European Bibliography Styles and MlBibTeX

Jean-Michel Hufflen
LIFC (FRE CNRS 2661)
University of Franche-Comté
16, route de Gray
25030 Besançon Cedex
France
hufflen@lifc.univ-fcomte.fr
http://lifc.univ-fcomte.fr/~hufflen

Abstract

We explain why MlBibTeX has evolved and why the last version (1.3) provides a new language — close to XSLT — for bibliography styles. We also show that this evolution has been influenced by an investigation regarding bibliography styles used throughout European countries.

Keywords: bibliographies, multilingual features, European bibliography styles, BibTeX, MlBibTeX, bst, nbst, XML, XSLT.

Résumé

Nous présentons les raisons de la récente évolution de MlBibTeX, la dernière version (1.3) adoptant un nouveau langage — proche de XSLT — pour les styles bibliographiques. Nous montrons aussi en quoi une enquête à propos des divers styles bibliographiques en usage dans les pays d'Europe a influencé cette évolution.

Mots-clés : bibliographies, multilinguisme, styles bibliographiques européens, BibTeX, MlBibTeX, bst, nbst, XML, XSLT.

Zusammenfassung

Für die bibliographischen Styles benutzt MlBibTeX seit der Version 1.3 eine neue Sprache, die mit XSLT nahe verwandt ist. Die Hintergründe dieser Entwicklung werden im folgenden dargelegt. Dazu werden die bibliographischen Styles, die in Europa benutzt werden, ausführlich untersucht, um den Einfluss dieser Analyse auf die neue Version von MlBibTeX zu zeigen.

Stichtwörter: Bibliographien, mehrsprachigen Funktionen, europäischen bibliographischen Styles, BibTeX, MlBibTeX, bst, nbst, XML, XSLT.

Introduction

Although end-users of LaTeX [27] can directly manage a bibliography for each document by means of the thebibliography environment, this text processor is often used with the BibTeX bibliography program [30]. BibTeX allows its users to group *bibliographical entries* in .bib files comparable to data bases. An example of such an entry is:

```
@BOOK{zelazny1969,
        AUTHOR = {Roger~Joseph Zelazny},
        TITLE = {Damnation Alley},
        PUBLISHER = {Putnam},
        YEAR = 1969}
```

Let us assume that this entry for zelazny1969 is cited within a document. If we run LaTeX, such citations are stored in the auxiliary .aux file. Then this information is used by BibTeX to generate a file containing items cited throughout the document. If we run LaTeX again, this generated file (.bbl file) is processed as part of the whole document, and *references* will look like:

> [1] Roger Joseph Zelazny. *Damnation Alley*. Putnam, 1969.

In this reference and the bibliography of this article, we have used the plain bibliography style, that is, references are labelled with numbers. There exist many other bibliography styles, most of them described in [9, § 13.2]. The bst language, used to put bibliography styles into action, is based on handling a stack, using postfix syntax. It is described in [29] and an example is given in Figure 1: the function used within bibliography styles of BibTeX to format the date of a reference.

In the last decade, LaTeX's multilingual features have been considerably extended. Some packages suitable for one particular language — for example, french

```
FUNCTION {format.date}
{ year empty$
    { month empty$
        { "" }
        { "there's a month but no year in "
          cite$ * warning$
          month
        }
      if$
    }
    { month empty$
        'year
        { month " " * year * }
      if$
    }
  if$
}
```

(The '*' operator denotes the concatenation of strings within the bst language.)

Fig. 1: Formatting dates in the .bst language.

[7], german [32], ... — have been developed. More generally, the babel package [2] can process all the languages it knows without giving any privilege to a particular one.

Since BibTeX does not provide as many multilingual features as LaTeX does — even if the insertion of some slight multilingual features have been put into action [9, § 13.8.2 & 13.9] — we started a new implementation, so-called MlBibTeX (for 'Multilingual BibTeX') [14]. Let us recall that this new bibliography program is in practice compatible with 'old' BibTeX and extends its syntax by using square brackets as syntactic delimiters.[1]

In particular, this reimplementation allows end-users to specify language-dependent optional parts: for example, let us look at the entry `zelazny1969` given in Figure 2. The string given by the syntax `[...] ! french'` — '`!`' meaning 'only' — will appear only if the corresponding reference is processed in French, that is, within a 'References' section for a document written in French, as far as possible. The syntax '`[...] : english`' is for a language change: even if `zelazny2000` concerns a French translation of a book (see the value of the `LANGUAGE` field), some parts — the author's name and the original title — are to be processed in English. This information about 'foreign words' could be useful if LaTeX hyphenates words belonging to such parts. Putting this information into the generated file does not cause errors, in the sense that switches to another language occur within the text produced by MlBibTeX only if this language is available, either by means of a specific package or by means of a suitable option of the babel package.

Last, let us notice the use of an abbreviation for the month name: see the value of the `MONTH` field. It can be processed in English and yield the string 'April'. Likewise, it can be processed in French (resp. German) in which case the result will be '*avril*' (resp. '*April*').

In addition, MlBibTeX allows users to control the use of languages precisely: each entry cited can be processed according to its own language (*language-dependent* approach) or all the entries can be processed w.r.t. the document's language (*document-dependent* approach). See [16, 22] for more details about these approaches.

Between Versions 1.1 and 1.3, MlBibTeX's implementation has deeply changed and now we are using a new language for bibliography styles [19]. This new language — so-called nbst for 'new bst' — is close to XSLT[2] [36], the language used to process and build XML[3] documents. Here we are going to explain why. From a 'philosophical' point of view, this article completes [22], which describes all the features of MlBibTeX Version 1.3, and [23], where we show how to make MlBibTeX fit for a particular language.

In the first section, 'A gulp of history', we show that MlBibTeX's first version could meet the requirements for a multilingual bibliography program, but it would result in a heavy program, in complicated bibliography styles, and the whole would be hard to maintain. Furthermore, we were able to get in touch with people originating from different countries, especially at EuroTeX conferences. So, we started an investigation about bibliographies in practice, and the second section 'Questions and answers' describes it and sums up its results. In the third section, we show why we think that the new version — which will be ready in July 2003 [21] — should be suitable at least for European bibliography styles. Besides, this new version using some features borrowed from XML opens new directions which we sketch in the last section.

Reading this article requires only a basic knowledge of LaTeX and BibTeX. It also requires some knowledge about XML. Good introductions to this meta-language are [8, 33] and [5] as a pocket guide. A more 'official' reference issued by the W3C[4] is [38].

A gulp of history

When we started the first version of MlBibTeX in October 2000, we already knew that natural languages were an open question.[5] But we thought that making 'Ref-

[1] In fact, 100%-compatible if 'old' .bib files do not include any occurrence of square brackets: see [14] for more details.

[2] E*X*tensible *S*tylesheet *L*anguage *T*ransformations.

[3] E*X*tensible *M*arkup *L*anguage.

[4] *W*orld *W*ide *W*eb *C*onsortium.

[5] In particular, we confess that reading [28] greatly impressed us about the diversity and relationships among natural languages.

```
@BOOK{zelazny1969,
        AUTHOR = {Roger~Joseph Zelazny},
        TITLE = {Damnation Alley},
        PUBLISHER = {Putnam},
        YEAR = 1969,
        NOTE = {[Titre de la premi\‘{e}re traduction fran\c{c}aise : ‘‘Les culbuteurs de l’enfer’’] !
                french},
        LANGUAGE = english}

@BOOK{zelazny2000,
        AUTHOR = {[Roger~Joseph Zelazny] : english},
        TITLE = {Route~666},
        PUBLISHER = {Deno\"{e}l},
        NUMBER = 666,
        SERIES = {Pr\’{e}sence du futur},
        YEAR = 2000,
        MONTH = apr,
        NOTE = {Nouvelle traduction de ‘‘[Damnation Alley] : english’’ par Thomas Bauduret}
        LANGUAGE = french}
```

Fig. 2: Examples of multilingual features in MlBibTeX.

erences’ sections for a word processor was a narrow domain with quite simple rules, so we were able to build such references by means of substitutions applied to keywords such as *and*, *chapter*, *editor*, ... and substitutions applied to delimiters for quotations: “‘...’” in British English, ‘« ... »’ in French, ‘ „...“ ’ in German, and so on. Anyway, we thought that European languages were sufficiently close each to others to put this solution into action. So, the first version (1.1) of MlBibTeX [14, 15] was designed this way. Even though this first version worked for several languages, we quickly realised that using only substitutions was incorrect. The simplest example is given by dates: in French, English and German, the month name within a date is put first, before the year. In Hungarian, the year comes first:[6]

September 2003	(English & German)
septembre 2003	(French)
2003 szeptember	(Hungarian)

We were able to fix this problem by extending the `format.date` function within bibliography styles. Specifically, we were able to add a test about the language used when we are building a reference. But let us recall that the bst language is not modular: several bibliography styles cannot share the definition of a single function put in one file. At this time we decided to enrich our implementation of the bst language by adding an ‘`INCLUDE`’ directive. This statement would look for a file and get access to the bst definitions included in it. But adding this directive was like applying a patch. Besides, although we were able to provide a correct multilingual implementation of the `format.date` function for most languages including Hungarian, at least no one but Hungarians signalled an error, we could see that we might have to rewrite it if there existed a language with a special way to put dates. The *truly* modular and progressive solution to this problem was to be able to define a *standard* way to put dates:

```
<function name="format.date">
  ...
</function>
```

and a *specific* function for languages putting dates in another way:

```
<function name="format.date"
          language="hungarian">
  ...
</function>
```

We wrote skeletons using an XML-like syntax as above, but at this stage, we had not decided yet to use a XML-like language for bibliography styles. Our idea was just to improve the bst language. To be able to perform experiments for this goal, we developed an interpreter of bst using Scheme. In particular, this functional language being interactive, this prototype would help users learn the bst language in an incremental and interactive way [17].

But when we demonstrated this prototype at the DANTE[7] conference in the fall of 2002, we were told

[6] In fact, we wrongly put a date in Hungarian in [16]. We thank Gyöngyi Bujdosó, Peter Szabó and Ferenc Wettl, who informed us of this mistake and the correct way to put a date in Hungarian.

[7] *Deutschsprachige Anwendervereinigung TeX e.V.*, TeX group of German-speaking users.

what we already knew: the bst language has difficult syntax, it is not user-friendly, and its variables are global only — it is wholly related to handling a stack. As a practical matter, it is easy to introduce small changes in an existing bibliography style (as shown in [9, § 13.8]), but programming the whole of a new bibliography style is tedious. Thus, why shall we enrich such a language? To make programming in it more difficult? In addition, there were already attempts to replace it: a new bibliographic module [13] suitable for ConTEXt, Hans Hagen's format [11], a reimplementation of BIBTEX in Common Lisp [26], an interpreter of the bst language in Java [25], and works based on converting bibliographical (.bib) files to XML files [10].

For the rest of 2002, we explored three directions. First, an extended syntax for person and organisation names within fields such as AUTHOR and EDITOR, was tried in MlBIBTEX's Version 1.2[8] [18]. Second, a project done by two graduate students extended BIBTEX so that it could process bibliography files in Unicode [34]. Our goal was not to write a new program which would be practically usable, but only to sum up all the problems related to using Unicode. The results of this project are given in [24].

Third, we again considered the DTD[9] given in [16]. Originally, we designed this DTD in order to compare MlBIBTEX's expressive power to a specification of multilingual bibliographical entries designed from scratch with tools originating from XML. We tried to use XSLT as a new language for bibliography styles, and XSL-FO[10] [40] to build printable outputs for bibliographies. In both cases, we took advantage of the pattern-matching XSLT provides via the match attribute of the xsl:template tag. Also, XPath expressions [35] were very suitable when we were moving within an XML tree for an entry. Still, it was difficult to implement multilinguism, even if XML knows an xml:lang attribute whose value is the code of a language. In particular, it was difficult to implement a refinement for a particular language, as shown above for the format.date function. Using modes (by means of the mode attribute of the xsl:template tag) provided by XSLT was not very convenient, and incorporating tests about the language used would have complicated our templates just like the same kind of tests in bst programs.

In addition, there were other drawbacks: for example, the problems caused by text nodes with white-space characters [36, § 3.4], although this only affected the readability of generated files. As a second example, some operations were difficult by using the functions provided by XPath: abbreviating a first name, implementing the change.case$ function of BIBTEX, that converts a string to lowercase characters only or uppercase characters only. On the other hand, XML is becoming standard, and we thought that developing a new version close to XML was interesting.

That is why we decided to define a new language, close to XSLT, but not equal to XSLT. This new language is very close to XSLT, so we think that learning it should be easy for people knowing XSLT. Besides, work on defining further versions of XSLT is still in progress, as is defining interfaces between XML trees and programming languages. Likewise, we hope that the development of nbst is not finished yet. Thus we can think that in a future version, nbst will become equal to a new version of XSLT. Likewise, we are conformant to the paths specified in XPath, but defined our specific functions: for example, the abbreviate function for abbreviating a first name.[11] Dealing with strings should be eased in Version 2.0 of XPath [42], so a future implementation of MlBIBTEX may be fully conformant to the XPath library. Anyway, before joining MlBIBTEX and XSLT, our work shows what is needed — and not provided by XPath and XSLT — to put a multilingual bibliography program into action nicely.

Within this new framework, the result of parsing a .bib file is a tree built according to DOM,[12] the model issued by the W3C to represent an XML document as a tree (cf. [33, p. 306–308] or [37]). As an example, Figure 3 shows the representation w.r.t. DOM of the entry zelazny1969 given in Figure 2. Let us notice that this choice allows us to avoid the problems caused by blank text nodes, because our parser rules out whitespace characters between two fields' specifications.

Questions and answers

Readers interested in typographical conventions regarding the typesetting of bibliographies can consult [4, § 15.54–15.76] or [12, p. 53–54]: these books are good references for the English language and give some information about bibliographies in languages other than English. But this information is not comparable to what can be found in a book specific to a particular language. Nevertheless, we do not know every language, no more than anyone else. On another point, we got some experience after developing the first version of MlBIBTEX. To learn more about bibliographies, we wrote a questionnaire about keywords and usages in bibliographies. An abridged version is given in Figure 4. The answers we have received do not cover all the European languages, but we think that the range is representative.

[8] This version was only a prototype. It has not been distributed publicly.

[9] *D*ocument *T*ype *D*efinition.

[10] E*X*tensible *S*tylesheet *L*anguage — *F*ormatting *O*bjects.

[11] All these functions are documented in [22].

[12] *D*ocument *O*bject *M*odel.

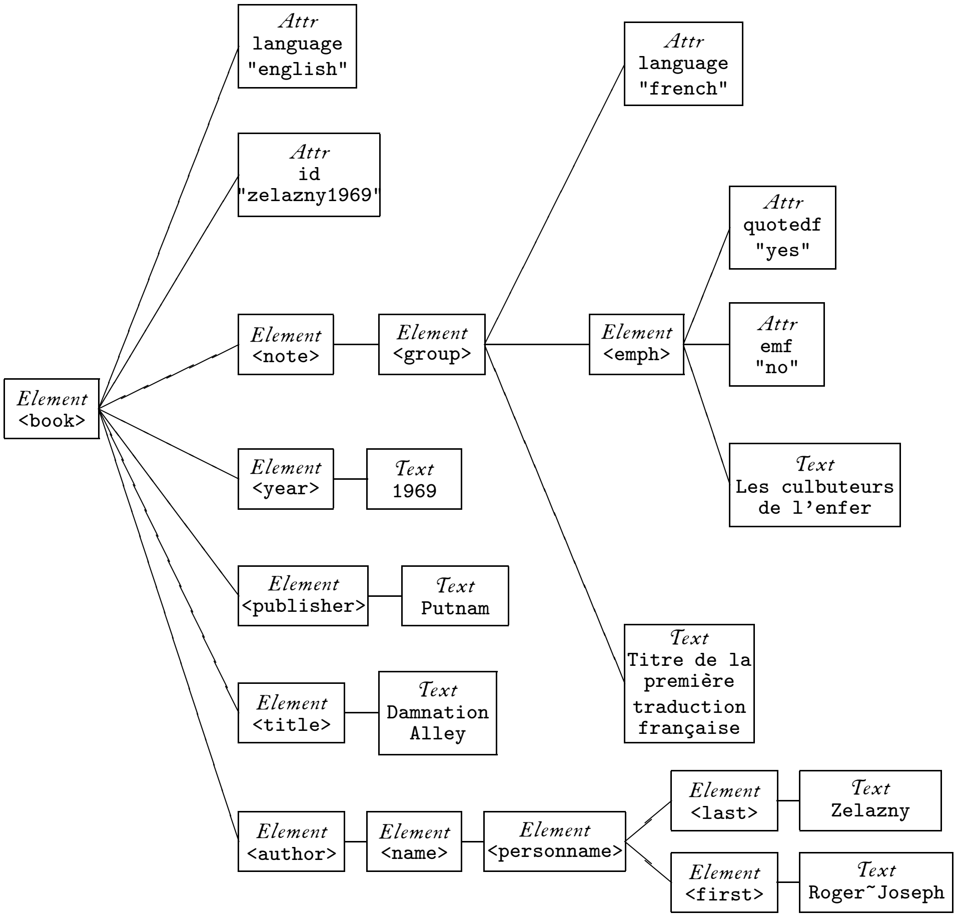

Fig. 3: A bibliographical entry as a tree according to DOM.

First of all, the notion of required and optional field is not the same in every language: for example, the publisher of a book is not required within a bibliography in Hungarian, but in contrast, the address where this book came out should be given, just before the year.

Usually, the translation of keywords does not cause problems — once we know the right word, of course. It can be solved by using LaTeX commands, for example:

```
\def\bbland{and}   (English)
\def\bbland{et}    (French)
\def\bbland{und}   (German)
\def\bbland{i}     (Polish)
...
```

The commands \addto and \extras⟨language-name⟩, provided by the babel package [2, § 6] are very suitable for implementing this kind of multilinguism:

```
\addto{\extrasfrench}{...
 \def\bbland{et}%
 ...
}
```

Likewise, the delimiters for quotations depend on languages and may be different from those used in English, but we can adapt the bblquotedtitle environment, already used in MlBibTeX. We show in [23] how to use this environment for first-level quotations as well as second-level ones, like:

```
\begin{bblquotedtitle}
Symphony no.~3
 \begin{bblquotedtitle}
 Symphony of Sorrowful Songs
 \end{bblquotedtitle}, op.~36
\end{bblquotedtitle}
```

which yields:

> “Symphony no. 3 ‘Symphony of Sorrowful Songs’, op. 36”

in English. Let us notice that opening and closing this environment may do nothing: for example, the Russian language does not use quotation marks in bibliographies.

The only keyword that sometimes has no equivalent is ‘of’, as in:

No. 8 of *Doc Savage* Series

This quiz aims to help me put down the part of MlBibTeX's styles related to your language. [...]

Please translate the following expressions (you can give an abbreviation, provided that it is well-known):

and
and others
before Christi
chapter
edited by
edition
editor
editors
in (*see below*)
Master's thesis
numero
of
(*one*) page
(*several*) pages
PhD thesis
technical report
translated by
volume
with

The '*in*' keyword introduces the title of a general work when you cite only a part. For example:

> Greg Egan. 'Oracle'. In *the Mammoth Book of Best New SF* 14, edited by Gardner Dozois, p. 465–504. Robinson, London, 2001.

As you can see in the example above, the 'enclosing' work's title is italicised in an English-speaking bibliography, whereas the 'enclosed' work is given between quotation marks. How do you put titles of works extracted from a more general work in your language?

How are ordinal numbers abbreviated in your language? For example, in English, the abbreviations are '...st' for '...1', '...nd' for '...2', '...rd' for '...3', '...th' otherwise.

Make precise how dates—as they should appear within a bibliographical reference—are put down in your language. Give some examples:

- a complete date, e.g., 26th June 2002: ...
- a date with only a month name and a year, e.g., June 2002: ...
- a date with only a year, e.g., 2002: ...

How quotations are put in your language? This question is about top-level quotations as well as enclosed quotations. In other words, how would you write the following quotation in your language?

'Symphony no. 3 "Symphony of Sorrowful Songs", op. 36'

How should the following enries be arranged? That is, which fields are viewed as quotations? which are italicised? etc. Hereafter we give some examples: entries for a manual, an article in a journal, a book, a technical report issued by an institution. [...]

Do you think that other fields could be added, in connection with your language? If yes, describe them shortly.

How are your letters alphabeticised? By 'letters', we mean letters with or without accents. [...]

Fig. 4: Our questionnaire.

because some languages use *declensions*[13] and do not need this preposition. Often a sign is used in this case: for example, the dash character ('—') in Russian bibliographies. As another example, the order is reversed in German and Hungarian bibliographies:

Perry Rhodan 5

We can observe the same in French, even though the 'of' word is translatable in this language.

Declensions may appear in a subtle form within dates: for example, 'September 2003' is translated by 'сентябрь 2003' in Russian, but '1*st* September 2003' is said '1st *of* September 2003',[14] and a different case is

[13] A language uses *declensions* when a word's function is directly expressed within this word, most often by a suffix. Originally, it seems that declensions existed in all languages, and they have disappeared in some modern languages: for example, in English, French and Portuguese. Other languages such as Estonian, Czech, German, Greek, Hungarian, Russian, Polish, Slovak ... still have declensions. To illustrate this with some examples in German:

- '*mein Freund*' is for 'my friend' as the subject of a sentence:

  Mein Freund wohnt in Brest.

  (my friend lives at Brest)
- when 'my friend' is used as a direct object:

  Ich besuche meinen Freund.

  (I am visiting my friend)
- when 'my friend' is for a person who receives something:

  Ich gebe meinem Freund ein Geschenk.

  (I am giving a gift to my friend)
- when 'my friend' is the owner of something:

  Der Wagen meines Freundes.

  (the car of my friend).

[14] Let us remark that we can observe the same in Portuguese: '1 *de Setembro de* 2003'.

used for the month name: ‘1 сентября 2003’. The Polish language presents the same feature: ‘*wrzesień* 2003’ but ‘1 *września* 2003’; so does the Greek language. This is not a problem for ‘usual’ references in scientific texts where the precise day is usually not given, but let us assume that we are citing articles in a daily newspaper: in such a case, writing the day number down is important and a new field should be added for that. To implement this feature, we need two LaTeX commands per month name. Focusing on the Polish language, that can be done by a mode attribute, which behaves as in XSLT:

```
<nbst:template match="sep"
               language="polish">
  wrzesie\'{n}
</nbst:template>
```

```
<nbst:template match="sep"
               language="polish"
               mode="genitive">
  wrze\'{s}nia
</nbst:template>
```

Some expressions may depend on a *gender*: in Romanian, ordinal numbers end with ‘ul’ in the masculine or ‘a’ in the feminine. The same feature holds in Greek and Portuguese. Here too, the problem is easily solved because ordinal numbers are usually given for an edition number, so we can know the grammatical gender of this word. On the contrary, the Polish form for ‘translated by’ is ‘*przełożył*’ for a masculine translator and ‘*przełożyła*’ for a feminine one, so we cannot know the right expression *a priori*. We confess that we have not found yet a satisfactory solution to this problem.

On another point, the English language uses several suffixes for ordinal numbers: ‘st’, ‘nd’, ‘rd’, ‘th’. The French language uses* ‘*er*’ and ‘*re*’ within ‘1er’ and ‘1re’ in the masculine and feminine, ‘…^{e}’ for all the other cases. Other languages systematically put a period … after the number (‘1.’, ‘2.): this is the case in Czech, Estonian, German, Hungarian, Slovak, Polish. In fact, we could consider that adding a period is the general method and refine it for English and French.

Let us consider the alphabetical order among words: there are two distinct points of view about diacritical signs.[15] In some languages — French, German, Greek, Hungarian, Italian, Irish, Spanish, Portuguese, … — a letter with an accent or other diacritical mark is regarded as a variant of the same letter without any mark. So, this letter and its variants are alphabeticised at the same place, except that the ‘bare’ letter takes precedence over an accented letter. Other languages — Czech, Estonian, Polish, Romanian, Slovak, Swedish, … — take letters with diacritical marks as distinct from ‘bare’ letters. This convention is the only common point among these languages: in Polish, ‘ą’ is placed just after ‘a’, whereas ‘ä’ is placed at the end of the alphabet in Estonian. In fact, each of these languages has its own order.[16]

In addition, some groups of letters may be regarded as one, and words beginning with such groups are listed separately in a dictionary, which complicates the algorithm for the alphabetic order. The ‘ch’ and ‘ll’ groups in Spanish are well-known examples. This is also the case for ‘cs’, ‘dz’, ‘dzs’, ‘gy’, ‘ly’, ‘ny’, ‘sz’, ‘ty’, ‘zs’ in Hungarian. In fact, the main problem is not the adaptation of the alphabetical order for a particular language, which is not difficult once this order is known, but the definition of a universal order when references originating from several countries have to coexist. This last point appears to be impossible.

MlBibTeX's new version

As mentioned in the section ‘A gulp of history’, parsing a .bib file results in a tree built according the DOM model within MlBibTeX's Version 1.3. For the sake of compatibility with previous .bib files, the delimiters for a quotation are those used in American English. For example, let us once again consider the entry zelazny1969 in Figure 2. The first French title is surrounded by double quotes (“…”) and this quotation is implemented by an element in the DOM tree, as we saw in Figure 3. Some LaTeX commands used within .bib files — \emph, \textbf, etc. — are processed in an analogous way.

We implemented the nbst language by getting some experience with public implementations of XSLT. We also implemented our version of XPath with our specific functions. An example of a template formatting dates is given in Figure 5. The effect of the nbst tags can easily be guessed from the corresponding tags in XSLT. This template has to be completed by pattern-matching on month names, for example:

```
<nbst:template match="jan">
  \bbljan{} <!--  This LaTeX command should
                  be defined for the current
                  language.   -->
</nbst:template>
```

and can be redefined for the Hungarian language — or for any language — by using a language attribute as shown below:

[15] The English language does not have this problem: only the grave accent is sometimes used in poetry and the diaeresis in some old texts.

[16] Among all the answers to our questionnaire, the language that seemed to us to be the most unusual example about letter ordering is Estonian, where ‘z’ and ‘ž’ are alphabeticised before ‘t’, ‘u’, ‘v’, ‘w’.

```
<nbst:template name="format.date">
  <nbst:choose>
    <nbst:when test="year">
      <nbst:choose>
        <nbst:when test="month">
          <nbst:apply-templates
            select="month"/>
          <nbst:text> </nbst:text>
          <nbst:value-of select="year"/>
        </nbst:when>
        <nbst:otherwise>
          <nbst:value-of select="year"/>
        </nbst:otherwise>
      </nbst:choose>
    </nbst:when>
    <nbst:otherwise>
      <nbst:if test="month">
        <nbst:warning>
          There's a month but no year in
          <nbst:value-of select="@id"/>
        </nbst:warning>
        <nbst:apply-templates select="month"/>
      </nbst:if>
    </nbst:otherwise>
  </nbst:choose>
</nbst:template>
```

Fig. 5: Putting a date with nbst.

```
<nbst:template name="format.date"
               language="hungarian">
  ...
</nbst:template>
```

As we discussed above, in the section 'Questions and answers', European bibliographies use a common framework, but we have to be able to use a special method for a particular language. To do that, a template with the `language` attribute has higher priority than a template without. This feature can also be useful to implement stylistic differences, that is, titles to be written using italicised characters instead of quotations or *vice versa*.

The implementation of the difference between language- and document-dependent approaches uses an attribute, too. Let us assume that the entries to be put in the bibliography of a document are the children of a tree. In 'classical' XSLT, such children can be processed recursively by means of the `xsl:apply-templates` tag. The same behaviour holds within nbst, but it is ruled by a so-called `use-language` attribute.

- The following statement:

  ```
  <nbst:apply-templates
    use-language="$document-language"/>
  ```

  means that the language used to process all the children — that is, the entries — is given by the `document-language` variable,
- whereas assigning the default `*self*` value to this variable:

  ```
  <nbst:apply-templates
    use-language="*self*"/>
  ```

  means that each reference is to be written according to the language of the corresponding entry.

For the sake of compatibility with MlBibTeX's previous versions, the language identifiers used as values of the `language` and `use-language` attributes are unambiguous prefixes of a multilingual package or an option of the babel package.[17]

As far as we know, there is no complete implementation of the `xsl:sort` tag in XSLT, according to the orders used in natural languages for strings. nbst provides partial support by means of the corresponding element, `nbst:sort`, but if this element is used to sort strings including letters outside the current language, the result is unspecified. This should be fixed in later versions.

In order to ease the transition between 'old' bst and nbst, bibliography style designers can call functions written using bst within nbst functions. See [20] for more details.

Future

The distribution of MlBibTeX's Version 1.3 [21] will include support for Czech, Danish, Dutch, English, Estonian, French, German, Greek, Hungarian, Irish, Italian, Norwegian, Polish, Portuguese, Romanian, Russian and Slovak. In addition, we explain in [23] how to add the support for a new language.

As for future directions, we think that MlBibTeX should be usable to generate bibliographies for other formats than LaTeX, and now we are investigating this for ConTeXt [11]. In particular, this will cause an adaptation regarding language identifiers, since ConTeXt deals with ISO codes for languages and countries (`en`, `fr`, `de`, ...) [1]. Another possible word processor is Texinfo [3], used for the documentation of GNU[18] software. Likewise, using a language close to XSLT for bibliography styles should ease building bibliographies for the DocBook format [43]. We are also interested in experiencing with Ω [31], as we could fully use this last program when we are able to deal with .bib files using Unicode.

Bibliographical entries can be organised into a data base, so we plan to rewrite the XML specification of MlBibTeX entries with XML Schema [33, p. 189–193],

[17] This choice of a non-ambiguous prefix allows a language identifier to get access to several ways to process a language. For example, a language identifier set to `french` works with the french option of the babel package [2, 6] as well as the french package [7].

[18] GNU's *N*ot *U*nix.

which offers more expressive power than DTDs, especially for data bases.

Another improvement, useful when several bibliographical data bases are shared by several people, would be the addition of *namespaces*, already present in XML [41]. To do that, we plan to use DOM Level 2 [39], and we will have to extend the syntax of .bib files.

Conclusion

We think that an important step has been made: replacing the bst language by a new language close to a standard, and allowing multilinguism within bibliography styles. This transition should be graceful for end-users since they can reuse their .bib files. Obviously, the present program has to be extended and improved: anyway, in order to reach 'actual' multilinguism, we have to include non-European languages. But the present version is ready for use, so we now expect constructive feedback.

Acknowledgements

I am pleased to thank all the people who answered my questionnaire and helped me go thoroughly into this notion of multilingual bibliographies. I confess that I asked some of them for more and more questions, I hope that I did not waste their time. So for being able to write this article, I am indebted to Janusz S. Bień, Giuseppe Bilotta, Gyöngyi Bujdosó, Kaja P. Christiansen, Peter Flynn, Hans Hagen, Zoltán Kása, Ольга Кушнаренко, Dag Langmyhr, Gyula A. Mayer, Paweł D. Mogielnicki, Simon Pepping, Horia F. Pop, Enrico Puppo, Pedro Quaresma de Almeida, Enn Saar, Petr Sojka, Seán Ua Súilleabháin, Ἀπόστολος Συρόπουλος, Marcel Takác. Thanks to the Centre of Teaching Computer Science of the University of Franche-Comté, which has supported most of the trips related to this work. Last but not least, I thank Yannis Haralambous (Γιάννης Χαραλάμπους), who kindly made this complete version available for the conference's audience, although I finished it late. Thanks also to this last version's proofreader, Karl Berry.

References

[1] Harald Tveit Alvestrand: *Request for Comments:* 1766. *Tags for the Identification of Languages.* UNINETT, Network Working Group. March 1995. http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1766.html.

[2] Joannes Braams: *Babel, a Multilingual Package for Use with LaTeX's Standard Document Classes.* Version 3.7. May 2002. CTAN:macros/latex/required/babel/babel.dvi.

[3] Robert J. Chassell and Richard M. Stallman: *Texinfo. The GNU Documentation System.* http://www.gnu.org/software/texinfo.

[4] *The Chicago Manual of Style.* The University of Chicago Press. The 14th edition of a manual of style revised and expanded. 1993.

[5] Robert Eckstein and Michel Casabianca: *XML Pocket Reference.* 2nd edition. O'Reilly & Associates, Inc. April 2001.

[6] Daniel Flipo: *A babel Language Definition File for French.* Version v1.5e. March 2001. http://www.tex.ac.uk.

[7] Bernard Gaulle : *Notice d'utilisation du style french multilingue pour LaTeX.* Version pro V5.01. Janvier 2001. CTAN:loria/language/french/pro/french/ALIRE.pdf.

[8] Michel Goossens: *XML—A New Start for the Web.* May 2000. http://webcast.cern.ch/Projects/WebUniversity/AcademicTraining/Goossens/.

[9] Michel Goossens, Frank Mittelbach and Alexander Samarin: *The LaTeX Companion.* 1st edition. Addison-Wesley Publishing Company, Reading, Massachusetts. 1994.

[10] Vidar Bronken Gundersen and Zeger W. Hendrikse: *BibTeX as XML Markup.* January 2003. http://bibtexml.sourceforge.net.

[11] Hans Hagen: *ConTeXt, the Manual.* November 2001. http://www.pragma-ade.com.

[12] *Hart's Rules for Composers and Readers at the University Press.* Oxford University Press. 39th edition. 1999.

[13] Taco Hoekwater: "The Bibliographic Module for ConTeXt". In: *EuroTeX* 2001, p. 61–73. Kerkrade (the Netherlands). September 2001.

[14] Jean-Michel Hufflen: "MlBibTeX: a New Implementation of BibTeX". In: *EuroTeX* 2001, p. 74–94. Kerkrade, The Netherlands. September 2001.

[15] Jean-Michel Hufflen: "Lessons from a Bibliography Program's Reimplementation". In: *LDTA* 2002, Vol. 65.3 of *ENTCS*. Elsevier, Grenoble, France. April 2002.

[16] Jean-Michel Hufflen: "Multilingual Features for Bibliography Programs: From XML to MlBibTeX". In: *EuroTeX* 2002, p. 46–59. Bachotek, Poland. April 2002.

[17] Jean-Michel Hufflen: *Interaktive BibTeX-Programmierung.* DANTE, Herbsttagung 2002, Augsburg. Oktober 2002.

[18] Jean-Michel Hufflen: *Towards MlBibTeX's Versions* 1.2 *&* 1.3. MaTeX Conference. Budapest, Hungary. November 2002.

[19] Jean-Michel Hufflen: *Die neue Sprache für MlBibTeX.* DANTE 2003, Bremen. April 2003.

[20] Jean-Michel HUFFLEN: "Mixing Two Bibliography Style Languages". In: *LDTA* 2003, Vol. 82.3 of *ENTCS*. Elsevier, Warsaw, Poland. April 2003.

[21] Jean-Michel HUFFLEN: *MlBIBTEX—Multilingual BIBTEX*. July 2003. `http://lifc.univ-fcomte.fr/~hufflen/texts/mlbibtex/mlbibtex/`.

[22] Jean-Michel HUFFLEN: "MlBIBTEX's Version 1.3". *TUGboat*, Vol. 24, no. 2, p. 249–262. TUG 2003, Outrigger Waikoloa Beach Resort, Hawaii. July 2003.

[23] Jean-Michel HUFFLEN: "Making MlBIBTEX Fit for a Particular Language. Example of the Polish Language". *Biuletyn GUST*, Vol. 21, p. 14–26. 2004.

[24] Laurent HUMBERTCLAUDE et Bastien SÉNÈQUE : *Adaptation de BIBTEX à l'utilisation d'Unicode*. Rapport de fin de projet DESS. Février 2003.

[25] Ronan KERYELL : *BIBTEX++*. Septembre 2001. `http://www.lit.enstb.org/~keryell/projets/BibTeX++/exposes/2001-09-19/`.

[26] Matthias KÖPPE: *A BIBTEX System in Common Lisp*. January 2003. `http://www.nongnu.org/cl-bibtex`.

[27] Leslie LAMPORT: *LATEX. A Document Preparation System. User's Guide and Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts. 1994.

[28] Jacqueline MANESSY-GUITTON : « La parenté généalogique ». *In* : André MARTINET, rédacteur en chef, *Le langage*, Vol. 25 : *Encyclopédie de la Pléiade*, p. 814–864. Gallimard. 1968.

[29] Oren PATASHNIK: *Designing BIBTEX Styles*. February 1988. Part of BIBTEX's distribution.

[30] Oren PATASHNIK: *BIBTEXing*. February 1988. Part of BIBTEX's distribution.

[31] John PLAICE and Yannis HARALAMBOUS: *Draft Documentation for the Ω System*. March 1998. `http://www.loria.fr/services/tex/english/moteurs.html`.

[32] Bernd RAICHLE: *Die Makropakete „german" und „ngerman" für LATEX 2ε, LATEX 2.09, Plain-TEX and andere darauf Basierende Formate*. Version 2.5. Juli 1998. Im Software LATEX.

[33] Erik T. RAY: *Learning XML*. O'Reilly & Associates, Inc. January 2001.

[34] THE UNICODE CONSORTIUM: *The Unicode Standard Version* 4.0. Addison-Wesley. August 2003.

[35] W3C: *XML Path Language (XPath). Version* 1.0. W3C Recommendation. Edited by James Clark and Steve DeRose. November 1999. `http://www.w3.org/TR/1999/REC-xpath-19991116`.

[36] W3C: *XSL Transformations (XSLT). Version* 1.0. W3C Recommendation. Written by Sharon Adler, Anders Berglund, Jeff Caruso, Stephen Deach, Tony Graham, Paul Grosso, Eduardo Gutentag, Alex Milowski, Scott Parnell, Jeremy Richman and Steve Zilles. November 1999. `http://www.w3.org/TR/1999/REC-xslt-19991116`.

[37] W3C: *Document Object Model (DOM) Level* 1 *Specification (Second Edition). Version* 1.0. W3C Recommendation. Edited by Laurent Wood, Arnaud Le Hors, Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Gavin Nicol, Jonathan Robie, Robert Sutor and Chris Wilson. September 2000. `http://www.w3.org/TR/2000/WD-DOM-Level-1-20000929`.

[38] W3C: *Extensible Markup Language (XML)* 1.0 *(Second Edition)*. W3C Recommendation. Edited by Tim Bray, Jean Paoli, C. M. Sperberg-McQueen and Eve Maler. October 2000. `http://www.w3.org/TR/2000/REC-xml-20001006`.

[39] W3C: *Document Object Model (DOM) Level* 2 *Core Specification (Second Edition). Version* 1.0. W3C Recommendation. Edited by Arnaud Le Hors, Philippe Le Hégaret, Laurent Wood, Gavin Nicol, Jonathan Robie, Mike Champion and Steve Byrne. November 2000. `http://www.w3.org/TR/2000/WD-DOM-Level-2-Core-20001113`.

[40] W3C: *Extensible Stylesheet Language (XSL). Version* 1.0. W3C Recommendation. Edited by James Clark. October 2001. `http://www.w3.org/TR/2001/REC-xsl-20011015/`.

[41] W3C: *Namespaces in XML* 1.1. W3C Recommendation. Edited by Tim Bray, Dave Hollander, Andrew Layman and Richard Tobin. April 2002. `http://www.w3.org/TR/xml-names11/`.

[42] W3C: *XML Path Language (XPath)* 2.0. W3C Working Draft. Edited by Anders Berglund, Scott Boag, Don Chamberlin, Mary F. Fernandez, Michael Kay, Jonathan Robie and Jérôme Siméon. November 2002. `http://www.w3.org/TR/2002/WD-xpath20-20021115`.

[43] Norman WALSH and Leonard MUELLNER: *DocBook: The Definitive Guide*. O'Reilly & Associates, Inc. October 1999.