Braille Math Extension to RoboBraille

A Universal Software Solution for Converting Math into Braille

Vlad Paul Cosma, Tanja Stevns, and Lars Ballieu Christensen(✉)

Sensus ApS, Hillerød, Denmark
{paul, tanja, lars}@sensus.dk

Abstract. While sighted mathematicians have long accepted the need for a uniform way of writing math, the situation amongst the blind is different: Rather than standardising, the tradition has been to develop language-, country- or even institution-specific codes for expressing math in Braille in addition to the national Braille codes for literary Braille. Using the RoboBraille service as a foundation, the paper describes an automated and flexible system that can shred, transcribe and reassemble documents containing math into Braille. Using existing components, the Braille Math extension to RoboBraille enables Braille readers to mix and match amongst supported national Braille codes and Braille math regimes, including Braille math systems like Nemeth and Marburg, and mainstream formats such as LaTeX and MathML. Finally, the paper discusses how the system may be enhanced to handle charts, graphs and tables.

Keywords: Braille math · Inclusive education for all · Automated transcription · RoboBraille · UMCL

1 Introduction

Math is important in mainstream education as it is an integral part of many subjects (math, biology, sociology, economics, and linguistics). As blind students are included in mainstream education, adequate transcription systems are required that can convert educational material containing mathematical notation into Braille in a timely, cost-efficient and easy way. Sighted mathematicians have long accepted the need for a unified way of expressing math. Using a common set of universally agreed mathematical symbols [1], this enables mathematicians from different cultures and countries to learn, exchange and discuss mathematical problems.

2 The Braille Math Challenge

In order to present math in Braille, the two-dimensional notation used by the sighted needs to be transcribed into a sequence of Braille characters. Although it has proven difficult to teach math to blind students and costly to transcribe textbook material in Braille, the approach in many countries has been to develop language-, country- or

K. Miesenberger et al. (Eds.): ICCHP 2016, Part I, LNCS 9758, pp. 15–18, 2016.
DOI: 10.1007/978-3-319-41264-1_2

even institution-specific Braille codes for math. Examples include Antoine (France), Marburg (Germany and others), as well as the British Math Braille code used in the UK and many commonwealth countries, the Nemeth code used in the US and many places in the Middle East and Asia, and many others. Some educational institutions experiment with using mainstream formats such as LaTeX and MathML directly.

For the text parts of the educational material, the situation is similar: The literary Braille codes differ from country to country and from language to language, with significant difference in how Braille signs are used to represent contractions, national characters, punctuation signs and more. Even though math Braille notation could likely benefit from standardisation (similar to the standardisation of music Braille), standardisation and international acceptance of a uniform way of expressing math in Braille is unlikely to happen in the foreseeable future.

3 Bringing Fragmented Solutions Together

To accommodate the blind, systems are required that allow to select between different national Braille codes and Braille math notation systems with minimum hassle.

Several solutions exist for transcribing math into Braille, including MathInBraille [2], LaBraDoor [3], BraMaNet [4], Math2Braille [5], Winsight [6] and many more.

The Braille Math extension to RoboBraille [7] consolidates several Braille transcription and math conversion libraries (Sensus SB4 [8], LibLouis [9], UMCL [10], fMath [11]) into a single solution, enabling users to mix and match amongst supported Braille codes and Braille math regimes.

4 The Braille Math Extension to RoboBraille (BMER)

The objective of the BMER was to create a system capable of automatically transcribing a digital representation of a mathematical schoolbook created for the sighted into a Braille book. As such, the BMER provides a unified interface to a range of efficient conversion tools specific for each document element (text, math, tables, graphics), under a single, accessible and user friendly interface that is universally available to students and others. Likely end-users are blind students and teachers that need books containing math equations converted into Braille.

The BMER comprises a set of programs and libraries designed to work together under a single web application. The BMER is based on a robust, vertically layered architecture, that is easily extendable to support the addition of new components. The BMER consists of three layers: 1. The Presentation Layer (top layer), the web application, handles the interaction between the user and the software. 2. The Domain Logic layer (middle layer) is responsible for transporting and manipulating data. And 3. The Data Source layer (bottom layer) contains the individual conversion libraries. They achieve the simple conversion tasks of transcribing text to literary Braille code, MathML to math Braille code or shapes to tactile graphics.

Once submitted for conversion, the document goes through three stages: (1) preparation; (2) processing; and (3) formatting. Stage 1 shreds the document into its basic elements (text, math, illustrations), while at the same time maintaining its structure. Stage 2 sends each element, along with relevant input parameters, to the appropriate data source layer library and retrieves the converted result. For consistency reasons, all results are represented within the unicode Braille range. Finally, stage 3 brings the elements back together to a single document and applies the final formatting, such as pagination, page numbering, conversion into the Portable Embosser Format (PEF) [12] or local character set transposition. Each mathematical equation is written on a new line, in order to distinguish text from math. For "Simplified LaTeX" conversions, the dollar symbol marks when an equation begins and ends, e.g., $ c − 1 = b $.

The domain logic layer constitutes the core implementation and functionality of the BMER. The layer is split into mechanisms for transportation and manipulation of data. Furthermore, it is responsible for linking the data source layer with the presentation layer. In order to achieve this, each of the four underlying libraries are encapsulated into their own unit of work that exposes their functionality through a clearly defined point of entry. Towards the presentation layer, communication is implemented with remote procedure calls to ensure extensibility and reliability.

Initially four libraries were selected to achieve the text and math conversions. Sensus SB4 and LibLouis are used for literary Braille conversions of text, while fMath and UMCL handle conversions from MathML to a variety of math Braille codes. fMath is used to convert MathML to LaTeX. UMCL is used to convert MathML into Braille math in Nemeth, Marburg, British, French, Italian and Czech Braille math codes.

5 Evaluation and Future Work

Following development and extensive testing, a few conclusions can be made: Mapping MathML constructs into their LaTeX representations appear accurate. The conversion of MathML into Nemeth, Marburg, British, French, Italian and Czech is limited due to the confines of the underlying UMCL as discussed in [13]; this limits the usefulness to cover only the curriculum until and including upper secondary level as concepts such as matrices, integrals, trigonometric functions or limits are not support-ed. Following final testing, the BMER is expected to be made publicly available as part of the RoboBraille service during the second half of 2016.

The resulting Braille text document reflects the structure of the source document. Reliability is assured by marking failed conversions in the resulting document, so that they may be resolved at a later time by manual intervention. The overall performance of the solution is linear to the size of the input file, which is a limitation of the underlying libraries. Utilizing messaging queues in a cloud-based architecture, multiple conversions can be handled in parallel.

During the development of the Braille Math extension to RoboBraille, several ide-as for improvements were identified: Support for scanned documents: The current implementation relies on documents in Microsoft Word with math equations composed using an equation editor. Charts and graphs: The current implementation does not support illustrations. As math material typically consists of text, mathematical notation as well as graphs and charts, a future solution would benefit from being able to export illustrations into documents that can be rendered as tactile graphics, 3D renditions or similar. Tables: The current implementation is capable of extracting table information from the document. A future solution could benefit from being able to map table data into a linearized Braille form.

References

1. ISO 80000-2:2009 Quantities and units - Part 2: Mathematical signs and symbols to be used in the natural sciences and technology. ISO (2009)
2. Miesenberger, K., Batusic, M., Heumader, P., Stöger, B.: MathInBraille online converter. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) ICCHP 2012, Part I. LNCS, vol. 7382, pp. 196–203. Springer, Heidelberg (2012)
3. Batusic, M., Miesenberger, K., Stöger, B.: Labradoor, a contribution to making mathematics accessible for the blind. In: Edwards, A., Arato, A., Zagler, W. (eds.) ICCHP 1998. Springer, Berlin (1998)
4. BraMaNet: http://handy.univ-lyon1.fr/MH/bramanet/bramanet.php
5. Crombie, D., Lenoir, R., McKenzie, N.R., Barker, A.: math2braille: opening access to mathematics. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 670–677. Springer, Heidelberg (2004)
6. Gopal, D., Wang, Q., Gupta, G., Chitnis, S., Guo, H., Karshmer, Arthur I.: Winsight: towards completely automatic backtranslation of Nemeth code. In: Stephanidis, C. (ed.) HCI 2007. LNCS, vol. 4556, pp. 309–318. Springer, Heidelberg (2007)
7. Christensen, L.B.: RoboBraille – automated braille translation by means of an e-mail robot. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 1102–1109. Springer, Heidelberg (2006)
8. Christensen, L.B.: Multilingual two-way braille translation. In: Klaus, J., et al. (eds.) Interdisciplinary Aspects on Computers Helping People with Special Needs, Österreichische Computer Gesellschaft/R. Oldenbourg Wien München (1996)
9. LibLouis: http://www.liblouis.org
10. Archambault, D., et al.: UMCL: https://sourceforge.net/projects/umcl
11. fMath: http://www.fmath.info
12. Portable Embosser Format (PEF): http://pef-format.org
13. Jamar, M.: Conversion of Mathematical Documents into Braille. Masaryk University, Faculty of Informatics (2012)

http://www.springer.com/978-3-319-41263-4

Computers Helping People with Special Needs
15th International Conference, ICCHP 2016, Linz, Austria, July 13-15, 2016, Proceedings, Part I
Miesenberger, K.; Bühler, C.; Penaz, P. (Eds.)
2016, XXIX, 580 p. 148 illus., Softcover
ISBN: 978-3-319-41263-4