SwiExr: Spatial math exercises and worksheets, in Braille and print

Nandan Bagchee and Eitan M. Gurari
Ohio State University
gurari@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~gurari

Abstract

LaTeX is a highly expressive authoring language considered to be the lingua franca of the mathematics community. Yet, until recently, except for a few contributions concerning long division, it offered very little support for expressing planar layouts of problems that employ elementary mathematic operations.

We present a highly configurable tool [2] (written in Java) for producing spatial representations of elementary math exercises and worksheets. The available configurations produce verbatim and tabular forms of exercises and worksheets in print and Nemeth Braille formats for inclusion in LaTeX, MathML, HTML, and text files. In addition, they ensure Braille output entirely equivalent to material prepared in print formats. Our current attention is devoted to the addition, subtraction, multiplication, division, and root operations.

We are interested in identifying potential users from the LaTeX community with the objective of developing widely acceptable LaTeX interfaces for requesting math exercises and worksheets.

1 General background

The arithmetic tasks of addition, subtraction, multiplication, division, and square root assume a central place in elementary math education. Spatial arrangements of problems with such operations offer the foundation for applying computational procedures to instances which involve operands with large values. Students are expected to acquire fluency with the algorithms involved in dealing with the representations and understanding of the underlying ideas. To reach this end, they typically practice the approaches to the point of automaticity. The teachers are required to supply variants of the problems for the students to drill with.

Printed and written material, in original and xeroxed forms, as well as exercise program generators, are readily available to sighted students and their teachers. When it comes to math in Braille, outstanding detailed guidelines are available for creating worksheets of exercises [7, 8] but very few actual resources for blind students are provided. The guidelines are legal requirements for Braille to be produced according to the Braille Authority of North America standards [4].

Figure 1 exhibits a sample of spatial layouts of problems provided in the guidelines. On the other hand, Table 1 shows some data obtained in searching the web for math worksheets. In the case of Braille, only one of the entries turned out to be relevant: it offered worksheets of arithmetic operations on numbers of single digits, and got listed in 1999 with a cost quote of $110 per 650 worksheets.

`math worksheets`	2,110,000
`"math worksheets"`	276,000
`math worksheets Braille`	17,600
`"math worksheets" Braille`	223

Table 1: Number of entries listed for different searches with Google.

The problem of a lack of mathematical and scientific content in Braille, and the high cost of producing such material, are well known. For instance, the American Printing House for the Blind estimated that, due to a severe shortage of transcribers, only 78 out of the 3000 general textbooks published in 1999 were available in Braille in January 2000 [11]. Transcribing a single textbook can take more than six months [11], and may cost up to $9,500 [6].

A vast amount of scientific content is available

Editor's note: This material is based upon work supported by the National Science Foundation under Award No. IIS-0312487. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

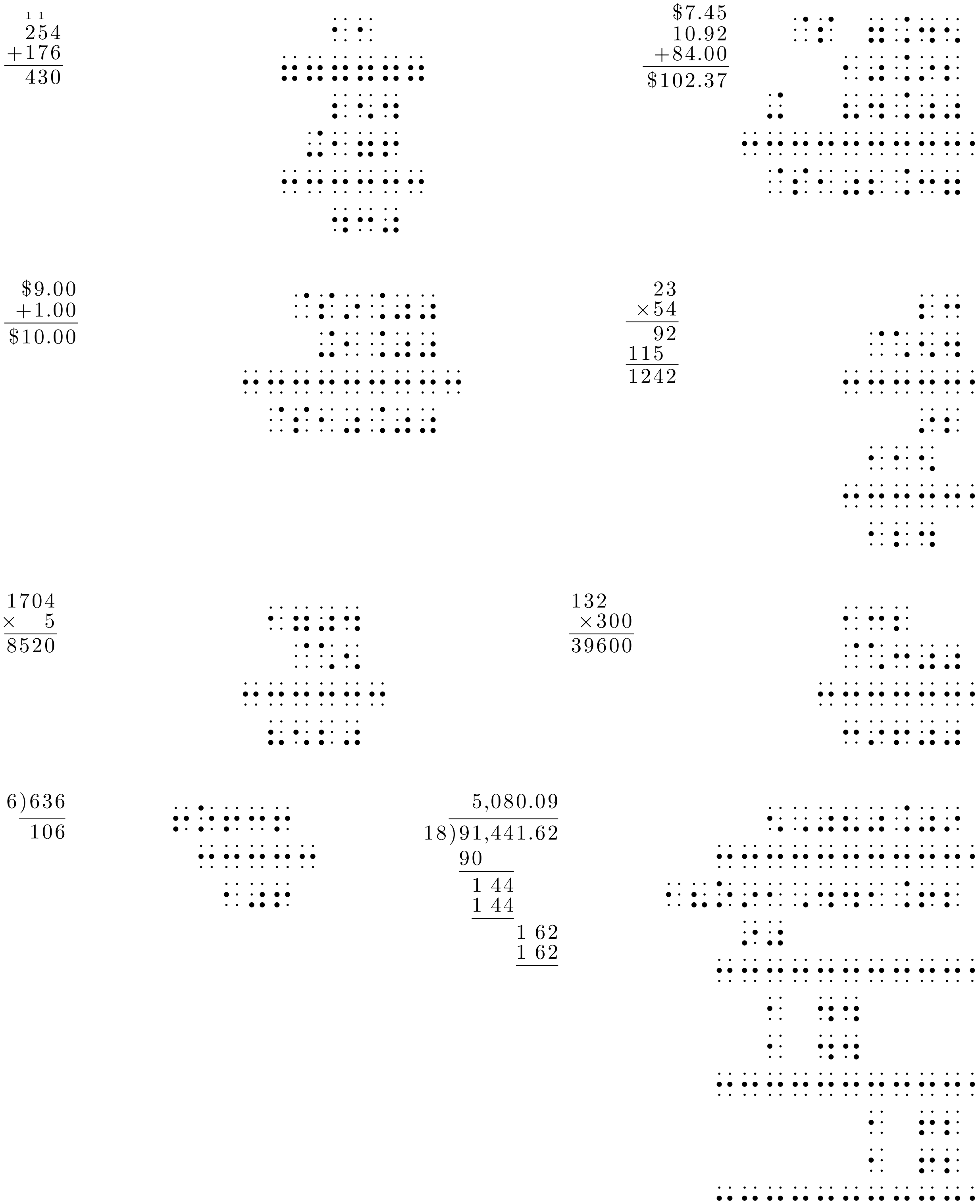

Figure 1: Spatial layout of math exercises in Braille.

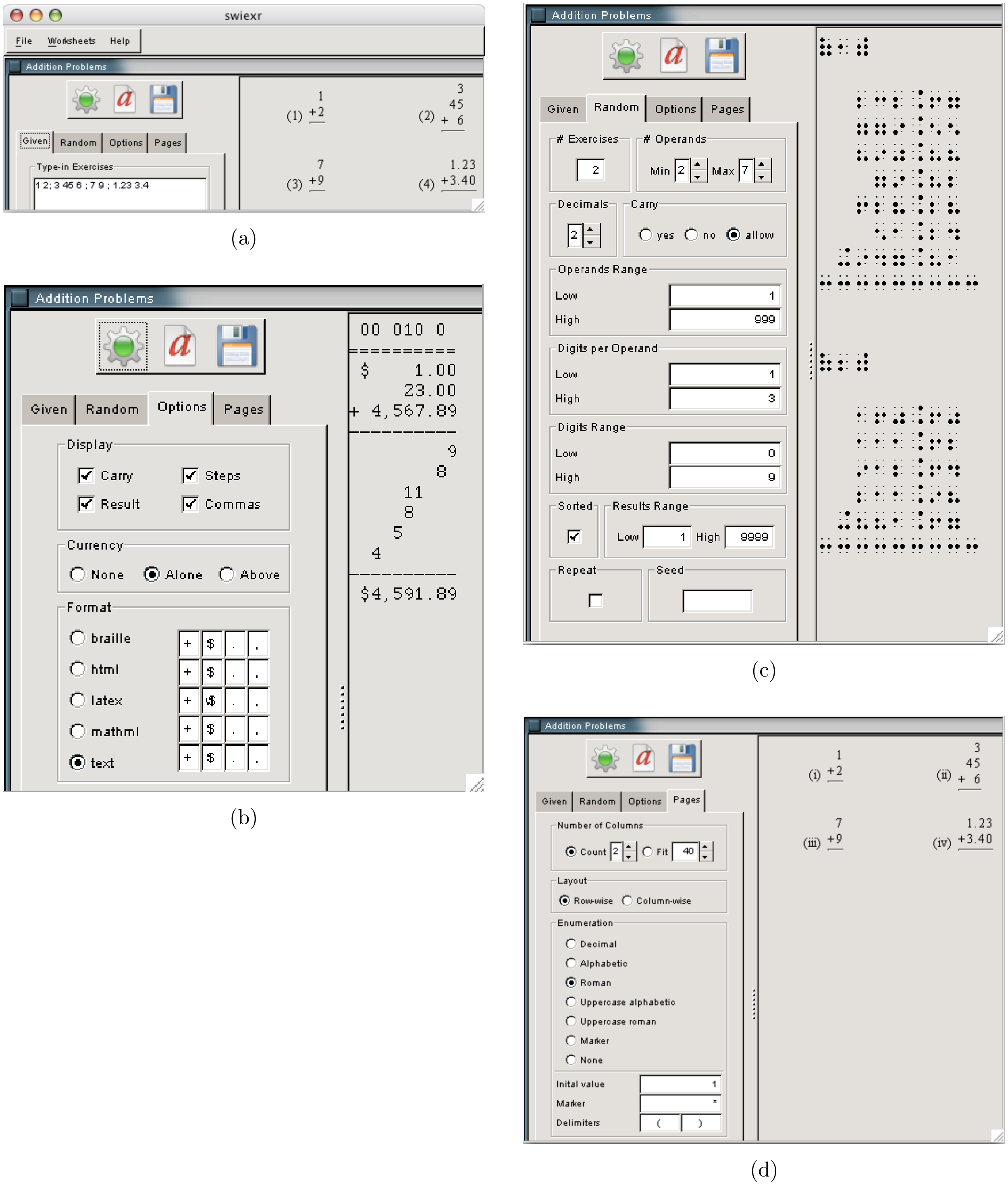

Figure 2: Graphical user interfaces for addition.

in LaTeX, authored either directly with text editors or indirectly through word processors that export LaTeX. LaTeX also seems to be the authoring language preferred by visually impaired scientists.

The objective of our work in general is to automate the production of technical and scientific material in Braille, with an emphasis on translating content available in LaTeX. Our current attention is devoted to the area of producing spatial arrangements for arithmetic exercises.

We introduce a highly configurable utility dedicated to this task, capable of producing worksheets in different forms for general users, while ensuring an option of Braille output for all supported features. We present a graphical user interface (GUI) of the tool, and exhibit a corresponding XML representation for the data to be fed into the tool. Then we consider LaTeX in the role of a user interface for the tool.

2 Graphical user interfaces

In order to serve a large base of users with different needs and preferences, our tool was designed in a modular manner which allows easy connection to independent standalone user interfaces. Figure 2 displays GUIs currently provided for the addition operation. GUIs of a similar nature are also offered for the other operations.

The GUI of Figure 2(a) lets the user explicitly introduce the problems to be processed. The GUI of Figure 2(b) provides the means to request the desirable content and representation for the problems. In particular, there are options to present the problems with their results, with intermediate steps, with the carries produced during the computations, and a currency sign. The addition sign may be placed in the same columns as the currency sign, or leftward. The output may be exported in LaTeX, MathML, text, or Braille format.

A user may request randomly generated problems satisfying desirable constraints via the GUI shown in Figure 2(c). The data to be included refers to the number of problems to be generated, the desirable number of decimal digits, lower and upper limits on the number of desirable operands, the total number of digits per operand, and the magnitudes of the digits. Figure 2(d) shows a GUI for determining the characteristics involved in typesetting the problems within the worksheets.

3 System architecture

Two aspects make the system highly configurable: a modular architecture which gives independent attention to the different functions of the system,

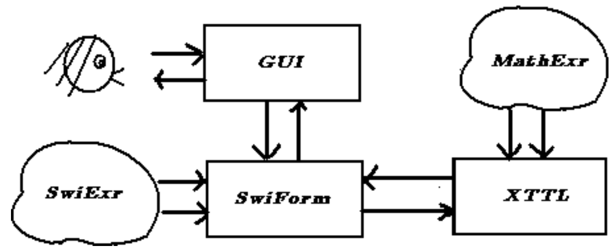

Figure 3: The underlying structure of the system.

and a script-based approach which provides for the functions to be described within external configuration files instead of being hardwired into the code. The system consists of four major components (Figure 3).

The back end of the system employs a script-based driver, called XTTL, for performing XML transformations. The driver is offered a library, called MathExr, of scripts for typesetting exercises and worksheets. In addition, the library includes a utility able to compute data for the exercises. The scripts for handling the different operations are independent of one another, and they can be supplemented and augmented by additional scripts to achieve alternative outcomes. The examples of Figure 1 were obtained with those scripts.

The front end of the system uses a script-based driver, called SwiForm, for managing the GUIs. The driver is built as an extension to the SwiXml system [10] and is offered a library, called SwiExr, of scripts that specify the desirable features for the GUIs. In addition, the library provides scripts and a utility for binding the front end to the back end, and for filtering the data communicated between these ends.

The MathExr scripts are currently tailored to receive the requests in XML format, and to deliver the exercises in Braille, HTML, LaTeX, MathML, and plain text formats. The type of the information being transmitted is fully dependent on MathExr, and that makes it possible to substitute a given graphical front end for another. A LaTeX front end is of interest for us here.

4 An XML view

The MathExr component takes as input a high level description of the desirable outcome and processes it into a detailed account of how the digits, symbols, and rulers of the exercises are to be placed on a grid. The input of MathExr in its XML forms is to a large degree a mirror of the visual format available from the GUIs.

```
<addition xttl="add-tex.xttl"
          carry="yes"
          steps="yes"
          result="yes"
          currency="alone">
  <operands>
    <operand>1</operand>
    <operand>23</operand>
  </operands>
</addition>
```

(a)

```
<multiplication steps="yes"
                result="yes"
                currency="alone"
                xttl="mult-txt.xttl">
  <operands>
    <operand>1234</operand>
    <operand>5</operand>
  </operands>
</multiplication>
```

(b)

```
<division steps="no"
          result="yes"
          remainder="yes"
          currency="yes"
          rulers="full"
          xttl="div-brl.xttl" >
  <operands>
    <operand>1234</operand>
    <operand>56</operand>
  </operands>
</division>
```

(c)

```
<generate xttl="worksheet-generate-tex"
          cols="3"
          enumerate="decimal"
          enumerateBefore="("
          enumerateAfter=")"
          enumerateStart="1">
  <exr type="addition"
       howmany="2"
       xttl="add-tex"
       carry="yes"
       steps="yes"
       result="yes"
       currency="alone">
    <minEntries>2</minEntries>
    <maxEntries>7</maxEntries>
    <low>100</low>
    <high>999</high>
    <minDigits>1</minDigits>
    <maxDigits>3</maxDigits>
    <decimals>3</decimals>
  </exr>
</generate>
```

(d)

Figure 4: XML requests for typesetting three types of problems and a worksheet.

Figure 4 contains examples of requests in XML for single exercises and for a worksheet of exercises. Each of the examples states the data required for deriving the outcome and the name of the script to be applied on the data. For instance, the first example asks for the script stored in a file named `add-tex.xttl`. That script typesets the addition problem onto a grid and exports the result in a tabular format acceptable to LaTeX.

5 A LaTeX perspective

The motivation to develop our tool was rooted in the desire to provide the means to prepare spatial arrangements of math exercises in Braille. The Braille exercises are to serve mainly blind students attending mainstream classes populated mostly by sighted children. To properly serve the blind students, it is crucial for the Braille exercises to accurately represent the problems given to the sighted students. Consequently, we programmed our tool to support standard print formats as well as Braille, with the hope of making it a useful tool for preparing worksheets in any format. In this way, we can ensure that the Braille material is entirely equivalent to the material in print formats.

LaTeX offers to our tool an obvious method of producing high quality printouts, through style files that are easy to tailor and modify. In addition, LaTeX also provides our tool with an option for a natural text-based user interface to request the exercises. A front end of this kind obviously enables authors to seamlessly incorporate exercises and worksheets into documents, and introduces a friendly authoring environment for those who shy away from WYSIWYG platforms. In particular, a non-WYSIWYG front end for accessing the tool is obviously crucial to blind users.

A LaTeX style file to support the input environment can consist of macros that imitate the high level structures captured by the XML code presented to MathExr. For instance, the following LaTeX-oriented command expresses the same request as the XML code of Figure 4(a).

```
\mathexr[op="addition"
         xttl="add"
         carry="yes"
         steps="yes"
         result="yes"
         currency="alone"]{1,23}
```

The implementation of the macros can be quite straightforward, relying on a process similar to that for bibliographies and indexes in LaTeX. Specifically, the arguments given to the macros may be written into a file in XML format, with the expectation that the file will be processed by MathExr. The exercises produced by MathExr can then be imported by the macros into the LaTeX source in consecutive compilations.

6 Connecting the dots

We introduced our tool to the public domain at the TUG 2005 International Typesetting Conference with the hope of promoting the tool with support from the LaTeX community. In particular, we looked for potential users and developers, interested in LaTeX interfaces and typesetting criteria for elementary math exercises. In addition, we wished to connect with potential developers for Braille and related proofreading fonts.

Despite the central role LaTeX plays in documenting mathematical and scientific content, until recently the only public support available for elementary math exercises was a single macro by Barbara Beeton and Donald Arseneau [3]. This macro uses input interfaces such as the macro invocation `\longdiv{12345}{13}` and produces output like that shown in Figure 5.

```
        949
13 ) 12345
     11700
       645
       520
       125
       117
         8
```

Figure 5: A display by `\longdiv`.

The situation has changed recently, as a new LaTeX package for typesetting spatial math exercises, named Xlop, was submitted to CTAN [5]. In spirit, the new package answers our wish for a LaTeX counterpart to our tool. We hope future work on Xlop and our tool will enrich both utilities with additional features, and will bridge the differences between the two utilities to allow to support each other. In particular, we would like to see Xlop providing a LaTeX extension to our system, and have our utility offer alternative output formats to Xlop in general and to include Braille in particular.

We are not aware of any LaTeX fonts for Braille. A substitute through LaTeX pictures is available via a style file [9]. The issue of backward translating math exercises in Braille to LaTeX got some attention in an experimental system called INSIGHT [1].

7 Acknowledgment

We are very grateful to Susan Jolly for introducing us to Braille and for her generous guidance and help. Thanks also to Barbara Beeton and Karl Berry for editing the manuscript.

References

[1] N. Annamalai, D. Gopal, G. Gupta, A. Karshmer, and H. Guo, *INSIGHT: A comprehensive system for translating Braille-based mathematical documents to LaTeX*, in Proceedings of the International Conference on Human Computer Interaction, pp. 1245–1249. 2003. (Demos available at `http://www.logicalsoft.net/frame/demo.htm`.)

[2] N. Bagchee and E. Gurari, *SwiExr*, `http://www.cse.ohio-state.edu/~gurari/mathexr/`.

[3] B. Beeton, D. Arseneau, *Long division*, *TUGboat* 18(2), June 1997, p. 75–76, `http://tug.org/TUGboat/Articles/tb18-2/tb55works.pdf`.

[4] Braille Authority of North America (BANA), `http://www.brailleauthority.org/`.

[5] J. Charpentier, *Xlop*, 28 April 2005, `http://www.ctan.org/tex-archive/macros/generic/xlop`.

[6] Computers To Help People, Inc. (CHPI), Sponsoring Technical Reference Books and Manuals, `http://www.chpi.org/refspons.htm`.

[7] R. Craig, *Learning the Nemeth Braille Code: A Manual for Teachers and Students*, American Printing House for the Blind, 1987.

[8] A. Nemeth, *The Nemeth Code of Braille Mathematics and Science Notation*. The Braille Authority of North America (BANA), American Printing house for the

Blind, 1972 revision. (French version: http://www.meq.gouv.qc.ca/ens-sup/ens-coll/Braille/Code_Braille_mathematique.pdf)

[9] W. Park, *LaTeX 2ε package for typesetting Braille*, April 1999, http://www.ctan.org/tex-archive/macros/latex/contrib/braille/braille.html.

[10] W. Paulus, *SwiXml*, http://www.swixml.org/.

[11] *Texas Partnership for Increasing Braille Production*, Report of Braille Production Specialist Focus Group Meeting, January 2000, http://www.tsbvi.edu/textbooks/afb/texas-transcriber.htm.