. Her interest in braille was inspired by a similar interest of her father's [58]. Susan suggested the project discussed in this proposal, is the major contributor of the background information needed to run the project, and she plans to take on a major role in the evaluation and proofreading stages.

John J. Boyer is a computer scientist and Nemeth expert. He founded and is the director of the non-profit organization, Computers to Help People, Inc. (CHPI). CHPI's main activity is the Nemeth braille transcription and publishing of technical textbooks with the major customers being institutions of higher learning [59]. He was the main developer of MegaMath, the mathematics feature of MegaDots [60], one of the most popular and powerful braille production programs in the world. He recently completed a Grade 2 braille module [57] for the brltty virtual terminal used with Linux and he is in the process of designing Gnomebrl, a comprehensive open source braille transcribing environment [20]. John's main contribution to the discussed project is as an authoritative consultant on braille, the technologies involved, and the production processes in use.

Both Susan's and John's work on this project are performed on a volunteer basis by personal choice.
C.4 Conclusion

\TeX is generally agreed to be by far the most sophisticated high-level language for authoring mathematical content. High quality scientific word processors can export documents in such format, and many authors also use it directly with text editors.

MathML is an emerging low-level standard markup language for describing mathematics. It is provided for machine processing of mathematics and for placing mathematics in web pages. XHTML is a complementary markup language for regular text.

\TeX4ht is a powerful conversion tool capable of translating \TeX to MathML plus XHTML. It is used for converting \TeX exported from word processors (for instance, \[44\]), and \TeX documents created directly with text editors (for instance, \[43\]).

This proposal outlines a project for automating the translation of both MathML and \TeX into Nemeth code. Since TeX4ht is already a working utility, the proposal addresses mainly issues that relate to MathML.

The work is expected to be directly applicable to other systems for producing MathML. The sophisticated capabilities of \TeX ensure that an especially rich and challenging array of MathML documents will form the basis of our investigation of the issues involved.

A project with this goal is the only feasible way for braille readers to acquire access to the huge amount of technical documentation that is available to sighted students, scientists, mathematicians, and engineers.

The proposal discusses very ambitious objectives. We believe our team is well rounded and highly qualified to deal with the posted problems. Moreover, we strongly believe we will be able to deliver a good and useful system addressing the core issues under the proposed frame work.
D References


[22] Eitan M. Gurari, “TeX4ht: Mi\TeX and \LaTeX for Hypertext”, http://www.cis.ohio-state.edu/~gurari/Tex4ht/.


[33] Ian Hutchinson, “TTH: The $\TeX$ to HTML Translator”, http://hutchinson.belmont.ma.us/tth/.

[34] Malcolm W. Clark, “Mathematical Communication with a Deaf and Blind Student Using $\TeX$”, *TUGboat* 5:2 (1984), 146.


[52] Cascading Style Sheets (CSS), http://www.w3.org/Style/CSS/.


E Biographical Sketches

Eitan M. Gurari (PI)

Professional Preparation

Technion - Israel Institute of Technology  Physics  B.S. (1971)
Technion - Israel Institute of Technology  Computer Science  M.S. (1974)
University of Minnesota  Computer Science  Ph.D. (1978)

Appointments

• 1982–present, Associate Professor, Ohio State University, Department of Computer and Information Science.

• 1980–1982, Assistant Professor, State University of New York at Buffalo, Department of Computer Science.

• 1978–1980, Assistant Professor, University of Wisconsin - Milwaukee, Department of Electrical Engineering and Computer Science.

Publications Related to the Proposed Project


Publications not Related to the Proposed Project


Collaborators

John J. Boyer, Co-PI on this proposed project (resume attached)
Paul Gartside, Department of Mathematics, University of Pittsburgh
Michel Goosen, CERN, Switzerland
Susan J. Jolly-Woodruff, Co-PI on this proposed project (resume attached)
Ross Moore, Macquarie University, Sydney
Sebastian Rahtz, Oxford University, England
Robert S. Sutor, IBM Research, Yorktown Heights
Philip A. Viton, Department of City and Regional Planning, Ohio State University

Graduate Advisors

Oscar Ibarra, University of California at Santa Barbara

Thesis Advisor and Postgraduate-Scholar Sponsor

Jesse Wu (Ph.D. student), Professor at the National Kaohsiung Normal University, Taiwan.
John J. Boyer (Co-PI)

Professional Preparation

College of St. Thomas, St. Paul
University of Wisconsin-Madison

<table>
<thead>
<tr>
<th>Institution</th>
<th>Major/Field</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics</td>
<td>B.A. (1961)</td>
<td></td>
</tr>
<tr>
<td>Computer Science (major),</td>
<td></td>
<td>M.S. (1982)</td>
</tr>
</tbody>
</table>

Appointments


Synergistic Activities

1. Founded and runs COMPUTERS TO HELP PEOPLE, INC. (CHPI) on a volunteer basis [59].


4. Developed the Grade 2 braille module for the brltty virtual terminal used with Linux. http://dave.mielke.cc/brltty/
Susan Jenifer Jolly-Woodruff (Co-PI)

Professional Preparation

Oberlin College Chemistry A.B. (1962)
Johns Hopkins University Chemistry M.A.T. (1963)
University of California at Irvine Theoretical Chemistry Ph.D. (1977)

Appointments


• 1979–1993, Technical Staff Member, LANL (took Voluntary Early Retirement Incentive Program in October 1993). Responsibilities encompassed model and code development and user liaison for TRAC and code development for other scientific applications including classified weapons design.

• 1977–1979, Director’s Funded Post-Doctoral Fellow, LANL.

• 1963–1966, Teacher, high school, Basic and Advanced Placement chemistry.

Publications not Related to the Proposed Project


F Proposal Budget

Salaries and Wages

1. Eitan Gurari (PI) is involved in the project as an employee of Ohio State University. Salaries and related benefits are requested for him.

2. John Boyer (Co-PI) is the founder and executive director of CHPI. He works there on a volunteer base, and he is not paid for any thing he does for the company. Susan Jolly-Woodruff (Co-PI) is a retiree with no affiliation to a work place. John and Susan are involved in the proposed project on a volunteer base. They are not being paid for this work, and no budget is requested for such a purpose.

3. Support is being asked for two graduate students. We would like to get them involved with different issues of the system development.

4. We are asking a budget for proofreading services (probably from CHPI), to be provided by professionals trained to work with Nemeth code. We would like to get such services for the preparation of specialized sample test material to be used in the debugging of the system.
Equipment and Software

Eitan  
| MS-based PC | $2,000 |
| MS-based C/CC compiler | $200 |
| Braille translation software: MegaDots, Scientific Notebook, Duxbury | $1,700 |
| Screenreader | $2,000 |
| Scanner & OCR software | $500 |
| OBR software | $850 |
| Linux-based server | $1,500 |

Susan  
| MS-based PC | $2,000 |
| Braille translation software: MegaDots, Scientific Notebook, Duxbury | $1,700 |
| Screenreader | $2,000 |
| CorelDraw & MS Office | $500 |
| Scanner & OCR software | $500 |
| OBR software | $850 |
| Brailler | $700 |

John  
| Tiger Embosser | $10,000 |
| OBR software | $850 |

1. John will be working at CHPI in Wisconsin (Madison), Susan at her home in New Mexico (Los Alamos), and Eitan at Ohio State University (Columbus). Two of us request a MS-based PC to facilitate the sharing of the tools we develop as well as for testing and demonstration.

2. Much of the software used by braille-reading people is tailored for MS-based PC environments. Eitan and Susan would like to test some of this software and get familiar with it. The proposed budget requests a few tools of this kind. The choice of the MS-base environment for the requested PC’s is in part motivated by the desire to get to test related software developed elsewhere.

3. The scanner and the OCR and OBR software are requested to help transporting into electronic form test material available in print form.

4. John is both blind and deaf. The requested Tiger embosser is for him to use for accessing tactile graphics, and for him to configure for the output of our proposed tool. Eitan and Susan will rely on simulated representations of such graphics.
5. For security reasons, the Computer and Information Science department at Ohio State University does not provide servers to its members. The budget includes a server which will be dedicated to remotely translating \LaTeX\ documents submitted by the general public into braille. The server will help us refine our tool and also promote its use. It will be placed at Ohio State University.

**Travel**

The following are the meetings we would like to attend. Since John is blind and deaf he will need to be accompanied by an interpreter.

1. ACM SIGCAPH Conference on Assistive Technologies, \url{http://www.acm.org/sigcaph/}, biennial, domestic.

2. The CSUN Technology and Persons with Disabilities Conference, \url{http://www.csun.edu/cod/}, annually, Northridge, California.


6. TUG Meeting and Conference, \url{http://www.tug.org}, annually, domestic.
G Current and Pending Support

A similar proposal is under consideration by the U.S. Department of Education.
H Facilities, Equipment and Other Resources

The major development work is expected to be conducted at Ohio State University using the Unix computing facilities available in the Computer and Information Science department. It will require standard software such as \TeX, \LaTeX, C/C++, and XML tools. Some of the software is already installed in the department, and the missing software is readily available as free downloads from the Internet.
I Special Information and Supplementary Documentation

The roles of the investigators in the collaborative proposal

The proposal asks support for a collaborative effort, aimed at developing software for making technical documentation accessible to persons who are braille readers. The collaboration involves three people: John J. Boyer, Eitan M. Gurari, and Susan J. Jolly-Woodruff.

Eitan M. Gurari is expected to be the main contributor to the design and code development for the proposed software which will be based on a tool, TeX4ht, that he developed. He is familiar with the technologies required for developing the software, but he is lacking the necessary insight to fully understand the needs of the community to which the software is intended.

Susan J. Jolly-Woodruff will be in charge of investigating proofreading by sighted people. In addition, she envisioned our project, had the leadership role in bringing our team together, and provided many good ideas and critical thinking for the definition of the project. Considering her contribution so far, and her great interest and involvement in the subject matter, she is expected to continue advancing the project in a similar manner.

John J. Boyer will provide the needed guidance for the project. His knowledge of braille from all possible aspects, his background in mathematics and computer science, his experience with programming, and his long time service to the community we want to address, make him the ideal person for this task.

Rationale for performance of all or part of the project off-campus or away from organizational headquarters.

Susan J. Jolly-Woodruff is a retiree with no affiliation to a work place. She is expected to use her home as base for her work.

Documentation of collaborative arrangements of significance to the proposal through letters of commitment.
J Appendices