Liblouis – a universal solution for Braille transcription services

Christian Egli
christian.egli@sbszh.ch

05 October 2009

Contents

1 Abstract

Software support for the generation of good Braille has historically been hampered by fragmented and small markets. Some languages have good commer-

cial support, others are lacking. Fortunately Liblouis, an open source tool for complete Braille transcription services, is emerging as a universal solution. Liblouis provides Braille translation for literary and computer Braille, offers support for contracted and uncontracted translation for many languages and includes support for Braille mathematics codes such as Nemeth. Liblouis also provides Braille formatting, which can handle many document formats including DTBook XML. Both the translation and the formatting can easily be adapted to new languages and document formats.

This paper shows how Liblouis will be used at the Swiss Library for the Blind and Visually Impaired to integrate Braille generation into a production workflow based around DTBook XML.

2 Introduction

Producing good Braille is not a trivial process. In particular the production of contracted Braille is difficult as the rules are sometimes ambiguous and sometimes originate from a time when all the Braille transcription was done manually (as explained in [15]). There are a number of good software solutions for Braille translation available that support a producer in this task. However, the viability of these tools is limited as the translation rules differ from language to language, i. e. the market for Braille transcription tools is not only limited to Braille users, but further fragmented by differing rules for each language. This makes it hard for any commercial entity to make a viable business by producing Braille translation software for languages which are not mainstream. Consequently many users are left in an unfortunate situation where they have outdated, inadequate or even no support for their language.

Fortunately for us, there is a silver lining: Liblouis[6] is an open source software solution for Braille transcription. It is freely available and runs on Windows, Mac OSX and any Unix operating system. Liblouis has support for many main- and not so mainstream languages and it can transcribe many document formats to Braille such as DTBook XML[12]. Last but not least, the free availability of the source code has attracted an enthusiastic developer community which ensures that Liblouis continues to grow and improve.

Support for different languages is driven through so called translation tables, a powerful and yet easy means to define the translation rules. We will look at how to adapt the language support in Section 5.1. Support for a new document format is driven trough so called semantic action files. We will cover these in Section 5.2. Finally Liblouis is designed as a library to be used in other programs such as production systems or even the Daisy Pipeline[3]. Section 5.4 outlines how Liblouis has been embedded for use in other programming languages such as Python or Java.

We will look at how Liblouis compares with other solutions and in Section

7 provide an outlook as to what can be expected from Liblouis in the future and how you can get involved.

3 The problem

At the Swiss Library for the Blind and Visually Impaired the process of transcribing Braille is based on fairly old tools that have a number of drawbacks.

- They are hard to maintain and it is challenging to bring them up-to-date with new requirements. We would like to move the production to a process based around DTBook XML.
- A lot of internal effort has gone into these tools and there is very limited collaboration or synergies with other organizations. We would like to move away from in-house developments only to pool our resources with other Braille producing organizations and maintain the required software in an open and collaborative fashion.

Presumably this situation is similar in other institutions.

4 A universal solution

Ideally what we need at the Swiss Library for the Blind and Visually Impaired is a software solution for Braille transcription which is open source, universal (in that it supports many languages), can easily be adapted and can handle the DTBook XML format. In addition, it should run on our preferred platform.

Open source To ensure the longevity of a project we have to make sure it is not dependent on one person or one institution. This is why we want a project which is developed collaboratively and the code is free to study, use, modify and share.

Easily adapted Since this software is to be used by many different institutions, there is a need to adapt it to meet different requirements. This means that Braille translation and Braille formatting have to be configurable outside of the code proper in easy to understand configuration files.

Support for many languages In order to be a universal solution, i. e. for this software to be usable by many organizations, it has to support Braille transcription for as many languages as possible out of the box.

Support for DTBook XML format Since we are moving to a process where Braille transcription is based around DTBook XML, we need the tools to support this format.

A bit of research will quickly lead to Liblouis and Liblouisxml[7] as the ideal candidates for a software solution to handle all our requirements. Liblouis handles the Braille translation while its companion Liblouisxml handles the Braille formatting.

Liblouis is Open Source Liblouis is freely available from its Google code web site[6] and is released under the terms of the GNU Lesser Public License (LGPL)[5]. It is written in C and runs on Windows, Mac OSX and any Unix.

Liblouis is easily adapted In Liblouis the translation of Braille is driven through text based translation tables that define the translation rules for a specific language in an easy and intuitive way. The formatting of Braille is defined in semantic mappings that define how a specific (XML) input tag is to be rendered in the Braille output. Additionally Liblouis can be used as a library, i.e. it can be embedded into existing applications or frameworks such as the Daisy Pipeline.

Liblouis supports many languages At present Liblouis supports Braille transcription (contracted and uncontracted) for over 40 languages. Support for new languages can easily be added and in fact new translation tables are contributed on a regular bases.

Liblouis supports DTBook XML In combination with Liblouisxml, Liblouis supports formatting of many document formats such as DTBook XML, XHTML, Docbook[4] or Microsoft Word XML.

5 Liblouis in detail

The process of Braille transcription involves both translating and formatting. Liblouis and its companion Liblouisxml offer both services in a stacked approach as shown in Figure 1. Liblouis provides Braille translation functionality by using translation tables that define the translation rules. Liblouisxml is built on top of Liblouis and uses the translation services of Liblouis to provide Braille formatting. The formatting is configured by formatting definitions. Liblouisxml takes XML input and renders it as formatted Braille.

5.1 Braille translation

The Braille translation process in Liblouis is entirely driven through translation tables which define the rules for the translation in easy to understand syntax. To generate uncontracted Braille they simply define a mapping between print character and the corresponding single-cell or multi-cell Braille symbol. In addition, the translation tables can also define rules for replacing

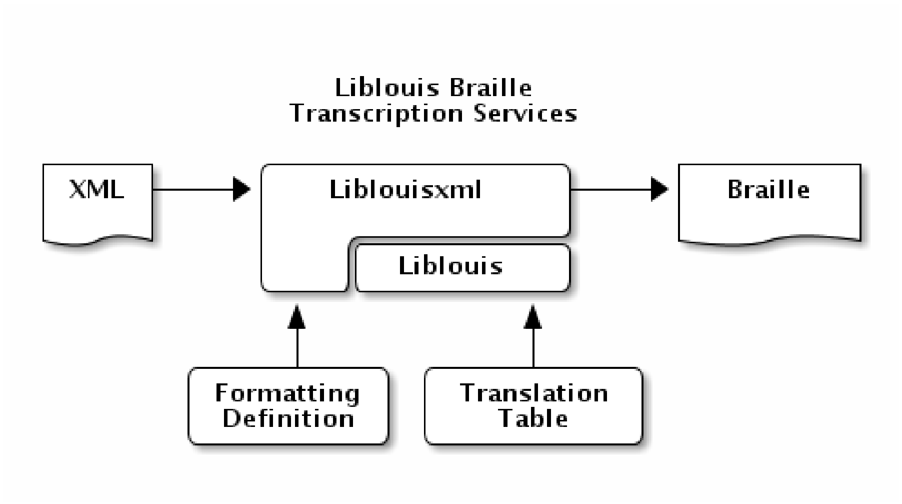

Figure 1: Architectural overview of Liblouis and Liblouisxml

common print words and other common sequences of print letters with special Braille symbols known as contractions. These are used for contracted Braille.

5.1.1 Uncontracted Braille

The automatic generation of uncontracted Braille is fairly straightforward. The translation table defines the basis for this translation. By way of example the following excerpt from the translation table for US English grade 1 defines multi-cell Braille symbols for signs. The command `sign` (an *opcode* in Liblouis terminology) defines a mapping between the first operand (in this case a Unicode print character) and the second operand (the Braille dots):

```
# mappings for signs
sign \x00A9 45-14                # COPYRIGHT SIGN
sign \x00AE 45-1235              # REGISTERED TRADE MARK SIGN
```

Mappings for punctuation and math are done in a similar fashion:

```
# punctuation
prepunc ( 2356
postpunc ) 2356
prepunc " 236
postpunc " 356

# mappings for math
math = 123456
math < 126
```

Braille indicators are marked with special opcodes. For example, the dot pattern which indicates capitalization of a single letter in English is dot `6`. Similarly, the dot pattern which begins a block of capital letters is `6-6`. The following example illustrates these and more definitions:

```
# Braille indicators
capsign 6
begcaps 6-6
endcaps 6-3
numsign 3456
```

5.1.2 Contracted Braille

Contracted Braille extends uncontracted Braille's replacement rules for print characters with additional rules for replacing common print words and other common sequences of print letters. The special braille symbols used to achieve these replacements are known as contractions. There are two main approaches to automate this translation: rule-based and dictionary-based. The rule-based approach tries to translate the replacement rules into computer logic while the dictionary-based approach simply has an extensive dictionary in which the Braille symbols for any print word are defined. Both approaches are discussed at great length in [15]. Liblouis simply avoids this debate and allows us to use either approach or even a combination of both in a translation table for translating print to contracted Braille.

In the following example for German grade 2 we define that the word "aber" is to be contracted to one Braille cell (`1`). However if "aber" is part of a word and is at the beginning then it is to be rendered as `2-1`. But the string "aberkenn" always should be translated as defined below:

```
word aber 1
begword aber 2-1
always aberkenn 1-12-12456-13-14-1345
```

There are many more instructions to allow for the definition of replacement rules such as opcodes for syllable handling, for print characters at the beginning, middle or end of a word, and so forth.

To implement a dictionary-based translation you simply add all the words and their corresponding Braille symbols in a translation table as in the following example (excerpt):

```
word aachen 1-1-1456-14
word aachenbesuch 1-1-1456-14-23-234-136-1456
word aachenbesuche 1-1-1456-14-23-234-136-1456-15
word aachenbesuchen 1-1-1456-14-23-234-136-1456-14
word aachenbesucher 1-1-1456-14-23-234-136-1456-12456
```

5.2 Braille formatting

Braille formatting is handled by Liblouisxml. It translates an XML or a text file into an embosser-ready Braille file. Many document formats, such as DTBook XML, XHTML, Docbook or Microsoft Word XML, are supported out of the box.

The formatting can be configured through so called semantic action files where there is a mapping between XML tags and a formatting specification. For example, if a `<h1>` is encountered in the XML file it should be rendered using the `heading1` style and if a `<p>` is encountered it should be rendered using the `para` style. The following is an extract from the semantic file for DTBook XML which illustrates this:

```
heading1 h1
heading2 h2
para p
```

The characteristics of styles like `heading1` and `para` can be defined separately as to how they should be rendered in the Braille output. For example, the `heading1` style is defined as follows:

```
style heading1
        linesBefore 1
        format centered
        linesAfter 1
```

5.3 Liblouis in action

Thus, for example, in order to transcribe the book on Valentin Haüy[16] (which is a sample DTBook XML file that comes with the Daisy Pipeline) we simply use the following command:

```
xml2brl -f liblouis.cfg hauy-2005-1-short.xml hauy-2005-1-short.brl
```

The command specifies a configuration file and an input and output file. The configuration defines settings needed for the transcription, such as line length, input encoding and translation table.

Here is a small excerpt of the resulting output; notice, for example, how headlines are handled:

```
           #c4 ,val5t9 ,ha`uy


    #c.a ,9troduc;n
  ,special $uc,n =! d1f1 ! visu,y
h&icapp$ &! 4a#d has xs roots 9 ! ,fr.e
```

```
(! ,5li<t5;t1 a p}iod ( gr{+ 9t}e/ 9 !
cre,n ( organis$ $uc,n =! h&icapp$4 ,9
#agfj1 ! a2‘e l’‘ep‘ee f.d$ ! f/ s*ool
=! d1f1 ": 8 $uc,nal me?ods 7 bas$ on !
t1*+ ( sign language
```

5.4 Other uses of Liblouis

Since Liblouis and also Liblouisxml are libraries they can be embedded into other tools and frameworks. Bindings for Python and Java have been contributed, i. e. Liblouis can be used with multiple programming languages. The Python bindings have been used to integrate Liblouis into NVDA[8], the NonVisual Desktop Access, a free and open source screen reader for the Microsoft Windows operating system. The GNOME screen reader Orca[9] also uses Liblouis.

ViewPlus[14], who have provided the resources that made the development of Liblouis possible, are using Liblouis and Liblouisxml in their TSS[13] software for translating both text and math.

Bookshare.org[2] whose mission it is to make the world of print accessible to people with disabilities, is using Liblouis to generate Braille[1].

The Swiss Library for the Blind and Visually Impaired plans to base its future Braille production on Liblouis as well. A translation table for German grade 2 is in preparation at the time of writing.

5.5 Liblouis ecosystem

While Liblouis has a host of impressive features, it is more than just a list of supported items to be checked off in a marketing flyer. Probably the most compelling quality of Liblouis is its development community. While the core developer team keeps the ship on course, there is a constant influx of new people who continue to contribute translation tables for new languages, bindings for other computer languages, bug fixes and user input. Within the last 6 months we have seen the addition of support for about 10 more languages, the contribution of Python and Java bindings and numerous smaller enhancements.

The source can be checked out from the Google code web site and discussions happen on the project mailing list. There are binaries for Windows and many Linux distributions are starting to include and distribute Liblouis and Liblouisxml.

[1]https://wiki.benetech.org/display/BQU/Dead+Heat

6 Related work

There are a number of related efforts to create an open source solution for Braille transcription. One interesting project is Autobraille[1]. It consists of the Braille translation engine RoboBraille/SB4 developed by Sensus ApS end Synscenter Refsnaes[11] and a Braille formatter developed by the Danish National Library for the Blind. Autobraille aims to automate the production of Braille books in contracted Braille and handles DTBook XML documents. Unfortunately, at the time of writing, only the formatter is open source while the translation engine remains proprietary. This precludes the user from modifying the translation tables and leaves small language communities with the same problems outlined in Section 2.

Another interesting development is PEF[10], the Portable Embosser Format, a data format for representing Braille accurately and unambiguously. The Daisy Pipeline has support for PEF built-in. Presently Liblouis does not have support for generating PEF as output format, but this would certainly be an interesting option for the future.

7 Conclusions and further work

Liblouis is a universal solution for Braille transcription services and can meet many if not all your Braille transcription needs. It is used in production at Bookshare.org and will be used at the Swiss Library for the Blind and Visually Impaired. It is freely available from http://code.google.com/p/liblouis/.

This is not to say that Liblouis is perfect. Our tests at Swiss Library for the Blind and Visually Impaired have not covered all of the DTBook XML tags such as tables or math for example. While there has been quite a bit of work put into math braille lately, it is quite likely that handling of tables will need improvement. Also, there is only skeletal support for SVG graphics, chemistry and music, which are all areas waiting for some developer time to be completed.

Fortunately, if something about Liblouis is not how you had envisioned it you are free to change and adapt it to your needs, discuss your changes on the mailing list and possibly even have your contribution included in Liblouis. The developers always welcome feedback, be it either a problem report for a translation table, a fix for a problem in a translation table or even an all new translation table for a language that is not yet supported.

References

[1] AutoBraille – a project with The Danish National Library for the Blind. `http://www.robobraille.org/rb/subpage238.aspx`.

[2] Bookshare - Accessible Books and Periodicals for Readers with Print Disabilities. `http://www.bookshare.org/`.

[3] DAISY Pipeline. `http://www.daisy.org/projects/pipeline/`.

[4] Docbook. `http://www.docbook.org/`.

[5] GNU Lesser General Public License. `http://www.gnu.org/copyleft/lesser.html`.

[6] Liblouis, A Braille translation and back-translation library. `http://code.google.com/p/liblouis/`.

[7] Liblouisxml, A Braille transcription software for xml documents. `http://code.google.com/p/liblouisxml/`.

[8] NonVisual Desktop Access (NVDA). `http://www.nvda-project.org`.

[9] Orca, a free, open source scriptable screen reader. `http://live.gnome.org/Orca`.

[10] Portable Embosser Format (PEF). `http://www.daisy.org/projects/braille/braille_workarea/pef.php`.

[11] Sensus ApS. `http://www.sensus.dk/sensus/frontpage39.aspx`.

[12] The ANSI/NISO Z39.86 Specifications for the Digital Talking Book. `http://www.daisy.org/z3986/`.

[13] Tiger Software Suite. `http://www.viewplus.com/products/touch-audio-learning/TSS/`.

[14] ViewPlus Technologies. `http://www.viewplus.com/`.

[15] Susan Jolly. Current Issues for Automated Conversion of Print to Braille: The Braille-in-DAISY Project. Technical report, dotless-braille.org, April 2008. Also available as `http://www.dotlessbraille.org/bidissues.htm`.

[16] Beatrice Christensen Sköld. *Valentin Haüy - the father of the education for the blind*. Swedish Library of Talking Books and Braille (TPB), 2006.