

Louis Braille Celebration

Braille Becomes Electric: The Trials and Triumphs of Braille Translation Software

Joe Sullivan

When a person converts the text of a print document to braille, it is usually called *transcription*. When a computer program does the same thing, it is often called *translation*. Some people think translation is an ill-chosen term, since the term usually connotes changing text from one natural language to another, whereas braille—thanks to the genius of Louis Braille’s original design—is not a separate language from print but rather an alternate way of writing the same text. There are a number of reasons described in this essay why early braille translation software came to use the term *translation*, and Louis Braille himself could not have imagined the ongoing controversies and never-ending intricacies of translation when computers were first used to produce braille from print and vice versa.

EARLY INNOVATIONS

The first software for braille translation was developed in the 1960s, well over a century after Louis Braille first developed his code. The in-house program developed for the IBM 704 computer by the American Printing House in Louisville, Kentucky, progressed in its development through several research projects until the first commercial braille translation product for literary braille appeared in 1975 (Sullivan, 2008). In those early days, the hardware that was typically available for running such application programs afforded vastly less computing

power than we enjoy, and often take for granted, today. At the time the translation program was being developed, it was not clear that the available computing power would be sufficient to create software that could implement the trickier rules of English Braille, American Edition (EBAE) with an acceptable degree of accuracy.

Although some artificial intelligence researchers believed automated natural language translation, such as from Russian to English, was just another grant and a few years away, individuals who were familiar with the intricacies of literary braille knew that its code was written with human transcribers in mind, and understood that teaching a computer to anticipate and apply correctly the many rules referring to the sound or meaning of the text to be transcribed would be challenging to say the least. For example, could a computer be taught to decide if “do” in a text refers to the verb denoting action or noun indicating the musical note? EBAE requires that the two “do”s be brailled differently in contracted braille, the most widely used form of English braille in which many words and letter groups are written in a shortened form. In contracted braille, “do” the verb is reduced to the letter “d.”

Despite the concern that computers cannot directly discern sound or meaning when translating braille, the need for at least partial automation was increasing rapidly in the 1970s. The trend of enrolling children who are blind in mainstream public schools, such as was seen in Atlanta around this time, created a need for rapid production of informal classroom materials in braille, which existing braille printers could not meet in a timely fashion. In addition, braille printing houses and braille producers required skilled braille transcribers to oversee the production process, and social and economic trends combined to make such transcribers ever harder to find. Under such conditions, the speed of braille production was of the greatest importance, and a low level of error was tolerated.

The braille translation software developed in the 1970s was able to accurately translate braille in an expedient manner. The deeply intractable

The guest editor of the *JVIB* Louis Braille Bicentennial Celebration is Susan Jay Spungin, Ed.D., consultant and retired vice president for International Programs and Special Projects, American Foundation for the Blind.

problem situations, subjects of considerable debate, turned out to be statistically infrequent. For example, “do” very seldom refers to the musical note except in text that is actually about music. The braille translation software, thus, was programmed to assume that “do” always referred to action. Exceptions to such rules needed to be handled by humans, who marked the text prior to translation or proofread it afterwards. Although human intervention is required for accuracy, especially in texts translated for young braille readers or examinations, such work can be done more quickly than if the entire transcription was performed manually. And, for less critical texts, when the time and effort required for such human intervention may be impractical, it was determined that an experienced braille reader was no more likely to be confused by a minor translation error than a newspaper reader would be thrown off by an occasional typographical error.

EXCEPTIONS TO EVERY RULE

Because the need for human intervention to make judgments about sound or meaning continued to be recognized as a drag on productivity in braille translation, an interesting, ongoing, and often controversial movement began to gain momentum. Sometime in the late 1970s, Dick Evensen, a braille user and then chair of the Braille Authority of North America’s (BANA) literary braille committee, asked me, as a developer of braille translation software, which braille rules were most troublesome. I put the “natural pause” provision at the head of the list. The “natural pause” was described in Rule 41 of the 1972 edition of EBAE, which said, in part: “There should be no space between the lower-sign contractions *to*, *into*, and *by*, and the word which follows if there is no natural pause between them. If in doubt about the pause, they should be joined. . . .” Under this rule, a “natural pause” required a space in phrases (for example, “to and fro”), but also in some general cases that required human interpretation of meaning (“What trouble have you gotten into this

time?”). Somewhat to my surprise, Dick Evensen told me the matter of natural pauses could easily be addressed, and the next update to EBAE prepared by his committee and eventually published by BANA in April 1980, did not include the “natural pause” requirement. “To” was joined to “and,” and “into” to “this,” in the foregoing examples.

Not everyone was happy with this outcome. Some transcriber groups felt that omitting the space between words that are not in the same grammatical phrase would be confusing to readers, and vowed to continue to observe the old rule. Readers, however, accepted the new rule, and within a few years the “natural pause” requirement was dropped from the British code as well.

The case of the “natural pause” was the first example of a now-familiar pattern among braille stakeholders: A change to a braille rule is proposed, motivated in part by braille translation issues, and controversy ensues, sometimes lasting many years as opponents seek either to prevent change or to bring about some other, inconsistent kind of change. Such controversies are normal in any human undertaking, and disputes are healthy to the extent that there are often good points on either side of a debate. However, the objection that proposals to change rules are designed only “for computers” and not for braille readers is unhelpful, since the point of seeking greater efficiency in braille translation is not to provide health and comfort to an electronic device, but rather to make it easier to produce braille and, hence, make braille more available in a world of finite resources for its production.

BRILLE-TO-PRINT TRANSLATION POSES FURTHER CHALLENGES

In the 1980s, personal electronic braille notetakers entered the marketplace and, as a result, braille-to-print translation software began to be developed along with the older print-to-braille technology. Once again, interpretation issues with the braille code emerged that had hitherto not drawn much attention. When a particular braille character can represent either a question mark or

the word “his,” for example, human readers can easily determine the meaning of the symbol through its context. It is impractical to incorporate that kind of understanding into computer software.

The algorithms used in braille translation programs are in the main not terribly complex. Many often resemble the kind of search-and-replace operations one performs using word-processing software. The tricky part is to define the replacement conditions, or what is termed “context.” It is easy enough to implement the rule that an “ea” contraction can only be used between letters. It is also not too difficult to devise ad hoc word-fragment rules so that the contraction for “mother” will be used in words such as “grandmotherly” but not in words such as “chemotherapy.” In some cases, such as in French braille-to-print translation and in the translation of mathematics, however, a much more far-ranging and complex analysis of the surrounding text may be required to ensure the translation is error free.

NEW ERA, ADDITIONAL PROBLEMS

In modern braille production, error-free translation remains a primary concern, but such accuracy is not necessarily the aspect of production that requires the most attention of software developers and transcribers. The average transcriber who uses such software spends most of his or her time converting the format, not the content, of the text. Because print format is characterized by unbounded diversity, the “rules” for format conversion are much less deterministic than those governing content, and as a result a certain degree of human judgment is inevitably required. Since the earliest days of braille translator development, it has been recognized that the problem of format conversion could be improved if structural markup—that is, coding of text to indicate hierarchy and organization of text elements—were more widely employed in print documents. Although such markup (usually done these days in a code known as XML) is employed in DAISY and NIMAS standards, the vast majority of print texts are not coded in this way.

There are countless other challenges to developing braille translators, since nearly every aspect of braille translation is constantly changing, presumably in the direction of improvement, although some “improvements” are certainly debatable. Computer hardware, including peripheral devices such as braille embossers, becomes faster. As operating systems change, the methods by which translation software interacts with them no longer works and needs to be changed as well. Related programs such as word processors and their file formats change, and both the old and new formats must be supported by translation software. New programs come into being and must be taken into account, and so on. In addition, the rules of all the various braille codes supported by such software (including codes in other languages, braille mathematics codes, and the international music code) change, sometimes under the influence of braille translators. In spite of these many considerations, if the ideal braille translator could be developed it would seamlessly serve the greater tool, braille itself. We are not completely there yet; the work continues.

REFERENCES

- Sullivan, J. (2008). *Early history of braille translators and embossers*. Westford, MA: Duxbury Systems. Retrieved June 19, 2009, from <http://www.duxburysystems.com/bthist.asp>

Joe Sullivan, M.S., president and chief architect of the Duxbury Braille Translator, Duxbury Systems, 270 Littleton Road, Unit 6, Westford, MA 01886-3523; e-mail: <joe@duxsys.com>.

MORE BRAILLE

As the celebration of Louis Braille’s 200th birthday continues, readers can find all the *JVIB* Louis Braille Bicentennial Celebration essays online at <www.afb.org/jvib/louisbraille>. Discover more interesting information on the life and impact of Louis Braille at: <www.afb.org/louisbraille>.