A comparative study of methods for bibliographies

Jean-Michel Hufflen

Abstract

First, we recall the successive steps of the task performed by a bibliography processor such as BibTeX. Then we sketch a brief history of the successive methods and fashions of processing the bibliographies of (LA)TeX documents. In particular, we show what is new in the LaTeX 2_ε packages natbib, jurabib, and biblatex. The problems unsolved or with difficult implementations are listed, and we show how other processors like Biber or MlBibTeX can help.

Keywords Bibliographies, bibliography processors, bibliography styles, Tib, BibTeX, MlBibTeX, Biber, natbib package, jurabib package, biblatex package, sorting bibliographies, updating bibliography database files.

Introduction

As mentioned at the beginning of "'Bibliography Generation", the 13th chapter of *The LaTeX Companion*'s Second Edition [35], the items of a printed document's bibliography may be composed manually, but this method is not recommended, since the result may not be reusable within another context. Indeed, bibliography layouts are very diverse: a publisher may require that authors' names are written *in extenso* as far as possible, whereas another prefers for first names to be abbreviated using only initials, etc. So the best way to deal with bibliographies is the use of *bibliography database files*, containing the whole information about bibliographical items. These database files are searched by a *bibliography processor*, which builds 'References' sections for printed or online documents.

As we recall below, BibTeX [38] was unrivalled for a long time as the bibliography processor used in conjunction with the LaTeX word processor. Now the landscape is changing and other comparable programs have come out. So this article aims to focus on the directions taken by BibTeX and its possible successors. In [19], we compared the programming languages used to design *bibliography styles*, controlling bibliographies' layout. The present article's purpose is different: we are interested in the *evolution* of some successive bibliography processors used in conjunction with LaTeX, this evolution still being in progress. First, in Section 1, we delineate the tasks to be performed by such a bibliography processor. We also explain the requirements for how such a program should be updated. Then Section 2 sketches the features of these successive bibliography processors. Finally, a synthesis is given in Section 3. As mentioned above, the present article only covers bibliography processors used in conjunction with LaTeX, it is complemented by [25] about bibliography processors used in conjunction with ConTeXt [11], another format built out of TeX. Reading this article only requires basic knowledge about LaTeX and BibTeX. Of course, the short descriptions we give hereafter do not aim to replace the complete documentation of the corresponding tools. Readers interested in typographical conventions for bibliographies can consult [4, Ch. 10] and [7, Ch. 15 & 16].

1 Tasks of a bibliography processor

In this section, we summarise the tasks to be performed by a bibliography processor such as BibTeX. Along the way, we give the terminology used throughout this article. Then we point out the features that are still unsolved or with difficult implementations. Of course, a bibliography processor works in conjunction with a text processor such as LaTeX or ConTeXt, denoted by 'the word processor' in the following.

End-users can use bibliography database files, containing bibliographical *entries*. The main role of a bibliography processor is to extract the bibliographical *references* of a document from these entries. According to BibTeX's standard use, bibliographical entries (resp. references) are stored in .bib (resp. .bbl) files. Let us notice that in some documents, there is no 'References' section, but rather bibliographical references are given as footnotes wherever they are cited. In other words, bibliographical references exist as *resources* and are intended to be typeset — so the bibliography processor must build them as processable by the word processor — possibly as a section or sparsely. Sometimes there are several 'References' sections, because each chapter of an important book has its own bibliography, or a unique bibliography is divided into several rubrics.

When several bibliographical references are to be grouped into a section, some bibliographies are *unsorted*, that is, the order of items must be the order of first citations of these items throughout the document. In practice, most bibliographies are 'sorted', most often according to first the authors' names,[1] second the dates: in such a case, it is up to the bibliography processor to perform this sort operation. Let us mention that the 'standard' sort given above is not universal: we personally were in charge of

[1] ... or editors' names, when there is no author, for example, for a conference's complete proceedings.

the publication list of our laboratory — the LIFC[2] — when the activity report was written according to the directives given by the AERES:[3] we had to sort this list first by research teams, second by *categories*,[4] third by years decreasingly, fourth by authors' names increasingly, fifth by months decreasingly.

Each bibliographical entry is supposed to be accessible from a *citation key*, that is, citation keys must be non-ambiguous. Source texts written by end-users only contain citation keys to point to bibliographical resources, whereas results typeset by the word processor deal with *bibliographical keys*. It is up to the bibliography processor to build a mapping between citation and bibliographical keys. These bibliographical keys depend on the *system* chosen; as an example, they are positive natural numbers in the number-only system. Sometimes, bibliographical keys are built from the first letters of authors' names, followed by the year and possibly by a letter; so does BibTeX's alpha bibliography style. In some other systems — e.g., the author-date or author-number system — some parts of an entry can identify it obviously: cf. [4, Ch. 10] or [35, Ch. 12] for a survey about these systems. Let us mention that the alpha bibliography style belongs to the number-only system, rather than the author-date one, because bibliographical keys are univoque — like natural numbers — and *atomic* in the sense that you cannot divide them into an author and year parts. In other words, they actually work like natural numbers, up to an isomorphism.

Given a document, rules governing the layout of bibliographical references and citations, including ordering bibliographical items in a 'sorted' 'References' section, comprise a *bibliography style*.

What we have expressed above could have been put down when BibTeX was designed and put into action, in the 1980s. Since that time, some additional requirements have appeared. First, the character encoding that was most commonly used at that time and for a long period was ASCII,[5] 7-bit based. Later, some 8-bit extensions — such as Latin 1 or Latin 2[6] — allowed some additional characters used in non-English languages to be included. Then a universal character encoding, Unicode [43], was designed, including some *formats*[7] — such as UTF-8 and UTF-16 — that allow the complete set of Unicode characters to be represented by byte or double-byte sequences. A modern bibliography processor should be able to deal with all these different encodings, in particular UTF-8, which is becoming more and more common. Another point related to multilinguism may be viewed as a particular case of *software localisation*. Let us give an example about person names: first names are usually put before last names in most languages written with the Latin alphabet; as a counter-example, that is not the case for the Hungarian language, where last names come first; a style suitable for bibliographies of documents written in Hungarian should take this point into account. Regarding the document's language, some information included in bibliographical entries should be included or discarded. For example, a transliteration of titles of works in Russian written with the Cyrillic alphabet may be of interest for a document in English, but would be useless for a document in Russian.

A modern bibliography processor should generate source texts for word processors that typeset documents to be printed — as mentioned above — but should also be able to build bibliographies for online documents. Besides, let us recall that for several years, the XML[8] metalanguage has become a central formalism for data interchange in general and for production process of documents in particular. In other words, a bibliography processor should be able to deal with languages using XML-like syntax — a good example is XSL-FO[9] — and languages for the Web, such as (X)HTML.[10]

Last but not least, a bibliography processor for (LA)TeX source texts should be able to deal with bibliography database (.bib) files written according to BibTeX's format, because of backward compatibility. As proof of this program's success, there is a *huge* number of such files in end-users' directories. However, that may be also viewed as *legacy*. Anyway, if another format for bibliographical entries was adopted, a converter from .bib files into this new format would be needed.

[2] ***L****aboratoire d'****I****nformatique de l'université de* ***F****ranche-****C****omté*.

[3] ***A****gence d'****É****valuation de la* ***R****echerche et de l'****E****nseignement* ***S****upérieur*, that is, 'agency evaluating research and university courses'.

[4] That is, articles in well-known international journals and in other international journals, articles in journals having 'national' scope, papers in conferences, etc.

[5] **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.

[6] More details about these encodings can be found in [35, § 7.5.2].

[7] 'UTF' stands for '**U**nicode **T**ransformation **F**ormat'.

[8] E**X**tensible **M**arkup **L**anguage. Readers interested in an introductory book to this formalism can consult [41].

[9] E**X**tensible **S**tylesheet **L**anguage — **F**ormatting **O**bjects. This XML dialect aims to describe high-quality output prints. See [40] for an introduction to it.

[10] (E**X**tensible) **H**yper**T**ext **M**arkup **L**anguage. XHTML is a reformulation of HTML — the original language of Web pages — using XML conventions. [36] is a good introduction to these languages.

```
%A Mike Newton
%T Resurgence
%I Worldwide Library
%S Don Pendleton's Mack Bolan
%N 141
%D |APR| 2011
%K additional key
```

Figure 1: Example using the Refer format.

2 A little bit of history

2.1 Early attempt

As far as we know, the first bibliography processor was Refer, used with the troff[11] text processor, already present within the first versions of the Unix system [2]. As shown in Fig. 1, the Refer format for bibliographical entries — used within .ref files — is line-oriented. Unlike BibTeX's format, it does not provide explicit information about entry *types*, such as `article`, `book`, ... Such information is to be determined dynamically, when entries are processed. A similar bibliography processor for TeX source files using this Refer format has been developed: Tib [1]; it seems that Tib has been used mainly for *Plain TeX* source texts. Tib was written in C [28]; like Refer, it is a *preprocessor* in the sense that the source text Tib processes may contain *incomplete citations*, surrounded by '`[.`' and '`.]`':

```
 ... see [.mack bolan resurgence.], ...   (1)
```

and such citations are *replaced* by bibliographical keys within the result built by Tib:

```
... see \Lcitemark Newton\Citebreak
2011\Rcitemark, ...
```

'`.[]`' at the beginning of a line within the source text processed by Tib is replaced within the result by the source text for the 'References' section. Fig. 1's entry would appear inside such a section as shown in Fig. 2. It can be seen that such a reference uses some TeX commands introduced by Tib to memoize values associated with fields. Then the `\Refformat` command formats the complete reference. Tib provides some styles to customise this operation. Likewise, Tib allows citation keys to be numbers or alpha-like keys, and provides additional commands such as `\Lcitemark` and `\Rcitemark` to control citation keys' layout.

[11] Several steps are needed to make clear this name's etymology. One of the first text formatting programs was runoff — for 'I'll run off a document' — written in the mid-1960s. When this program was adapted in 1969, the new name was abridged in 'roff'. Further reimplementations of roff were called 'nroff' — for '**N**ewer **ROFF**' — and 'troff' — for '**T**ypesetter **ROFF**'. There is a modern version of this program, groff, provided by the GNU (**G**nu's **N**ot **U**nix) project. The groff package includes a new version of Refer.

```
{\Resetstrings%
 \def\Loccittest{}\def\Abbtest{}%
 \def\Capssmallcapstest{}\def\Edabbtest{}%
 \def\Edcapsmallcapstest{}\def\Underlinetest{}%
 \def\NoArev{0}\def\NoErev{0}\def\Acnt{1}%
 \def\Ecnt{0}\def\acnt{0}\def\ecnt{0}%
 \def\Ftest{ }\def\Fstr{10}%
 \def\Atest{ }\def\Astr{Mike Newton}%
 \def\Ttest{ }\def\Tstr{Resurgence}%
 \def\Itest{ }\def\Istr{Worldwide Library}%
 \def\Stest{ }%
 \def\Sstr{Don Pendleton's Mack Bolan}%
 \def\Ntest{ }\def\Nstr{141}%
 \def\Dtest{ }\def\Dstr{April 2011}%
 \Refformat}
```

Figure 2: Reference generated by Tib.

When Tib processes the argument of an incomplete citation, it truncates each word to 6 characters, and looks for a unique entry including all these truncated words as prefixes. For example, the entry given in Fig. 1 matches since it includes the three words '`mack`', '`bolan`', '`resurg`', given in (1). You can use keys at the lines labelled by `%K` — values associated with this field are not printed out in references — but in practice, many Tib users put authors' names and significant words of a title in incomplete citations. *Symbols* can be defined: in Fig. 1, we use this feature as a workaround that allows the month information to be put or not, depending on the value associated with the `APR` symbol. Let us notice that Tib can produce sorted bibliographies: when you call Tib, you can specify a first field for a primary sort key, a second field for a second sort key, and so on. However, only lexicographical sorts are possible: for example, 2 comes after `1999`! Of course, this point is not very important in practice because years coming from 'actual' bibliography database files are often close to each other; it is rare to include entries for documents written in the 1st and 20th centuries, but in such a case, the sort operation would fail.[12] In addition, let us recall that values associated with `%D` fields are supposed to be *dates*. From a theoretical point of view, some accurate encoding would allow complete dates — years, months, days — to be sorted but this operation is quite tedious and in practice, only years are used.

2.2 BibTeX's age

The first edition of the LaTeX manual included an introduction to BibTeX; [32, App. B] reads:

[12] This error disappears if '2' is replaced by '`0002`' in the values associated with the `%D` field. But that causes '`0002`' to be printed in the generated document processed by (LA)TeX, so it is an imperfect workaround.

```
@BOOK{newton2011,
        AUTHOR = {Mike Newton},
        TITLE = {Resurgence},
        SERIES = {Don Pendleton's Mack Bolan},
        NUMBER = 141,
        PUBLISHER = {Worldwide Library},
        TOTALPAGES = 320,
        MONTH = apr,
        YEAR = 2011}
```

Figure 3: Example using BibTeX's format.

> *Once you learn to use BibTeX, you will find it easier to let BibTeX make your reference list than to do it yourself. [...]*

The BibTeX program was written in the WEB system, used to program TeX's kernel.[13] In fact, BibTeX was initially designed to work in conjunction with the SCRIBE word processor[14] [42]. That is why the markup of .bib files is introduced with an '@' sign: this convention originates from SCRIBE.[15] Fig. 3 is the specification, for BibTeX, of the entry given in Fig. 1 in the Refer format.

For many years, BibTeX has been intensively used and was unrivalled as the bibliography processor associated with LaTeX. In comparison with Refer or Tib, BibTeX is not a preprocessor: users keep the same source text before and after running BibTeX. To cite Fig. 3's entry, just put:

`\cite{newton2011}`

inside your source text, as mentioned in any LaTeX manual. In the typeset result, the 'References' section appears where a `\bibliography` command has been put down within the source text.[16] In fact, given a source (.tex) file, BibTeX never reads it, and only parses the corresponding auxiliary (.aux) file generated by LaTeX in order to store information about cross-references[17] [35, § 12.1.3]. BibTeX is a very robust program, very suitable for bibliographical entries concerning works written in English, and whose authors or editors have English or American names. As a proof that BibTeX is widespread, you can find a huge number of .bib files on the Web. In addition, there are many bibliography database management tools based on the .bib format, some graphical, as reported in [35, § 13.4] and [44, § 9.1].

```
\documentclass{article}
\usepackage{natbib}

\begin{document}
\citep{newton2011} is a thriller. The Albanian
Mafia is powerful, as mentioned by
\citeauthor{newton2011}.

\bibliographystyle{plainnat}
\bibliography{mb}  %  That is, Fig. 3.
\end{document}
```

Figure 4: Example using the natbib package.

```
\documentclass{article}
\usepackage{jurabib}
\jurabibsetup{titleformat=italic}

\begin{document}
\citetitleonly{newton2011} is a thriller. The
Albanian Mafia is powerful, as mentioned by
\cite{newton2011}.

\bibliographystyle{jurabib}
\bibliography{mb}
\end{document}
```

Figure 5: Example using the jurabib package.

As mentioned in [35, § 12.1.2], the number-only system is the default method supported by standard LaTeX, even if bibliographical keys may be identifiers, as in the alpha bibliography style. After some attempts [35, § 12.3.1], the author-date system has been implemented successfully by the natbib package — as well as a simplified version of the author-number system [35, §§ 12.3.2 & 12.4] — used in conjunction with accurate bibliography styles. In particular, this package provides an interface with references, in that some commands — e.g., `\citeauthor`, `\citeyear` — can get access to particular fields. An example is given in Fig. 4, and typesetting it looks like this:

> [Newton, 2011] is a thriller. The Albanian Mafia is powerful, as mentioned by Newton.

A more complete interface is provided by the jurabib package [35, § 12.5.1], implementing the author-date and short-title systems. The example given in Fig. 5 results in:[18]

> *Resurgence* is a thriller. The Albanian Mafia is powerful, as mentioned by Newton. (2)

Such an approach is possible since the `\bibitem` command's optional argument — giving a citation

[13] A recent description of the capabilities of this WEB system, related to *literate programming*, is [30].

[14] SCRIBE influenced LaTeX's design by introducing the notion of *document style*, the ancestor of the notion of LaTeX 2_ε's *document class*.

[15] ... and is still followed by Texinfo, the program used to typeset the GNU project's software manuals [6].

[16] ... in most of BibTeX's bibliography styles. A counter-example will be given in § 2.4.

[17] Let *f* be a file name without suffix, 'bibtex *f*' is equivalent to 'bibtex *f*.aux'.

[18] In fact, jurabib's `\citetitleonly` command, used in Fig. 5, uses the contents of the SHORTTITLE field if it is available, the contents of the TITLE field otherwise [35, pp. 719–720].

```
\bibitem[Newton(2011)]{newton2011} Mike Newton.
\newblock \emph{Resurgence}.
\newblock Number 141 in Don Pendleton’s Mack
Bolan. Worldwide Library, April 2011.
\end{document}
```

Figure 6: Reference for the natbib package.

key [35, § 12.1.2] — is structured. This structure remains light for the natbib package (cf. Fig. 6), becomes heavy for the jurabib package (cf. Fig. 7).

As we can see in Fig. 7 about a reference built by a bibliography style suitable for the jurabib package, the text following a `\bibitem` command and its argument is marked up with LaTeX commands. In fact, such a reference inside a bibliography is not directly formatted by the jurabib bibliography style, but this operation is *deferred* to LaTeX, since these commands are defined within the jurabib package. For example, the `\bibtfont` command is used for books' titles.[19] As another example, the `\bibnf` command applies to five arguments representing the components of a person name — *in extenso* and abbreviated — and controls the layout of such a name. Here is the default layout of a reference built by the jurabib bibliography style:

Newton, Mike: Resurgence. Worldwide Library, April 2011, Don Pendleton's Mack Bolan 141

Considering a name of an author, if you want the last name to be typeset using small capitals, followed by the first name surrounded by parentheses when the *von* and *Junior* parts are absent, just redefine the following commands:

```
\renewcommand{\biblnfont}[1]{\textsc{#1}}
\renewcommand{\bibfnfont}[1]{\textrm{#1}}
\renewcommand{\jbNotRevedNoVonNoJr}{%
 \biblnfmt{\jbLast} %
 (\bibfnfmt{\jbCheckedFirst})}
```

BibTeX has some drawbacks, even if they are solved by workarounds. In fact, since BibTeX has been mainly used for texts to be processed by LaTeX, users get used to put LaTeX commands inside values associated with BibTeX fields. That idea is quite good, but the problem is that BibTeX's conventions are not LaTeX's. As a simple example, you can write '`Pierre V\'{e}ry`' within a LaTeX source text, but you must put:

```
AUTHOR = {Pierre V{\'{e}}ry}
```

[19] As shown by Fig. 7, this `\bibtfont` command is used when the reference is typeset. If you want to customise the titles' layout when they appear throughout your text, use the titleformat option of the jurabib package or the `\jurabibsetup` command, as shown in Fig. 5.

```
\bibitem[{Newton\jbdy {2011}}{}%
 {{0}{}{book}{2011}{}{}{}{}%
  {Worldwide Library\bibbdsep {} 2011}}%
 {{Resurgence}{}{}{2}{}{}{}{}{}}%
 ]{newton2011}
\jbbibargs {\bibnf {Newton} {Mike} {M.} {} {}}
{Mike Newton} {au}
{\bibtfont {Resurgence}\bibatsep\ \apyformat
 {Worldwide Library\bibbdsep {} \aprname\ 2011}
 \numberandseries {141}{Don Pendleton’s Mack
  Bolan}} {\bibhowcited} \jbdoitem
{{Newton}{Mike}{M.}{}{}} {} {} \bibAnnoteFile
{newton2011}
```

Figure 7: Reference for the jurabib package.

within a .bib file, especially if you would like to use the alpha bibliography style, as explained in [35, pp. 768–769]. Another example, given in [35, p. 767]:

```
AUTHOR =
  {Maria {\MakeUppercase{de} La} Cruz}
```

in order for the group '`De La`' to be recognised as the name's particle — that is, the *von* part, w.r.t. BibTeX's terminology — even though it does not begin with a downcase letter, as in BibTeX's conventions. In fact, this group's initial uppercase letter will be typeset by LaTeX by means of the predefined command `\MakeUppercase`. From a general point of view, inserting LaTeX commands inside such values is not recommended for .bib files shared by several users or put on the Web, especially if these commands belong to particular packages or should be user-defined. Besides, such commands may be misunderstood by other formats or programs related to TeX, e.g., ConTeXt or LuaTeX [12]. Finally, these commands complicate the conversion of .bib files into HTML pages.[20]

Of course, these drawbacks have appeared only recently in relation to the date of BibTeX's first version. However, a modern version of a format for bibliography database files cannot ignore them. As another drawback, BibTeX — like Tib — provides only lexicographical sorts, as reported in [18], though at least end-users can choose their own sort keys. Last but not least, BibTeX's bibliography styles (.bst files) are written using bst [37], an old-fashioned language using postfixed notations and based on manipulating a stack.

2.3 BibTeX's successors

The BibTeX program has remained stable for more than a decade. A new version (1.0) has been announced in [39], but is not available yet. Other

[20] An example of such a converter is BibTeX2HTML [9]. Other comparable tools are listed in [44, § 9.2].

programs, which have come out quite recently, may be viewed as BibTeX's successors, in the sense that they behave like BibTeX: they use .bib files, look into .aux files. Some aim to replace the bst language of BibTeX by another programming language, more modern.

2.3.1 Programs based on BibTeX

In this section, we consider the programs that do not actually replace BibTeX, because their source files are revisions of BibTeX's, or they need BibTeX when they run.

Whereas BibTeX's original version, written by Oren Patashnik, can only deal with ASCII texts, BibTeX8 [35, § 13.1.1] is a revision of BibTeX that allows end-users to store .bib files using 8-bit codes such as Latin 1 or Latin 2. However, you have to use the same encoding for all the .bib files parsed when you order BibTeX8 to run. In other words, you cannot build a 'References' section by using a .bib file encoded in Latin 1 and a second encoded in Latin 2. In addition, BibTeX8's capacity can be enlarged by means of options, whereas the same operation on BibTeX needs source files to be recompiled.

BibTeXu is another revision of BibTeX, compatible with UTF-8 and integrating sort routines coming from the ICU[21] library. BibTeXu is briefly described in [44, § 4.3].

We include Bibulus [46] in this section because Bibulus needs BibTeX whenever it runs. Bibulus includes the bib2xml bibliography style — written using the bst language — which converts the selected entries into an XML-like format. Then such XML files are processed by a program written using Perl.[22] Whereas BibTeX's bibliography styles written using bst are monolithic programs — identical parts are copied *verbatim* from a style onto another — the features of Bibulus' bibliography styles are controlled by arguments of the \bibulus command:

```
\usepackage{bibulus}
\bibulus{
 ignorevon,cite=alpha,punctuation=/,
 authorfont=\sc}
```

to be put in a LaTeX document's preamble.

2.3.2 Complete reimplementations

These programs aim to replace BibTeX, that is, provide the same service. They are 'complete' reimplementations in the sense that they do not use source files of the BibTeX program. Three allow a .bst

[21] **I**nternational **C**omponents for **U**nicode.

[22] **P**ractical **E**xtraction and **R**eport **L**anguage. A good introduction to this language is [45].

```
(book
 (@ (id "newton2011") ...)
 (author (name (personname (first "Mike")
                           (last "Newton"))))
 (title "Resurgence")
 (publisher "Worldwide Library") (year "2011")
 (month (apr)) (number "141")
 (series "Don Pendleton's Mack Bolan")
 (totalpages "320"))
```

Figure 8: SXML format used by MlBibTeX.

bibliography style to be run. BibTeX++ [8] and cl-bibtex [31] compile it to functions written using their implementation language, Java [26] for the former, ANSI[23] Common Lisp [10] for the latter. This is provided as a compatibility mode; users are encouraged to develop new bibliography styles using these implementation languages.

MlBibTeX[24] [14] is written using the Scheme functional programming language [27] and allows a .bst bibliography style to be interpreted [15]. You can write a bibliography style using Scheme, as we did in [23]; another choice is to use nbst,[25] a language close to XSLT,[26] the language of transformations used for XML documents [41, Ch. 6]. In fact, when MlBibTeX parses a .bib file, the result can be viewed as an XML document, formatted using SXML[27] conventions [29], as shown in Fig. 8. MlBibTeX provides syntactical extensions for .bib files, described in [14]. Most are related to MlBibTeX's features about multilinguism, others ease the specification of person names [16], as shown in Fig. 9.

There is a big difference between BibTeX and MlBibTeX: the latter performs a more precise analysis of .bib files. When a field name is not recognised, a warning message is emitted.[28] That may be viewed as an advantage: if you type '`EDITORS = ...`' instead of '`EDITOR␣ = ...`' inside an entry of type `@INPROCEEDINGS`, MlBibTeX will warn you whereas BibTeX will silently ignore that field. This feature may also be viewed as a drawback: if you specify a `MONTH` field, the associated value must be a symbol among `jan`, `feb`, ..., `dec`. Otherwise, MlBibTeX stops with an error message. This convention may appear as too restrictive,[29] but MlBibTeX can sort

[23] **A**merican **N**ational **S**tandard **I**nstitute.

[24] **M**ulti**L**ingual BibTeX.

[25] **N**ew **B**ibliography **ST**yle.

[26] e**X**tensible **S**tylesheet **L**anguage **T**ransformations.

[27] **S**cheme implementation of XML.

[28] ... but this is just a warning message; the corresponding information is not lost.

[29] If some information about the day of the month is relevant for an entry, some manuals — e.g., [35, § 13.2.3] — recommend to include it into the `MONTH` field. From our point of view,

```
@BOOK{cussler2000,
  AUTHOR = {Clive Cussler with Paul Kemprecos},
  TITLE = {Blue Gold},
  SERIES = {Numa},
  PUBLISHER = {Pocket Books},
  YEAR = 2000}

AUTHOR = {first => Maria, von => De La,
          last => Cruz}

AUTHOR = {Henry Rider Haggard, abbr => H. Rider}

AUTHOR = {John L White, abbr => J. L}

AUTHOR = {org => Word Wide Web Consortium,
          sortingkey => W3C}
```

Figure 9: Syntactical extensions for author names provided by MlBibTeX.

w.r.t. month names[30] whereas BibTeX's standard bibliography styles do not. To perform such an operation, month names must be recognised. Likewise, when years are to be sorted, MlBibTeX applies a numerical sort whereas Tib and BibTeX sort years as strings, as mentioned above. So the value associated with a `YEAR` field must be an integer;[31] otherwise, an error message is emitted.[32] Let us end with mentioning that MlBibTeX's library provides powerful functions for language-defined lexicographical sorts [17, 18] and numerical ones. In particular, MlBibTeX allows successive order keys to be *chained* easily.[33]

As mentioned above, MlBibTeX has been successfully used to process the bibliography of our laboratory's activity report. As explained in [21], this bibliography had to conform with very precise requirements that were not implemented in any bibliography style of BibTeX and would be tedious to program using BibTeX's language. MlBibTeX has been also used to populate the official French site for Open Archives, HAL,[34] as reported in [22, 23, 24]. MlBibTeX can handle (LA)TeX commands that produce accented letters, and it can deal with .bib files using Latin 1 encoding. A future version is planned with other encodings such as Latin 2 or UTF-8.

these two values — months and day numbers — are subject to sort. So mixing them seems to us to be bad technique, unless a precise format is defined, as done within the biblatex package's `DATE` field (cf. § 2.4). Another solution could be the introduction of a `DAY` field.

[30] This information is optional, but MlBibTeX addresses that by means of the function `<month-position`; the Scheme expression (`<month-position` *T default-value*) returns the month's rank if *T* is an SXML subtree containing month information, *default-value* if this information is not supplied. So this default value — which is an integer, in practice — allows us to sort any entry regarding this function's results.

[31] Negative values, for years BCE, are allowed.

[32] More precisely, the standard fields subject to additional check are `AUTHOR`, `EDITOR`, `YEAR`, `MONTH`, and `PAGES`.

[33] Let us consider a simple example with two person names whose last names are l_0, l_1, and first names are f_0, f_1. The expression:

```
(<english? l0 l1 (lambda () (<english? f0 f1)))
```

yields `#t` (resp. `#f`), the true (resp. false) value, if l_0 comes before (resp. after) l_1 w.r.t. the lexicographic order in English. If l_0 and l_1 are equal, the third argument is called. That is, if last names are equal, the comparison focuses on first names. This third argument of MlBibTeX's order relations is optional and defaults to `(lambda () #f)`, as for a strict order relation, that is, irreflexive. MlBibTeX's functions like `<english?` are case-sensitive by default — uppercase letters take precedence — another optional argument allows users to customise this behaviour.

```
@AUTHOR{mb141a, NAME = {Mike Newton}}

@CONFERENCE{tug,
  SHORTNAME = {TUG},
  LONGNAME = {{\TeX} Users Group Conference},
  [YEAR = 2008] ADDRESS = {Cork}, MONTH = jul,
  [YEAR = 2011]
     ADDRESS = {Trivandrum}, MONTH = oct}
```

Figure 10: Extensions recognised by CrossTeX.

CrossTeX [3] is written in Python [34] and implements a kind of object-oriented paradigm about bibliography database (.xtx) files. Such files look like .bib files, but everything is an object, defined by a key and some fields. More precisely, CrossTeX extends the cross-reference mechanism of BibTeX. For example, we can define an author as shown in Fig. 10 and use it to specify Fig. 3 more concisely:

```
@BOOK{newton2011, AUTHOR = mb141a, ...}
```

The initial entry type library of CrossTeX extends BibTeX's by other object

definitions such as `@AUTHOR`. Some objects are new entry types, such as `@PATENT`. Fig. 10 also gives an example of *conditional fields*, depending on a field's value. CrossTeX can be used as a replacement of BibTeX — in which case only basic BibTeX's bibliography styles have been implemented — and it also provides a converter from .xtx files into .bib ones and into pages using HTML.

2.4 The biblatex package

Let us recall that the .bbl files suitable for the jurabib package contain texts marked up with LaTeX commands (cf. Fig. 7), that is, formatting 'References' sections is deferred to LaTeX. The biblatex package and bibliography style [33] go further with this approach, as shown in Fig. 11. In this framework, BibTeX is used only to sort bibliographical items and generate labels. A variant of our example is shown in Fig. 12 using the biblatex package. The result looks like the same example with the jurabib package:

[34] ***Hyper-Article** en **Ligne***, that is, 'hyper-article on-line'.

```
\entry{newton2011}{book}{}
\name{author}{1}{}{%
  {{}{Newton}{N.}{Mike}{M.}{}{}{}{}}%
 }\list{publisher}{1}{{Worldwide Library}}
 \strng{namehash}{NM1}
 \strng{fullhash}{NM1}
 \field{number}{141}
 \field{series}{Don Pendleton's Mack Bolan}
 \field{title}{Resurgence}
 \field{month}{04}
 \field{year}{2011}
\endentry
```

Figure 11: Reference for the biblatex package.

cf. (2). We can notice commands interfacing the fields of a reference, such as:

`\citeauthor` `\citetitle` `\citetitle*`[35]

Only one bibliography style — biblatex — is used with the biblatex package; thus, the `\bibliographystyle` command is not given. The `\bibliography` command must be included in the document's preamble,[36] and just specifies the .bib files to be searched in order to build the bibliography.[37] Inserting the 'References' section within the document is done by the `\printbibliography` command.

If you would like to make a source text generated by biblatex.bst conform to a particular bibliography style, you can redefine some LaTeX commands introduced by the biblatex package. For example:

```
\renewcommand{\mkbibnamelast}[1]{%
 \textsc{#1}}
\DeclareFieldFormat[book]{number}{\##1}
```

The former allows last names of people to be typeset using small capitals; the latter puts a '#' sign just before the contents of the `NUMBER` field for a book. You can organise the elements of a particular entry type, e.g., `\DeclareBibliographyDriver{book}{...}`.

Such technique may lead to a great number of redefinitions; so a better method is to customise the layout of a bibliography by means of the biblatex package's *options*. In fact, using this package is analogous to using Bibulus (cf. § 2.3.1), in the sense that a bibliography style is built by assembling its features as options of this package, according to the syntax '*key*=*value*'.[38] A rich library of styles is provided. For example:

```
\usepackage[style=authoryear,abbreviate=false,%
 uniquename=init,firstinits]{biblatex}
```

puts the author-date system into action, does not abbreviate keywords such as month names, assumes that author and editor names can be determined using last names only, and retains only the initial letters of authors' first names within the 'References' section. In fact, the style option is the union of two 'suboptions', e.g., '`style=authoryear`' is equivalent to:

`bibstyle=authoryear,citestyle=authoryear`

bibstyle controls the layout of 'References' sections, whereas citestyle applies to citations throughout the document. The former (resp. latter) refers to a .bbx (resp. .cbx) file. Even when the inputenc package is used, you can handle .bib files encoded differently by specifying the bibencoding option for biblatex. You can also specify keys for the sort operation, by means of mnemonics: '`sorting=nyt`' causes bibliographical items to be sorted w.r.t. authors' names, year, title. This option defaults to '`sorting=nty`'. Notice that only ASCII code order is used for sorting if .bib files are searched by means of BibTeX. An unsorted bibliography is produced by '`sorting=none`'.

Many additional fields are recognised, for example, `SUBTITLE`, for a work's subtitle, in addition to its title. You can use the standard fields `YEAR` and `MONTH`, or replace them by the `DATE` field, which allows the specification of *date ranges*:

`DATE = {2011-10-19/2011-10-21}`

The specification of fields recognised by biblatex use *types*: for example, `AUTHOR` is a *name list*, `SUBTITLE` and `TITLE` are *literals*. Some types are described by means of regular expressions, e.g., the *date* type.

```
\documentclass{article}
\usepackage{biblatex}
\bibliography{mb}

\begin{document}
\citetitle*{newton2011} is a thriller. The
Albanian Mafia is powerful, as mentioned by
\citeauthor{newton2011}.

\printbibliography
\end{document}
```

Figure 12: Example using the biblatex package.

[35] Within the biblatex package, the `\citetitle` command behaves like jurabib's `\citetitleonly` command (cf. Footnote 18), p. 292), whereas the `\citetitle*` command always puts the `TITLE` field's value, even if a short title is present.

[36] We used TeX Live 2010 for our examples. The manual of biblatex's next version [33, § 3.5.1] specifies that the `\bibliography` command has been deprecated and replaced by the `\addbibresource` command [33, § 3.5.1]. More generally, notice that biblatex's development is still in progress, so some details may be slightly out of date. The same remark is suitable about the Biber program (cf. § 2.5).

[37] In BibTeX's 'standard' bibliography styles, this command serves two purposes: specifying .bib files, and the place where the bibliography is to be typeset (cf. § 2.2). In some styles for which references are only typeset as footnotes, the `\nobibliography` command may be used instead [35, § 12.5].

[38] If a key *k* is given without a value, this is equivalent to '*k*`=true`'.

```
@BOOKINBOOK{robeson1983b,
        AUTHOR = {Kenneth Robeson},
        TITLE = {Death in Silver},
        BOOKTITLE = {Doc Savage #26-27},
        PAGES = {1-133},
        PUBLISHER = {Bantam Books},
        YEAR = 1983,
        MONTH = nov}
```

Figure 13: Nonstandard type recognised by biblatex.

Additional *entry types* can be handled, for example, @BOOKINBOOK, for items originally published as a standalone book and reprinted in collected works of an author (cf. Fig. 13). BibTeX's cross-reference mechanism has been extended [33, § 2.4.1].

Last but not least, biblatex's options encompass some features that have been implemented in separate packages. More precisely, the packages bibtopic, bibunits, chapterbib, and multibib [35, § 12.6], allowing multiple bibliographies within the same document are replaced by the environments refsection and refsegment [33, § 3.5] or by using *filters* [33, § 3.10]; similarly, the babelbib package [13], providing support for multilingual bibliographies should be replaced by the babel option [33, § 3.1.2].

2.5 The Biber program

Roughly speaking, if you use the biblatex package in conjunction with BibTeX, you go on using the latter for tasks it does not perform satisfactorily.[39] In particular, that is true about sorting, since BibTeX's sort procedures do not meet present requirements for multilinguism, as mentioned in § 1. So a new bibliography processor, Biber [5], written using Perl, has been developed. It aims to replace BibTeX for biblatex users;[40] it does not replace BibTeX wholly, since it only generates .bbl files for biblatex. The use of Biber is specified with the backend option of biblatex:[41]

```
\usepackage[backend=biber]{biblatex}
```

Such an order causes a .bcf[42] configuration file — using XML-like syntax — to be built. Let *f* be a file name without suffix, the command '`biber f`' is equivalent to '`biber f.bcf`'. An example of such a .bcf file is given in Fig. 15.

The Biber program is able to deal with the full range of the UTF-8 encoding as well as partial encodings such as Latin 1 or Latin 2. However, several encodings for several .bib files cannot be used

```
\DeclareSortingScheme{aeres}{
 \sort{presort}\sort[final]{sortkey}
 \sort[direction=descending]{
  \field{sortyear}\field{year}}
 \sort{
  \name{sortname}\name{author}\name{editor}}
 \sort[direction=descending]{
  \field{month}\literal{00}}}
```

Figure 14: Specification of an order relation for Biber.

for the same job, as in BibTeX8. biblatex and Biber are tightly coupled; some biblatex options are only available if Biber is used.

For example, '`sorting=aeres`' allows you to use the successive keys given in Fig. 14 by means of the `\DeclareSortingScheme` command, usable only if the backend is Biber, and to be put in a document's preamble: this sorting scheme partly implements the order relation used to sort the bibliography for the LIFC's activity report, as mentioned in § 1.

In fact, biblatex may use additional fields for sorting: the first pass is controlled by the PRESORT field; some fields only used for sorting — such as SORTNAME, SORTYEAR, SORTTITLE — take precedence over the corresponding fields for 'actual' information — that is, AUTHOR or EDITOR, YEAR, and TITLE. In particular, this feature is useful when these fields are marked up — in which case some alternative information without markup can be used for sorting — or when a prefix of the title has to be dropped.[43]

The construct '`\sortname[final]{...}`' overrides all subsequent `\sort` commands if the corresponding field is available. As shown in Fig. 14, the sort process stops if the SORTKEY field is available. Within the `\sort` command's argument, the three commands `\name`, `\field` and `\literal` — they correspond with the types recognised by the biblatex package (cf. § 2.4) — give the fields to be considered in turn. We also see that the `\literal` command is used as a fallback when a field is not available: here, entries without MONTH information are to be placed after all the comparable entries with this information.

A language-dependent *collation* can be used by Biber for sorting: '`sortlocale=de`'. Like BibTeX, BibTeX8, and BibTeXu, the Biber program does not perform numerical sorts, it only sorts lexicographically. Sorts are case-sensitive by default, but can be case-insensitive. As in BibTeX, additional fields are ignored silently.

3 A point of view

Let us be honest: we cannot be fully objective since

[39] ... although some points are improved with BibTeX8.

[40] The Biber program is tightly coupled with the biblatex package: if you install both, pay attention to take compatible versions. See also Footnote 36, p. 296.

[41] The other values allowed for this option are `bibtex` (by default), `bibtex8`, `bibtexu`.

[42] **B**iblatex **C**ontrol **F**ile.

[43] This *modus operandi* is not specific to Biber, it is implemented within BibTeX's biblatex bibliography style.

```
<?xml version="1.0" encoding="UTF-8"?>
<bcf:controlfile version="0.9" xmlns:bcf="https://sourceforge.net/projects/biblatex">
  <bcf:options component="biber" type="global">     <!--  Biber options   -->
    <bcf:option type="singlevalued">
      <bcf:key>bibencoding</bcf:key><bcf:value>ascii</bcf:value>
    </bcf:option>     ...
    <bcf:option type="singlevalued">
      <bcf:key>inputenc</bcf:key><bcf:value>ascii</bcf:value>
    </bcf:option>     ...
  </bcf:options>
  <bcf:sorting type="global">                       <!--  Sorting spec   -->
    <bcf:sort order="1">
      <bcf:sortitem order="1" substring_side="left" substring_width="2">presort</bcf:sortitem>
      <bcf:sortitem order="2">mm</bcf:sortitem>
    </bcf:sort>
    <bcf:sort order="2"><bcf:sortitem order="1" final="1">sortkey</bcf:sortitem></bcf:sort>
    <bcf:sort order="3">
      <bcf:sortitem order="1">sortname</bcf:sortitem><bcf:sortitem order="2">author</bcf:sortitem>
      <bcf:sortitem order="3">editor</bcf:sortitem><bcf:sortitem order="4">translator</bcf:sortitem>
      <bcf:sortitem order="5">sorttitle</bcf:sortitem><bcf:sortitem order="6">title</bcf:sortitem>
    </bcf:sort>
    <bcf:sort order="4">
      <bcf:sortitem order="1">sorttitle</bcf:sortitem><bcf:sortitem order="2">title</bcf:sortitem>
    </bcf:sort>
    <bcf:sort order="5">
      <bcf:sortitem order="1">sortyear</bcf:sortitem><bcf:sortitem order="2">year</bcf:sortitem>
    </bcf:sort>    ...
  </bcf:sorting>
</bcf:controlfile>
```

Figure 15: Example of a .bcf file.

we are MlBibTeX's author. Nevertheless, we recall that this program has been successfully used to build the publication list of our laboratory's activity report [21], and to export this publication list to an open archive site [22, 23, 24]. This list approximately contained more than 500 items. We were able to detect all the typing mistakes in BibTeX's field names.[44] Moreover, during this work, we noticed that many entries being of type `@ARTICLE` included a `PUBLISHER` field. Of course, scientific journals are often published by a professional publisher, but BibTeX's standard bibliography styles do not deal with this information, which is ignored, pure and simple. Roughly speaking, the people who specified such information in .bib files — the editors of a conference's proceedings or a journal's publisher, these two mistakes have been reported many times — probably did not know that it would never be put down in any standard bibliography style, that is, they probably would never learn that by using 'old' BibTeX. So checking all the fields of an entry may be very useful and MlBibTeX is unrivalled for this task: it has the advantages and drawbacks of a non-permissive program. By the way, adding supplementary checks is easy.[45] Also, the executable file `mlbibtex2xml` allows MlBibTeX to be used as a converter from .bib files into XML-like ones.[46] In particular, that simplifies the production of HTML pages.

Anyway, we do not deny this program's drawbacks: it is currently unable to deal with encodings other than pure ASCII and Latin 1; we plan to improve that in the next version. Besides, the interface is rudimentary: on the one hand, MlBibTeX's kernel is highly customisable,[47] on the other hand, writing some functions using Scheme in order to assemble elementary parts is often needed, apart from standard

[44] For example, '`EDITORS`' instead of '`EDITOR`', as mentioned in § 2.3.2.

[45] For example, when we exported .bib files to HAL [22], a specific format was required for the `ADDRESS` field, and additional check was added easily.

[46] Metadata for HAL are expressed using an XML dialect [22]. To convert a .bib file into something suitable for HAL, first we produce an XML file according to our internal format, then a second step — the transformation into HAL's format — is delegated to an XSLT processor.

[47] Probably more so than Biber regarding the order relations used for sorting, including language-dependent order relations. The same, writing new functions to *label* references in .bbl files is easier.

use, in which case the executable file `mlbibtex` fits well. That can be viewed as an advantage: end-users can get the full power of a programming language. In fact, programming such *drivers* for MlBibTeX is not very difficult, though we admit that this may restrict the number of potential end-users.

An objective point is that BibTeX's successors have implemented many extensions, often incompatible. These numerous extensions are the proof that this activity domain — looking for some 'better BibTeX' — is productive. Anyway, some of these projects may not pursue the same goal: for example, CrossTeX aims to reduce information redundancy as far as possible by an inheritance mechanism, whereas MlBibTeX focuses on multilingual aspects.[48] But the same notion may be implemented differently. For example, some BibTeX bibliography styles use an additional field for the total number of a book's pages. Often this field is named `TOTALPAGES` — e.g., within the jurabib bibliography style — but the tools related to biblatex know this information as `PAGETOTAL`. Likewise, some styles deal with a `DAY` field, in addition to the fields `YEAR` and `MONTH`.[49] A namesake field is internally used by the biblatex package, but the bibliography style does not recognise it as an 'actual' BibTeX field within .bib files; it is not recognised by Biber, either. These examples — the fields `DAY` and `TOTALPAGES`, there are others — show that the .bib format should be refined regarding the modern tools dealing with such files. That could lead to some convergence among such extensions and allow end-users to experiment with more bibliography processors in compatible ways.

There is a big problem with refining the .bib format: is it sufficient to add new fields, or do we have to extend the syntax of values associated with fields? Obviously, the creators of biblatex and Biber, working in tandem, have chosen not to extend values' syntax, so a kind of meta-information has been included in configuration files, possibly redefined by end-users. For example, we give an extract of Biber's default configuration file — biber.conf — in Fig. 16. These three regular expressions[50] express that prefixes — sequences of two lowercase letters before a punctuation sign, as '`al-`' in '`al-Hassan`' — and diacritics — e.g., the '`‘`' sign in '`‘Hσίοδος`' — are not considered when person names are sorted; also, the sequence '`The␣`' at the beginning of a title is to be ignored during the sort operation.

```
<nosort>
  type_name   \A\p{L}{2}\p{Pd}
  type_name   [\x{2bf}\x{2018}]
  type_title  \AThe\s+
</nosort>
```

Figure 16: Configuration of Biber (extract).

These examples are quite convincing since uncapitalised prefixes of person names are generally ruled out before sorting, the '`‘`' sign is only used in Greek[51] and does not have any influence on sorting; concerning the word '`The`', we do not know a language other than English where this word is used... but who knows, after all?[52] This information about prefixes to be dropped is handled by Biber in a language-independent way, which might be error-prone from our point of view. Anyway, here is another example: the abbreviation of first names. In French, *digraphs* should not be cut away, as shown in the first two examples:

Philippe	⟶	Ph.	*(French)*
Christian	⟶	Ch.	*(French)*
—	⟶	Chr.	*(German)*
Henry Rider Haggard	⟶	H. Rider Haggard	

The first name 'Christian' also exists in German and is abbreviated differently in this language.[53] The last example recalls that some person names retain the middle name when they are abbreviated. Even if some general information may be stored in general configuration files, it seems to us that we have to enrich .bib files' syntax in order to add such information about abbreviating first names.[54]

Finally, let us remark that MlBibTeX may be used as is with the biblatex bibliography style: an advantage is that .bib files with enriched syntax for person names can be used now with this style. Some adaptation of MlBibTeX could make it very suitable for this style — e.g., static check of the `DATE` field — and some adaptation of the bibliography style could take as much advantage as possible of MlBibTeX's

[48] However, we studied an extension of the cross-reference mechanism in order to specify translations without information redundancy [20]. This feature's implementation, planned for MlBibTeX's next version, is not finished yet; it uses a `TRANSLATOR` field, as in jurabib and biblatex.

[49] For example, the styles 'apa...' used by the American Psychology Association. See also Footnote 29, p. 294.

[50] The syntax of regular expressions used within the Perl language — Biber's implementation language — is briefly described in [45, Ch. 5].

[51] More precisely, this sign — the *rough breathing* — denoted an aspirate in ancient Greek and has disappeared in modern Greek since 1981.

[52] Accented versions of this word exist: '*thé*' for 'tea' in French, and in the Vietnamese name Hàn Thế Thành, as Karl Berry reminds me. The unaccented word might also exist in another language; who could affirm the contrary?

[53] In fact, German friends told us that this point is debatable. However, we personally saw this abbreviation in a German book.

[54] See Fig. 9 for how MlBibTeX addresses such information.

	Advantages	Drawbacks
BibTeX	Very stable; very robust; many .bib files in use.	Old-fashioned language for bibliography styles; lack of support for Unicode and multilinguism.
CrossTeX	Bibliography database files are more concise.	Lack of suitable bibliography styles.
Biber	UTF-8 supported; better backend for biblatex.	Quite slow, only usable with biblatex.
MlBibTeX	Useful check, possibly user-defined; powerful sort procedures; enriched syntax for person names.	Interface should be improved.

Table 1: Bibliography processors for LaTeX: our synthesis.

multilingual features. Likewise, an additional option '`[backend=mlbibtex]`' could take advantage of MlBibTeX's sort procedures.

4 Conclusion

The result of our synthesis is summarised in Table 1. As mentioned above, a bibliography processor usable with LaTeX should be able to deal with a huge number of extant .bib files. That is a kind of inertia: an efficient parser for .bib files is difficult to write because this syntax is old-fashioned. But recent implementations using various programming languages have shown that this difficulty can be overcome. Now the biblatex package seems to raise much interest within the LaTeX community. But this package shows that a bibliography processor more powerful than BibTeX is needed. Maybe biblatex will be the future standard for bibliographies typeset with LaTeX. Nevertheless, we think that the .bib format should be enlarged into a new format accepted by most of BibTeX's possible successors. So the extensions developed by these programs should remain compatible as far as possible. We personally plan to orient MlBibTeX's further development towards this direction.

Acknowledgements

Thanks to Barbara Beeton and Karl Berry for patiently waiting for and then proofreading this article.

◇ Jean-Michel Hufflen
LIFC — University of Franche-Comté
16, route de Gray
25030 Besançon Cedex, France
`jmhufflen (at) lifc dot univ-fcomte dot fr`
`http://lifc.univ-fcomte.fr/home/~jmhufflen`

References

[1] James C. Alexander: *Tib: A TeX Bibliographic Preprocessor*. Version 2.2. `mirror.ctan.org/biblios/tib/tibdoc.tex`. 1989.

[2] Steve R. Bourne: *The* Unix *System*. Addison-Wesley. 1983.

[3] Robert Burgess and Emil Gün Sirer: "CrossTeX: A Modern Bibliography Management Tool". *TUGboat*, Vol. 28, no. 3, pp. 342–349. In Proc. TUG 2007. 2007.

[4] Judith Butcher: *Copy-Editing. The Cambridge Handbook for Editors, Authors, Publishers.* 3rd edition. Cambridge University Press. 1992.

[5] François Charette and Philip Kime: *Biber: A Backend Bibliography Processor for biblatex. Version biber 0.9 (biblatex 1.6).* August 2011. `http://biblatex-biber.sourceforge.net`.

[6] Robert J. Chassell and Richard M. Stallman: *Texinfo. The GNU Documentation System. Version 4.13.* `http://www.gnu.org/software/texinfo`. September 2008.

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

[8] Fabien Dagnat, Ronan Keryell, Laura Barrero Sastre, Emmanuel Donin de Rosière and Nicolas Torneri: "BibTeX++: Towards Higher-Order BibTeXing". *TUGboat*, Vol. 24, no. 3, pp. 472–488. EuroTeX 2003, Brest, France. June 2003.

[9] Jean-Christophe Filliâtre and Claude Marché: *The BibTeX2HTML Home Page.* June 2006. `http://www.lri.fr/~filliatr/bibtex2html/`.

[10] Paul Graham: *ANSI Common Lisp.* Series in Artificial Intelligence. Prentice Hall, Englewood Cliffs, New Jersey. 1996.

[11] Hans Hagen: *ConTeXt, the Manual.* November 2001. `http://www.pragma-ade.com/general/manuals/cont-enp.pdf`.

[12] Hans Hagen: "The Luafication of TeX and ConTeXt". In: *Proc. BachoTeX 2008 Conference*, pp. 114–123. April 2008.

[13] Harald Harders: "Multilingual Bibliographies: Using and Extending the babelbib Package". *TUGboat*, Vol. 23, no. 3–4, pp. 344–353. 2002.

[14] Jean-Michel Hufflen: "MlBibTeX's Version 1.3". *TUGboat*, Vol. 24, no. 2, pp. 249–262. July 2003.

[15] Jean-Michel Hufflen: "BibTeX, MlBibTeX and Bibliography Styles". *Biuletyn GUST*, Vol. 23, pp. 76–80. In *BachoTeX 2006 conference.* April 2006.

[16] Jean-Michel Hufflen: "Names in BibTeX and MlBibTeX". *TUGboat*, Vol. 27, no. 2, pp. 243–253.

TUG 2006 proceedings, Marrakesh, Morocco. November 2006.

[17] Jean-Michel Hufflen: "Managing Order Relations in MlBibTEX". *TUGboat*, Vol. 29, no. 1, pp. 101–108. EuroBachoTEX 2007 proceedings. 2007.

[18] Jean-Michel Hufflen: "Revisiting Lexicographic Order Relations on Person Names". In: *Proc. BachoTEX 2008 Conference*, pp. 82–90. April 2008.

[19] Jean-Michel Hufflen: "Languages for Bibliography Styles". *TUGboat*, Vol. 2008, no. 3, pp. 401–412. TUG 2008 proceedings, Cork, Ireland. July 2008.

[20] Jean-Michel Hufflen: "Specifying Translated Works in Bibliographies". *ArsTEXnica*, Vol. 6, pp. 93–97. In GUIT 2008 meeting. October 2008.

[21] Jean-Michel Hufflen : *Classe superreport — Manuel d'utilisation.* Mars 2010. http://lifc.univ-fcomte.fr/home/~jmhufflen/superreport/superreport-readme.pdf.

[22] Jean-Michel Hufflen: "Using MlBibTEX to Populate Open Archives". In: Tomasz Przechlewski, Karl Berry, Gaby Gic-Grusza, Ewa Kolsar and Jerzy B. Ludwichowski, eds., *Typographers and Programmers: Mutual Inspirations. Proc. BachoTEX 2010 Conference*, pp. 45–48. April 2010.

[23] Jean-Michel Hufflen : *Utilisation du convertisseur .bib* $\longrightarrow$ *HAL.* Octobre 2010. http://lifc.univ-fcomte.fr/home/~jmhufflen/superreport/.

[24] Jean-Michel Hufflen: "From Bibliography Files to Open Archives: The Sequel". In: Karl Berry, Jerzy B. Ludwichowski and Tomasz Przechlewski, eds., *Proc. EuroBachoTEX 2011 Conference*, pp. 61–66. Bachotek, Poland. April 2011.

[25] Jean-Michel Hufflen: "Bibliography Tools and ConTEXt/LuaTEX". To appear in Proc. ConTEXt meeting 2011. September 2011.

[26] *Java Technology.* March 2008. http://java.sun.com.

[27] Richard Kelsey, William D. Clinger, and Jonathan A. Rees, with Harold Abelson, Norman I. Adams iv, David H. Bartley, Gary Brooks, R. Kent Dybvig, Daniel P. Friedman, Robert Halstead, Chris Hanson, Christopher T. Haynes, Eugene Edmund Kohlbecker, Jr, Donald Oxley, Kent M. Pitman, Guillermo J. Rozas, Guy Lewis Steele, Jr, Gerald Jay Sussman and Mitchell Wand: "Revised^5 Report on the Algorithmic Language Scheme". *HOSC*, Vol. 11, no. 1, pp. 7–105. August 1998.

[28] Brian W. Kernighan and Dennis M. Ritchie: *The C Programming Language.* 2nd edition. Prentice Hall. 1988.

[29] Oleg E. Kiselyov: *XML and Scheme.* September 2005. http://okmij.org/ftp/Scheme/xml.html.

[30] Donald Ervin Knuth and Silvio Levy: *The CWEB System of Structured Documentation.* Addison-Wesley, Reading, Massachusetts. 1993.

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

[32] Leslie Lamport: *LATEX: A Document Preparation System.* Addison-Wesley Publishing Company, Reading, Massachusetts. 1986.

[33] Philipp Lehmann: *The biblatex Package: Programmable Bibliographies and Citations. Version 1.6.* 29 July 2011. http://ctan.org/pkg/biblatex.

[34] Alex Martelli: *Python in a Nutshell.* 2nd edition. O'Reilly. July 2006.

[35] Frank Mittelbach and Michel Goossens, with Johannes Braams, David Carlisle, Chris A. Rowley, Christine Detig and Joachim Schrod: *The LATEX Companion.* 2nd edition. Addison-Wesley Publishing Company, Reading, Massachusetts. August 2004.

[36] Chuck Musciano and Bill Kennedy: *HTML & XHTML: The Definitive Guide.* 6th edition. O'Reilly & Associates, Inc. October 2006.

[37] Oren Patashnik: *Designing BIBTEX Styles.* February 1988. Part of the BibTEX distribution.

[38] Oren Patashnik: *BIBTEXing.* February 1988. Part of the BibTEX distribution.

[39] Oren Patashnik: "BibTEX 1.0". *TUGboat*, Vol. 15, no. 3, pp. 269–273. September 1994.

[40] Dave Pawson: *XSL-FO.* O'Reilly & Associates, Inc. August 2002.

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

[42] Brian Keith Reid: *SCRIBE Document Production System User Manual.* Technical Report, Unilogic, Ltd. 1984.

[43] The Unicode Consortium: *The Unicode Standard Version 5.0.* Addison-Wesley. November 2006.

[44] Herbert Voss: *Bibliografien mit LATEX.* Lehmans Media, Berlin. 2011.

[45] Larry Wall, Tom Christiansen and Jon Orwant: *Programming Perl.* 3rd edition. O'Reilly & Associates, Inc. July 2000.

[46] Thomas Widman: "Bibulus — a Perl/XML Replacement for BibTEX". *TUGboat*, Vol. 24, no. 3, pp. 468–471. EuroTEX 2003, Brest, France. June 2003.