ELECTRONIC PUBLISHING, VOL. 1(2), 117–131 (SEPTEMBER 1988)

Computerized Braille typesetting: another view of mark-up standards

R. ARRABITO AND H. JÜRGENSEN
Department of Computer Science
The University of Western Ontario
London, Ontario
Canada, N6A 5B7

SUMMARY

Recent advances in computerized text processing will not only revolutionize methods of publication, but may also increase the availability of information for the handicapped—especially for blind or visually impaired individuals. In this paper we discuss the feasibility of a direct translation of typesetter input into Braille output with special emphasis on scientific and mathematical text. To do so we use the TEX computer typesetting system as a paradigm; however, the essence of our conclusions is true for other systems too. We briefly describe the present state of a related implementation project. Our study derives several recommendations concerning the standards for mark-up languages and for Braille encodings. They strongly support the development of "semantic" mark-up.

KEY WORDS Braille Markup Typesetting TEX

INTRODUCTION

Recent advances in computerized text processing are revolutionizing our methods of publication and dissemination of information. One could hope that they also have a positive impact on the integration of handicapped individuals into the exchange of information and thus into society in general. Our particular concern which led to this paper was the availability of information—especially scientific texts—for blind readers. We focus on problems arising in the translation of typesetter input into standard Braille output; however, many of our comments would also apply, *mutatis mutandis*, to other unusual types of output like voice synthesizers, for instance[1].

Traditionally, printed information is accessible to a blind person either as Braille encoded hardcopy or acoustically through a reader. Several benefits for the blind can be expected to result from the introduction of "electronic publication" methods. They include the following:

- large volume production of Braille coded texts;
- fast access to documents on demand;
- simplified information exchange between blind and sighted individuals;
- high quality document composition by a blind person.

Some of these expectations and the available technology are discussed below in greater detail.

The main issue of this paper is to present the results of a feasibility study into the possibility of an automatic translation of typesetter input into grade II Braille and Nemeth

0894–3982/88/020117–15$07.50
Received 4 April 1988
Revised 21 June 1988

Braille (the Nemeth Braille code is used as a "standard" in North America to represent mathematical and scientific notation). We mainly consider scientific texts with a special emphasis on mathematical textbooks and research papers. The typesetting system used as a paradigm is TEX[2]; however, we believe that this choice of a system does not affect the essence of the result of our investigation. In the course of this study we addressed the following items:

- development of a Braille output driver;
- development of a macro package to supersede the plain TEX macros;
- potential changes to the TEX source code;
- potential changes to the Braille transcription rules.

Our conclusions can be summarized as follows:

- An automatic translation of TEX input into standard Braille output is impossible in a very strict sense.
- The major reason for this difficulty is the fact that TEX and—to some extent—also the Braille codes emphasize the "syntactic" aspect of typesetting (or mark-up).
- These problems are not specific to TEX or Braille; they would arise as well with most other mark-up systems when used for uncommon output devices.

Our findings add another set of arguments in favour of "semantic" (or descriptive) mark-up to the ongoing discussion concerning mark-up standards[3].

The paper is structured as follows: Section 2 briefly introduces the problem. In Section 3 we give a brief survey of Braille standards. Section 4 contains a description of computer equipment designed specifically for the blind. Section 5 reviews those basic properties of the TEX system which are referred to in this paper. A survey of Braille transcription and other work related to this paper is given in Section 6. In Section 7 we discuss general aspects of the translation of typesetting system input to Braille output and then—as an example—outline issues which are specific to TEX. Sections 8–11 deal with the following special aspects: Braille printer drivers, literary grade II Braille, Nemeth Braille, and the textbook transcription standards. In these sections our main point of interest is to derive criteria for designing translatable mark-up systems. The conclusions are summarized in Section 12. In Section 13 some directions for future work are indicated.

The final remark in this introduction concerns our list of references. The literature on computerized Braille is vast. Therefore, our list is necessarily incomplete as far as computerized Braille production in general is concerned. However, we made every attempt to ensure that all publications specifically relevant to the relation between computer typesetting mark-up and Braille printing are listed.

THE PROBLEM

When one of us (H. J.) first heard the question of obtaining Braille output from TEX input raised by M. Clark at the 1983 TEX users' group meeting (see also [4]) he did not imagine that he would get involved in precisely this kind of work. Two years later when he met the first author (R. A.), who was then attending some of his computer science courses, he was confronted with the question again. The idea for this research originated from communication problems encountered during those computer science courses. The instructor used TEX to typeset the texts of course material, assignments, and exams. In

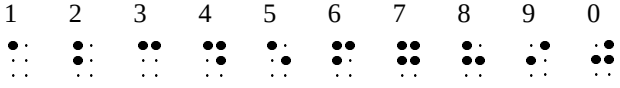
addition to the printed copy made available to every participant of the course, the student (R. A.) also received a copy of the TEX input file by computer mail, which he could then print on his Braille terminal. Although this was better than having no Braille material at all, some of the TEX input was, of course, quite unreadable. To illustrate this point, the reader should imagine having to read and understand a complex mathematical table given by an `\halign` and a few macros.

It was then that we started thinking about how to automate the translation process. The first idea was just to write a new driver which would transform a `DVI`-file into Braille. We realized very soon, however, that this "quick solution" would not lead to acceptable results. The next step was to investigate how the plain TEX macros would have to be re-defined in order to achieve the desired result. Some quite subtle difficulties have meanwhile convinced us that the goals cannot be reached without changes to TEX itself. In fact, we are forced to conclude that the very structures of the TEX language and the Braille transcription rules prohibit a universal translation algorithm.

A BRIEF SURVEY OF BRAILLE

Braille is a system of encodings of print in embossed dot patterns used for reading and writing by the blind. Each Braille character occupies a cell of fixed size. It consists of two columns of dots, numbered 1,2,3 and 4,5,6 from top to bottom. (There are also Braille codes using two columns with four dots eac[5,6].) In this paper, Braille characters are represented by dot images; as examples we list the Braille representations of the first ten characters of the alphabet which, when preceded by the indicator for numbers, also denote the ten digits.

a	b	c	d	e	f	g	h	i	j
1	2	3	4	5	6	7	8	9	0



For instance, the symbol  represents the character with the dots 1,2,4,5 raised. There are $2^6 = 64$ Braille characters altogether with  used as the space character. The dimensions of the Braille cell, according to the Library of Congress standards, are given in[7].

Certain Braille printers also permit a graphics mode. However, in this paper we do not consider Braille graphics at all, even though, in our experience with computer science teaching, it has turned out to be quite an important issue.

Given the fact that only 64 Braille characters are available, it is clear that special encoding rules had to be developed for different applications. Different languages use different encodings (see [8,9], for instance). In this paper, we restrict our attention to the Braille "standards" used in North America.

In addition to the variation according to language, there exist different "grades" for the encoding: Grade I Braille, for instance, renders text with all details concerning punctuation, capitalization, spelling, and numerals, whereas grade II Braille employs a system of contractions and abbreviations[8,10]. Typically, a grade II Braille text is 30% shorter than its grade I counterpart.

Similarly, Brailling "standards" exist for special applications like music, phonetics, computer codes, and mathematics—to mention only a few. For mathematics and science, the Nemeth Braille code has the rôle of a standard in North America[11]. Other codes for mathematics exist or have been proposed (see [12,13,14,15], for instance).

In addition to the various Braille codes there are transcription standards which concern the format of the transcribed text. Again, these vary depending on the application. As far as textbooks are concerned, the rules accepted in North America are listed and illustrated in[15].

In this paper, we focus on transcription—or rather translation—of TEX input into grade II English Braille for plain text and into Nemeth Braille for mathematics.

Several important observations need to be made with respect to Braille transcription:

- There are standards to be obeyed which cannot be changed easily; a substantial change of the standards would necessitate re-training of readers and transcribers.
- Braille makes little use of planar information; most transcription rules call for linear information representation.
- Braille coding is highly context-sensitive.
- Very often, the Braille encoding rules use the printed version of the document as the source of reference rather than a semantic abstraction of it.
- The encoding rules do not appear to be designed with automatic transcription in mind.
- There are few documented rules by which the Braille encodings of new features could be derived automatically.

Some of these remarks will be addressed again in subsequent sections of this paper.

COMPUTER EQUIPMENT FOR THE BLIND

In this section we give a brief list of some equipment which can be attached to a computer and used by a blind person.

There is a variety of Braille terminals. Typically, such a terminal has a display of up to 40 Braille cells realized by $40 \times 6 = 240$ pins that can be raised or lowered according to the required dot patterns. A line of more than 40 cells is broken into several lines. This operating mode makes reading long documents on a terminal rather awkward and slow. The keyboard of a Braille terminal is either a "qwerty" keyboard or a 6-dot Braille keyboard as used on Braille embossers or a combination of these. Some Braille terminals are equipped with local back-up memory to facilitate "scrolling".

As far as normal text is concerned, interaction with a computer is made much easier by voice synthesizers which can be programmed to read the text on the screen. The sound quality of present-day versions is not impressive, but one can adjust to it. However, when technical text is shown on the screen, the voice synthesizer usually turns to just pronouncing the single characters whereas a sighted person would read according to the meaning of what is displayed. In part, this may be a problem of re-programming the synthesizer. However, to some extent this is also related to the problem of appropriate mark-up.

Braille printers are available for hardcopies. Some Braille printers even permit a graphics mode. Hardcopies seem to be indispensable, however bulky they are.

Various reading machines have been developed which convert printed material directly into uncontracted Braille or voice output. They seem to work well only with normal textual material under tight constraints concerning the type and quality of the paper and the printing.

Recently a graphics display for the blind has been introduced[16]. Its largest version consists of a matrix of 120 × 120 pins which can be raised or lowered. Although the resolution of this display is very low, it can serve a very important purpose, as many simple graphics applications do not really require a higher resolution.

A BRIEF INTRODUCTION TO TEX

TEX is a language for setting type by computer. Its first version was designed about ten years ago; its revised and final version[2] has rapidly been accepted by the scientific community as well as by major publishers of mathematics and computer science books. Although TEX was designed with mainly mathematics typesetting in mind, it has meanwhile been shown to be quite adaptable to other typesetting tasks as reported at the 1987 TEX users' group meeting[17,18].

The simplest way to think about TEX is to consider it as a programming language. Every token in the TEX input file, which is produced by the author for instance, causes some action: the setting of type, the calculation of parameters or dimensions, or the arrangement of complicated structures. Thus, the TEX input file is a linear representation of the two-dimensional layout of the printed paper. TEX includes powerful mechanisms for alignments, line breaking, page breaking, etc. It grants the user control over minute details of the setting process; at the same time, it gives the user a macro instruction mechanism to program quite complex typesetting tasks.

A typical typesetting process with TEX works as follows: The TEX system receives the TEX input file containing the document description. In addition, TEX needs metric information about the fonts being used which is contained in `TFM`-files. TEX's output is sent to a so-called `DVI`-file, which contains printing device independent instructions of how to print the document. A separate program, specific for every kind of printer, will then execute the instructions contained in the `DVI`-file to produce the printed document. This program, the `DVI`-driver, uses information about the actual shape of the characters to be used in the printing process. The relevant information is contained in `GF`-files or similar, depending on the implementation.

The TEX system itself consists essentially of two layers. A small set of basic features is defined by the actual TEX system. Most of the typesetting instructions of TEX are then implemented using a set of control sequences and macro definitions, a macro package, for short. TEX itself comes with the macro package 'plain TEX'; other major packages are LATEX[19] and $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX[20], for instance. These packages provide frameworks for definitions of typesetting styles. However, in spite of being relatively high-level they usually permit the user to access all of TEX's primitives as well. Thus at any time the writer can control every detail of the typesetting process if he/she prefers to do so.

Some aspects of TEX which are relevant to the subsequent study of TEX-to-Braille translation can be summarized as follows:

- TEX represents the document layout in a completely linear fashion; however, it employs primitives which express planar information.

- Rather than being an encoding of the text to be typeset, the TEX input file is a program for setting the document.
- TEX permits a mixture of high-level and low-level instructions; its "source" of reference is the final printed version of the document rather than its semantics.
- TEX's macro facility allows one to define arbitrary new symbols; its primitives enable one to create any kind of layout.

The combination of these properties of TEX with some of the requirements of Braille turns out to lead to inherently unsolvable problems in the task of automatic translation of TEX input into Braille output. Details are addressed in subsequent sections of this paper.

We should like to emphasize that in this paper we use TEX only as a paradigm and that our conclusions will apply to most existing computerized typesetting or mark-up systems.

A SURVEY OF BRAILLE TRANSCRIPTION

Before the introduction of computers into the production process, Braille transcription worked essentially as follows: a person who had been trained in the relevant Braille codes would have a printed copy of the text in question and produce a Braille copy using a Braille writing machine. Typically, such a Braille writing machine has six keys for embossing a Braille character, one for each of the dots of the Braille cell, and a few more keys for operations like 'space', 'backspace', etc. In the case of single copy, the Braille is produced directly onto heavy stock paper. For large scale production, the Braille is embossed onto zinc plates which are then applied under pressure to the heavy stock paper[21]. The Braille version would have to be proofread and corrected; corrections could require that complete pages be re-done.

Braille transcription is dificult, time intensive and costly. The training time for a transcriber is given as between 6 months and a year[21]; even longer periods are required for training in complicated codes like Nemeth Braille. A page of Braille takes about 30 minutes to produce on the average. This includes proofreading and corrections. Typically, a page of print results in about two pages of Braille. Given these conditions, it is clear that only a very small part of the printed publications can actually be transcribed. Moreover, access to less frequently demanded documents is very slow. Given the traditional set-up, browsing through scholarly journals or recent technical reports is something a blind reader can only dream of.

With the introduction of computers into the transcription process certain simplifications became feasible. In the first step, the text is put into machine readable form. The task of translating the input into Braille is then left to a computer program. Several programs exist that afford the conversion from ASCII Braille to (literary) grade II Braille (see [22,23,24], for example). Some successful attempts at computer-aided transcription of mathematics have also been made[25]. However, to our knowledge, the automatic transcription into the Nemeth Braille code for science notation is not offered by any software package up to now. For technical texts, a typist with special training is still required, who would introduce the correct transcriptions, like

'.a' for 'α' denoting the Braille sequence 

for example. Whereas for "plain texts" one may hope to be able to rely on compositor tapes[26] or automatic text reader output, for scientific texts, human interference in the text acquisition step has not been eliminated so far. Thus the present moderately computerized Braille transcription process is very expensive; we were given the figure of $3.30 as the estimated cost per Braille page of a technical text.

The introduction of computers into the Braille transcription process in this quite limited sense has resulted in a drastic decrease in production time. Fortier[21] reports a typical turn-around-time of about a week from the reception of a book to the completion of its Brailled version; by 1984, the University of Manitoba Braille Project produced transcriptions at a rate of 30,000 pages per year. On a slightly smaller scale, our experience with the computer based Braille transcription service performed by the Computer Braille Facility at the University of Western Ontario indicates the enormous advantages of computer availability in the Brailling process. One has to realize, however, that even this increased speed of Braille transcription does not—by any means—come close to what is actually required to make access to information nearly as easy for a blind person as for a sighted one.

It ought to be mentioned too, that computerization of Braille transcription will also help with respect to text storage. Whereas keeping large numbers of zinc plates over extended periods was economically and practically impossible, the availability of electronic or magnetic storage media allows one to build libraries of sources of Brailled texts.

In view of our project, the following aspects of the Braille transcription process—as it is performed today—deserve special mention:

- The file created in the input phase is expected to contain only plain ASCII text or the ASCII equivalents of special Braille notation.
- Programs which produce grade II literary Braille from ASCII Braille exist. However, no program produces Nemeth Braille automatically.
- Braille transcription programs do not produce output according to the 'Code of Braille Textbook Formats and Techniques' automatically[15].
- For technical text, the input phase relies mainly on Braille experts who would retype and encode the relevant parts of the texts.
- The printed copy of the text is the source of reference for the transcription process.

Our work addresses precisely these points. Instead of using the printed copy of the text as the unique source of reference, we propose to use the author's copy of the text (or some equivalent version of the text in question)—in the case of books or scientific papers, which have been prepared using a mark-up language like TEX, the source for the transcription should be the TEX input file. This paper determines the potential and limitations of this proposal.

REQUIREMENTS FOR AUTOMATIC BRAILLE TRANSCRIPTION

As indicated in the previous sections, the crucial problem in computerized Braille transcription is that of obtaining the text in machine readable form with enough detail. In addition to the actual text, the required information includes certain layout specifications similar to those outlined in[15], and linearized encodings of non-linear structures and special symbols, like mathematical formulae. In both respects, it is of course not

necessary that the input file conform to some specific Braille coding rules as long as a translation respecting those can be achieved.

The trend in the publishing industry is clearly towards further computerization. As far as the printing of mathematical texts is concerned, a model of future production processes can be illustrated using the paradigm of TEX: the author supplies his text, marked up with parameterized macros; the publishers check and correct the mark-up, add their macro package and initiate the printing process. In fact, as mentioned above, other branches of the publishing industry seem to envisage a similar model of the publication process with the actual setting of type based on TEX or some other mark-up language.

In view of these remarks, it seems natural to postulate that the source of information for the Braille transcription should be the same as the for the normal publication process, that is, the typesetting input file. (Of course, certain copyright problems would have to be addressed in this context.) This idea has several advantages:

- Since complete information about the appearance of the printed text is present in the input file, no information needs to be added for the transcription process.
- Correctness or rather consistency of the Braille transcription is guaranteed.
- Corrections or changes made in the original are automatically included in the Braille version.
- The Braille production of a text is nearly as fast as the normal printing because the time-consuming input phase is eliminated from the process.
- Braille copies of a text can be produced on demand and on short notice. This is particularly important with textbooks for blind students integrated into regular curricula.
- Braille versions of short-lived but frequent publications (like newspapers or magazines) can be made available quickly.
- Fast and reliable communication between sighted and blind individuals—even involving complicated technical documents—is possible using computer mail (or a similar information exchange process) if based on a common representation of both the printed and the Braille version of the text.

With these ideas in mind we started to investigate the feasibility of a translation of TEX input files into Braille output. TEX was chosen as the paradigm on which to base the study for various reasons: TEX had been in use locally for some time, with some expertise concerning the details of the system available; some major publishing companies for mathematics and computer science books had adopted TEX; a large amount of 'semi'-published material in mathematics and computer science was typeset using TEX; there seemed to be a trend towards having more and more of the local course material written in TEX.

As mentioned before, TEX is only used as a paradigm and as a test case in this project. Our observations and conclusions are not really specific to TEX. Similar ones could probably have been obtained from the study of any other computer typesetting language for technical texts.

Ideally, the goal of TEX-to-Braille translation is very simple: given any TEX input file, the translation program would automatically produce a correct Braille version. As TEX allows one to define one's own environment by using macro packages, such a *universal* translation program would have to be based on the following components:

(1) Braille "fonts" replacing the usual fonts with the necessary information contained in TFM-files;
(2) a DVI-to-BRAILLE driver which knows about the characteristics of the Braille printer to be used (of course, no GF-files or similar files are required);
(3) a re-definition of TEX's primitives.

As is shown in the sequel, this rigorous approach is impossible due to certain incompatible properties of TEX and Braille. Thus, we also consider an additional component:

(4) a partial re-definition of macro packages like, for instance, the plain TEX.

Of course this last component leads away from a universal translation program. Whether this is acceptable will have to be determined by pragmatic considerations.

As a very pragmatic approach to the TEX-to-Braille translation problem we mention the program described in[27]. There the transcription process is guided by a file which for every "TEX instruction" contains "its Braille equivalent" and "a parameter" which defines special actions to be taken. This approach cannot achieve universal TEX-to-Braille translation when TEX is used at its full power for two reasons: first, the translation file would have to be changed to include equivalents of the macros defined in the input—depending on the parameter part, this could even involve inserting subroutines into the translation program itself; second, as is explained in more detail below, the fact that TEX and Braille emphasize the "syntax" component of the layout allows one to introduce TEX constructs which have no unique Braille equivalent.

In the next four sections of this paper we discuss the requirements and implementation components for ideal TEX-to-Braille translation separately. The results are then summarized in a section containing conclusions and recommendations.

BRAILLE PRINTER DRIVER

Among the components for a universal TEX-to-Braille translation system, the DVI-to-BRAILLE driver is the simplest one to obtain. The assumption is that only such instructions are contained in the DVI-file created by an appropriately re-defined TEX, which can actually be executed by the Braille printer.

Our approach to building such a DVI-to-BRAILLE driver and the TFM-files for the Braille fonts is based on a driver generator called GENDRIV, which was developed by K. Guntermann[28,29,30]. We use a version modified to run under VAX/VMS and subsequently equipped with an "interactive" input program[31].

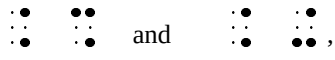
The driver generator takes a description of the printer, the fonts to be printed, and the device-independent file format as input and produces TFM-file equivalents and a WEB program as output. The latter, when linked together with a 'kernel' program, forms the actual DVI driver.

GENDRIV makes certain assumptions about the printers. They are satisfied by most dot-matrix printers. However, no Braille printer meets these requirements. Thus it was not completely unexpected that certain problems would be encountered. After all, we were trying to abuse that system. So far, as a proof of feasibility in principle, we have created a small and incomplete DVI-to-BRAILLE driver. The specification of a complete one is planned for the near future.

Some of the reasons for going this seemingly complicated route of abusing a driver generator rather than just writing a simple DVI-to-BRAILLE driver directly are as follows:

- The specifications made for one printer can easily be modified if a different printer is used. Generating a driver will be easier than writing one, once the process is well understood and its pitfalls are known.
- GENDRIV allows one to create fonts based on graphics facilities of the printer. This may later turn into a way of exploiting the graphics capabilities of certain Braille printers.
- GENDRIV permits the definition of composite characters in fonts. For example, the symbol ≠ could be obtained by printing the slash over the equal sign on certain dot matrix printers. The DVI driver would take care of generating the necessary instructions for the printer.

In particular, the last feature is quite useful in our context. Many mathematical characters, which TEX expects to find as single characters in its special fonts, are rendered as character sequences in Nemeth Braille. For example, the symbols ∩ and ∪ are transcribed as

 and ,

respectively. Typically, TEX expects to find these symbols in the font cmsy10; with the TFM-file of cmsy10 defined appropriately, the DVI-to-BRAILLE driver can take care of this part of the transcription process automatically.

OBTAINING GRADE II BRAILLE

Several programs for the translation of plain text—in ASCII, say—into literary grade II Braille exist (see [22,21,23], for example). In the TEX-to-Braille translation process, the creation of the grade II version must precede all setting of type as the conversion into grade II will change the amount of text to be set.

The obvious solution is to use a pre-processsor to perform this conversion—and this is the solution which we are implementing despite its shortcomings.

Any pre-processor solution has to find a reasonable compromise between completeness, correctness, and efficiency of the conversion process. A simple approach is to subject to the grade II conversion only those parts of the input file which are clearly recognized as text; that is, everything inside the first level of mathematics mode or inside macro definitions would be ignored. Clearly, in this way certain parts which should be converted are left unchanged, and incorrect conversions can arise from macros substituting parts of words. Moreover, as an author may re-define the symbol indicating mathematics mode—and, even worse, he/she may do so in some deeply nested macros—no simple general way of distinguishing mathematics and text modes exists. Note that in plain TEX the various modes can easily be abused—and are in fact "abused" frequently for special effects. For instance, to set the raised footnote reference one commonly uses mathematics mode. To create displayed text, the TEXbook recommends to use display style which again implies mathematics mode ([2], p. 185).

Thus, a general pre-processor for grade II conversion would incorporate nearly all the

syntax analysis and macro expansion features of TEX itself. At present, we opt for the simpler version and accept a certain error rate. This allows us to use a modified version of one of the existing grade II translation programs as the pre-processor.

OBTAINING NEMETH BRAILLE

Both Nemeth Braille and TEX input express planar information in a linear way. One would assume that it is an easy task to translate one into the other. However, the two linearizations turn out to be crucially incompatible in some respects. Some details are given in the folowing.

The easiest part of Nemeth Braille transcription is to create the Braille versions of the basic mathematical symbols, that is, most of those mathematical symbols which are contained in TEX's fonts. In fact, the major part of this task can be achieved in the `DVI-to-BRAILLE` driver as mentioned above.

A few other simple parts of mathematics printing can also be implemented quite easily by re-defining some of TEX's primitives; fortunately, TEX has been designed with this option in mind. Thus, with some but still little pain, exponents, subscripts, etc. can be transcribed properly. We had considerable problems with the capital letters for instance. In normal Braille text, capitals are usually not indicated as such. On the other hand, the distinction has to be made in mathematics mode. The obvious 'solution' of making all capital letters active did not work because then one could not use them in control sequences any more.

Really serious problems arise from the following situations:

- TEX input uses the same syntactical structures for many different purposes. For example, the caret symbol `^` is used for exponents, upper limits of sums and integrals, and just to raise text, as for footnotes. In Braille different constructs would be used depending on the context or rather, the meaning of the symbol in any particular instance. The very definitions of TEX make heavy use of such ambiguities. For Braille transcriptions these ambiguities have to be resolved, a difficult, if not impossible task.
- The author of a TEX input file may create his own symbols; if this is done using only "high-level" constructs like normal characters, subscripting, superscripting, etc., an analogous construction in Braille should yield the corresponding semantically equivalent symbol.[1] Due to the lack of meta-rules in Nemeth Braille, this process is not guaranteed to be successful. There is an even worse scenario, however: Assume that the author uses "low-level" constructs like line segments and direct positioning to create symbols (the Braille characters in this paper have been defined in this way). Clearly, no adequate automatic Braille transcription can be given, even if the final printed version of the author's creation turns out to be a character with a well-defined Braille representation.

The difficulties with automatic transcription of TEX into Nemeth Braille can be summarized as follows:

[1] Even in such cases, the freedom given in TEX results in serious ambiguities: A user could create a new function symbol like '`log`' which requires roman type; if he/she does so without using the `\mathop` construct, is the resulting string still in mathematics mode, that is, should grade II contraction apply? Of course, a user shouldn't do this, but it happens nevertheless.

- Nemeth Braille does not provide meta-rules for its own expansion.
- TEX does not enforce a clear distinction between high-level and low-level constructs.
- TEX does not enforce semantic unambiguity of syntactic constructs.

These problems seem to render an *automatic* and *universal* TEX-to-Braille translation system impossible.

THE TEXTBOOK TRANSCRIPTION STANDARDS

So far, we have studied the problems posed by the textbook transcription standards as defined in[15] only in a superficial manner. Whereas most of the constraints on the layout seem to be easily implemented by re-definitions of some of TEX's primitives, other requirements related to cross-referencing with the printed version will turn out to be feasible but quite costly in terms of computing resources. The studies indicate that some of the problems encountered with Nemeth Braille transcription have their counterparts with regard to the standards for transcribing textbooks. Again the crucial difficulty is that TEX and—to some extent—also the Braille standard focus on "syntactic" rather than "semantic" mark-up.

CONCLUSIONS AND RECOMMENDATIONS

Assuming a computer typesetting environment, it seems to be natural to require that the Braille transcription input be the same as the usual typesetting input. This would result in a shorter publication delay of the Braille versions of publications, a decrease of the cost of Braille production, an achievement which is necessary if one plans to meet the demands of a modern society, which recognizes the special needs and abilities of its minority groups. The case study of a TEX-to-Braille translation system has revealed the following serious problems in the way printed documents and their Braille translations are described at present:

(1) Issues of syntax and semantics are not clearly distinguished. Mark-up languages allow for and make use of semantic ambiguities of syntactic constructs. In this way, a semantics-preserving translation of typesetter input to Braille is made impossible.
(2) Nemeth Braille lacks meta-rules to define new constructs. Thus, there is no accepted and automatic way to map an author-defined new symbol onto a new Nemeth Braille symbol.
(3) The layers of detail of representation in a mark-up language like TEX are not clearly distinguished. Thus, a clean style of writing a document description is not enforced.

Admittedly, it may sound like a strange idea to use TEX, a system for high quality typesetting, for printing on a Braille printer, a device with only very crude capabilities. However, the idea of using the TEX input also as the input to Braille transcription seems convincing when one considers the greater accuracy and speed of Braille production to be achieved in this way.

From the perspective of translating typesetter input directly into Braille, one could argue that TEX should not be given to the authors as a mark-up language—it provides too much freedom. Instead, the source for both printing and the Braille transcription process

should be written in a mark-up language which retains the semantics of constructs, avoids ambiguities, and enforces a writing style which does not resort to purely syntactic, low-level features. Such a system could and should be based on TEX or a similar powerful typesetting language. However, as one would avoid assembly code in a program written in Pascal, in the same way, one should avoid using basic positioning commands like "`\raise`" in the author's part of a document. Macro packages like $\mathcal{AMS}$-TEX and LATEX took a few steps in this direction. However, they did not really propose or implement the radically different, layered solution which seems to be required.

The problems encountered in TEX-to-Braille translation strengthen the arguments in favour of "descriptive" mark-up[3] as proposed by SGML[32], for instance. Our criticism does not imply that minute control over typesetting details should be unavailable to an author. Instead, we advocate that the different layers of mark-up be carefully and strictly separated. The text itself should only contain "semantic" and unambiguous mark-up. Only in this way can text processing through media which the author did not foresee be made feasible without change of the mark-up.

Of course, the low-level mark-up languages like TEX[2], SCRIBE[33], *troff*[34], etc. allow the user to make this separation of layers of mark-up. However, the problem is that they don't enforce it. In fact this is a very familar situation in computer programming as well. In this sense, the proliferation of low-level desk-top publishing systems without enforced standards of descriptive mark-up may turn out to generate another Babylonian language dilemma. Adhering to "semantic" mark-up does not preclude the usage of a system like TEX as a basis (see [35] for instance).

Also the Braille standards may require certain modifications to render the code more flexible. As an example, meta-rules describing construction methods rather than specific constructions would help to match the Braille transcription to the way new document structures are created.

FUTURE WORK

The project of implementing a TEX-to-Braille translation system seemed easy in the beginning; it has turned out to be quite difficult, and in its full generality it is impossible. After the feasibility study, which forms the basis of this paper, we shall continue this work in two directions:

- As a short-term goal we want to obtain a complete `DVI-to-BRAILLE` driver and a small usable subset of re-defined TEX primitives. This would allow us to obtain transcriptions on a limited scale and thus gain some more experience with an automated transcription process.
- As a more long-term goal, we plan to study principles of mark-up languages and Braille encodings with the goal of laying out at least the foundations of a system that lends itself to translation into both high-quality type and correct Braille. In particular, we plan to study SGML more closely in view of Braille transcription.

We have been—and shall continue to be—asking agencies for the blind for advice; we are grateful for the helpful recommendations we have received from them in the past. We invite input and assistance from publishing houses, too. To us it seems important that in the development of new mark-up styles, the concerns expressed above be taken into account. The task of making information accessible to all parts of society is as important

as ever; the introduction of computers in the production process could move us closer to its realization.

ACKNOWLEDGEMENTS

This work was partially supported by the Natural Science and Engineering Research Council of Canada, Grant A0243.

We are grateful to Henry Baragar, Jamie Beck, Caryn Bursch, Mary Caco, Phil Du Toit, Ingrid Harrms, Elizabeth Fitzmaurice, Helen Forester, Vinay Jain, Julie Lee, Laurel Linetsky, Bill Magee, Brad McGee, Catherine Nolan, Susan Poaps, Corinne Schriver, Rick Secco and Laurie Summers. They generously and patiently volunteered their time to read the TEXbook to R. A. We also thank Shane Dunne for many helpful discussions.

REFERENCES

1. W. A. Slaby and F.-P. Spellmann, 'Automatische Übersetzung von Texten in gesprochene Sprache', *Textverarbeitung und Informatik, Fachtagung der GI, Bayreuth, Mai 1980*, P. R. Wossidlo ed., Informatik-Fachberichte **30**, Springer-Verlag, Berlin, 1980, pp.64–72.
2. D. E. Knuth, *Computers and Typesetting*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
3. J. H. Coombs, A. H. Renear and S. J. DeRose, 'Markup Systems and the Future of Scholarly Text Processing', *Communications of the ACM* **30**, 933–947 (1987).
4. M. W. Clark, 'Letter to the Editor', *TUGboat* **5**, No. 2, 146 (1984).
5. W. Schweikhardt, *Die Stuttgarter Mathematikschrift für Blinde, Vorschlag für eine 8-Punkt-Mathematikschrift für Blinde*, Report 9/1983, Inst. f. Informatik, Universität Stuttgart, 1983.
6. O. Sueda, Eight-Dot Braille Code and Automatic Braille Translation Boards', *Computerised Braille Production, Proceedings of the 5th International Workshop, Winterthur, 1985*, J. M. Ebersold, Th. Schwyter and W. A. Slaby ed. Katholische Universität Eichstätt, Schriftenreihe des Universitätsrechenzentrums **1**, Eichstätt, 1986, pp.273–303.
7. A. M. Goldberg, E. M. Schreier, J. D. Leventhal and J. C. DeWitt, A look at five Braille printers. *Journal of Visual Impairment and Blindness* **81**, No. 6, 272 (1987).
8. *English Braille, American Edition, 1959, Revised 1962, 1966, 1968, 1970, 1972, (with) Changes (of) 1980*, American Printing House for the Blind, Louisville, Kentucky, 1984.
9. E. Freud, *Leitfaden der deutschen Blindenkurzschrift*, Marburg, 1973.
10. M. B. Dorf and E. R. Scharry, *Instruction Manual for Braille Transcribing*, Division for the Blind and Physically Handicapped, Library of Congress, Washington, D. C., 1979.
11. *The Nemeth Braille Code for Mathematics and Science Notation, 1972 Revision*, American Printing House for the Blind, Louisville, Kentucky, 1985.
12. J. Meekel and M. Truquet, 'A System for Transcribing Mathematics into Braille', *SIGCAPH Newsletter* **31**, 27–32 (1983).
13. K. Britz, H. Epheser, G. v. d. Mey and F. M. Scheid, *Neufassung und Vervollständigung des Systems der internationalen Mathematikschrift für Blinde*. Marburg, 1955.
14. S. R. Hussey, *Mathematical Notation. The Halifax Code*, Revised by D. Harmer, L. Legge, D. Hagen. Sir Frederick Fraser School, Halifax, 1981.
15. *Code of Braille Textbook Formats and Techniques, 1977*, American Printing House for the Blind, Louisville, Kentucky, 1986.
16. W. Schweikhardt, T. Fehrle, 'Ein rechnerunterstützter Zeichenplatz für Blinde', *Computerised Braille Production, Proceedings of the 5th International Workshop, Winterthur, 1985*, J. M. Ebersold, Th. Schwyter, W. A. Slaby ed., Katholische Universität Eichstätt, Schriftenreihe des Universitätsrechenzentrums **1**, Eichstätt, 1986, pp.251–261.
17. D. Ness, 'The Use of TEX in a Commercial Environment', *TEX Users Group, Eighth Annual Meeting, Seattle, August 26–26, 1987, Conference Proceedings, TEXniques* **5**, TEX Users Group, Providence, R. I., 1988, pp.115–123.

18. E. M. Barnhart, 'TEX in the Commercial Environment, Setting Multi-Column Output', *TUGboat* **8**, 185–189 (1987).
19. L. Lamport, LATEX. A Document Preparation Language, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
20. M. D. Spivak, *The Joy of TEX. A Gourmet Guide to Typesetting with the AMS-TEX Macro Package*, American Mathematical Society, Providence, Rhode Island, 1986.
21. P. A. Fortier, 'A Computer Aid for the Visually Handicapped: Braille Production at the University of Manitoba', *Proceedings of the First International Conference on Computers and Applications*. IEEE Computer Society Press, Silver Spring, Maryland, 1984, pp.208–214.
22. Braille and Computers, *Aids and Appliances Review* **11** (1984), Carroll Centre for the Blind, 770 Centre Street, Newton, Massachusetts.
23. D. McHale, *A Simple Algorithm for Automated Braille Translation*, Department of Computer Science, University of Washington, Seattle, Washington, Technical Report No. 85-08-04, 1985.
24. B. Eickenscheidt, W. A. Slaby and H. Werner, 'Automatische Übertragung von Texten in Blindenschrift', *Textverarbeitung und Informatik, Fachtagung der GI, Bayreuth, Mai 1980*, P. R. Wossidlo ed. Informatik Fachberichte **30**, Springer-Verlag, Berlin, 1980, pp.50–63.
25. W. Schweikhardt, Rechnerunterstützte Übertragung einer mathematischen Formelsammlung in die 'Stuttgarter Mathematikschrift für Blinde', *Computerised Braille Production, Proceedings of the 5th International Workshop, Winterthur, 1985*, J. M. Ebersold, Th. Schwyter, W. A. Slaby ed., Katholische Universität Eichstätt, Schriftenreihe des Universitätsrechenzentrums **1**, Eichstätt, 1986, pp.239–249.
26. A. Schack and J. Schack, 'Computer Conversion of Compositor Tapes to Grade Two Braille', *Proceedings of the Braille Research and Development Conference, 1966*. Sensory Aids Evaluation Development Center, M.I.T., Cambridge, Massachusetts, 1966, pp.44–64.
27. H. Pinell, 'Automatische Übertragung mathematischer Formeln in 6-Punkt-Mathematikschrift', *Computerised Braille Production, Proceedings of the 5th International Workshop, Winterthur, 1985*, J. M. Ebersold, Th. Schwyter, W. A. Slaby ed. Katholische Universität Eichstätt, Schriftenreihe des Universitätsrechenzentrums **1**, Eichstätt, 1986, pp.211–216.
28. K. Guntermann, *Ein Ansatz zur automatischen Erzeugung von Ausgabetreibern für die Ansteuerung von Zeichendruckern durch Textformatierer*, Dr.-Ing. Thesis, Technische Hochschule Darmstadt, Darmstadt, 1984.
29. K. Guntermann, '`GENDRIV`, a Driver Generator for Low Cost Devices Using Built-in Fonts', in *Proceedings of the First European Conference on TEX for Scientific Documentation*, D. Lucarella ed., Addison-Wesley Publishing Company, Reading, Massachusetts, 1985, pp.197–204.
30. K. Guntermann, *Benutzeranweisung zur Treibergenerierung für TEX*, Manuscript, Technische Hochschule Darmstadt, 1984. (Partial translation into English by H. Jürgensen, London, Ontario, 1985.)
31. E. Rowe, *Driver Generator for TEX on a Family of Line Printers*, Undergraduate Project Report, Department of Computer Science, The University of Western Ontario, London, Ontario, 1986.
32. *Information Processing—Text and Office Systems—Standard Generalized Markup Language (SGML)*, ISO 8879-1986, American National Standards Institute, New York, 1986.
33. *SCRIBE Document Production System User Manual*, (1984), Unilogic Ltd., Pittsburgh, Pennsylvania.
34. J. F. Ossanna, *NROFF/TROFF User's Manual*, Department of Computer Science, Bell Laboratories, Murray Hill, New Jersey, Technical Report No. 54, 1976.
35. L. A. Price and J. Schneider, *MARKUP Reference Manual*, Hewlett-Packard Co., Palo Alto, California, 1987.