

Dealing with Ancient Works in Bibliographies

Jean-Michel Hufflen

[...] *The race of men that the immortals who dwell on Olympus made first of all was of gold. They were in the time of Kronos, when he was king in heaven; and they lived like gods, with carefree heart, remote from toil and misery. Wretched old age did not affect them either, but with hands and feet ever unchanged they enjoyed themselves in feasting, beyond all ills, and they died as if overcome by sleep. All good things were theirs, and the grain-giving soil bore its fruits of its own accord in unstinted plenty, while they at their leisure harvested their fields in contentment amid abundance.* [...]

Works and Days (8th century BC),
[Hesiod, 1988, Lines 109–121].

Abstract

Bibliography processors used in conjunction with word processors have been designed in order to deal with recent works. That is, authors and publishing dates are known. Often, these informations are used to sort bibliographies, so they are supposed to be precise. This is not the case if ancient works are cited. We show how an extension of our `MLBIBTEX` bibliography processor can allow us to deal with such cases by means of *patterns*.

Keywords Citing ancient works, inexact year specification, unknown authors, patterns, translated works, cross-referencing, `BIBTEX`, extension of `MLBIBTEX`, `mlbiblatex` program.

Sommario

Le elaborazioni bibliografiche collegate con i word processor sono state pensate per gestire lavori recenti. In altre parole per questi lavori sono noti sia gli autori, sia le date di pubblicazione. Spesso queste informazioni sono usate per ordinare gli elenchi bibliografici, e perciò ci si aspetta che esse siano precise. Non è così quando ci si riferisce a lavori antichi. Qui mostriamo come una estensione al nostro programma di elaborazione bibliografica `MLBIBTEX` ci permette di gestire questi casi mediante *pattern*.

Parole chiave Citazione di lavori antichi, incertezza sull'anno di pubblicazione, autori ignoti, pattern, lavori tradotti, riferimenti incrociati, `BIBTEX`, estensioni per `MLBIBTEX`, programma `mlbiblatex`.

0 Introduction

Modern typesetting systems have been designed in order for new written documents to be developed, processed, and updated easily. Even if an ancient document's body reached a final state and does not have to be developed or updated, using such a modern processor to typeset ancient documents may be interesting, too, in order to digitalise works of the past or build new critical editions of such works. Good examples related to this approach are given by Delprat and Orevkov [2012] and also by Pignalberi et al. [2011, 2012]. Now let us consider *bibliography processors*, extracting *references* from bibliography database files. They have also been designed in order to build 'References' sections for new written documents. But often bibliographies are *sorted*, that is, bibliographical references are sorted with respect to authors' names, and different works written by the same authors are sorted chronologically within the final list¹. Such a *modus operandi* is suitable if authors and publication dates are known precisely. That is obviously true for recent works, not for ancient ones:

Ἡσίοδος. Ἔργα καὶ ἡμέραι. 8th century BC.

It can be noticed that such a reference has only historical interest, it does not point to a resource—printed or on-line—that readers can get. Most often, modern translations are used and cited, and such a translation has a publisher and a precise publication date:

Hesiod. *Theogony — Works and Days* [Θεογονία — Ἔργα καὶ ἡμέραι], translated from ancient Greek by Martin Lichtfield West (Oxford University Press, 1988).

However, several translations of the same ancient work may be cited throughout the same article, if we aim to avoid the repetition of the original title and metadata associated with all these translations, a possible solution may look like:

[1] Hesiod. *Theogony — Works and Days*. English translation of [3] by

1. There exist *unsorted* bibliographies, in which case the order of the references is the order of first citations of these items throughout the text. Unsorted bibliographies are traditional in some specific areas such as Medicine or History, but it seems to us that most bibliographies are sorted.

Martin Lichtfield West. Oxford University Press, 1988.

- [2] HÉSIODE : *Théogonie — Les travaux et les jours*. Traduction française de [3] par Paul MASON. Société d'Édition « Les Belles lettres », Paris, 1979.

- [3] Ἡσίοδος. *Θεογονία — Ἔργα καὶ ἡμέραι*. 8th century BC.

that is, by using a cross-referencing system for translations, as sketched in Hufflen [2008b]. In order for such a system to be put into action, the third item has to be specified as a separate entry, being type @MISC or @UNPUBLISHED if the BibTeX bibliography processor [Patashnik, 1988a] is used².

In this article, we do not go thoroughly into this notion of cross-references, but focus instead on the features provided by a new experimental extension of MIBibTeX³ [Hufflen, 2003], so-called **inexact**, for *imprecise* metadata, such as inexact dates of publication or surmised author's identity. As introduced before, we consider that an **inexact date** is a date whose year is not precisely known, e.g., *Oedipus the King*, a tragedy by Sophocles (495 BC–406 BC), was first performed around 429 BC. Another example is given by Hesiod's *Works and Days*, as shown before. Concerning doubtful identity of authors, the following example is related to Italian music:

[Mersenne Marin? (1588–1648)] *Campagne Parisiennes — Aria*. In *Ancient Airs and Dances, 2nd Suite*, by Ottorino Respighi (1879–1936).

Another suite arranged by Ottorino Respighi gives an example of an unknown author:

[Ignoto] *Italiana*. In *Ancient Airs and Dances, 3rd Suite*, by Ottorino Respighi.

In Section 1, we explain why such metadata are difficult to handle if ‘old’ BibTeX is used. Section 2 briefly recalls the perspectives open by BibTeX's successors. Section 3 introduces and discusses MIBibTeX's new syntactic features for ancient works' metadata. Throughout the article, we mention that MIBibTeX provides advanced features for sorting bibliographical items, described in Hufflen [2012b]. In an annex that may be viewed as a complement to this document, we show how they have been updated. Reading this article only requires basic knowledge about BibTeX and MIBibTeX.

2. In this article, we use MIBibTeX's terminology: a bibliography database file contains bibliographical *entries*, an article's ‘References’ section contains bibliographical *references*.

3. MultiLingual BibTeX.

1 Doing it with BibTeX

BibTeX was designed in the '80s and, as mentioned above, is suitable for ‘modern’ documentation: bibliographical entries refer to works that came out for a short period of time. Years are supposed to be written using the same format, in practice, either two digits⁴ or four ones. As a counter-example, let us consider two bibliographical items specified this way:

```
@...{i0, ..., YEAR = 800, ...}
@...{i1, ..., YEAR = 1492, ...}
```

the i_0 item will be ranked after the i_1 item because ‘800’ and ‘1492’ are viewed as *strings* rather than numbers; a string beginning with ‘1’ takes precedence over a string beginning with ‘8’. In fact, the sort key used by BibTeX's standard bibliography styles for an entry is a concatenation of authors' names, year, and title⁵. As a consequence, any string may be associated with an entry's YEAR field. For example, we could write:

```
YEAR = {8th century BC}
```

That would not cause any error, BibTeX would not crash, but if there are other bibliographical entries with the same author name—e.g., some modern translations—the sort's result might be strange. For the same reasons, negative years are syntactically allowed within YEAR fields⁶:

```
YEAR = {-753}
```

but BibTeX does not handle them correctly.

The situation is better about uncertain identity for authors, because we can use a command for a dummy accent:

```
AUTHOR = {Mersenne Marin{\unsure}}
```

provided that this \unsure command has been defined within L^AT_EX documents using this entry. If this command is surrounded by additional braces within values associated with AUTHOR fields, BibTeX views it as an accent command and ignores it during the sorting step. However, let us notice that the two strings ‘Mersenne Marin’ and ‘Mersenne Marin{\unsure}’ are viewed equivalent during the sort step. That is, the ‘actual’ works of this author are merged with the works that could be attributed improperly⁷.

4. When BibTeX was launched, this choice was very common: that moment was a long time before Y2K (the *Year 2000 problem* or *Millennium bug*).

5. This operation is fully explained in Hufflen [2008a].

6. Let us recall that in BibTeX, delimiters—braces or double quotes—can be omitted around natural numbers. If a ‘-’ sign is used for a negative number, the complete value must be surrounded with delimiters. Concerning years, we also recall that the ‘+’ sign is never used for a year AD (*Anno Domini*), so only years BC (*Before Christ*) are written using a sign.

7. In fact, BibTeX's sort procedure is not very discriminating [see Hufflen, 2008a for complete details]. As another

2 BibT_EX's successors

For a long time, BibT_EX was unrivalled as *the* bibliography processor associated with the L^AT_EX word processor [Mittelbach and Goossens, 2004]. However this program is ageing and difficultly meets some new requirements about bibliographies, as we explained in Hufflen [2011]. So modern programs aim to replace it. In particular, they aim to get rid of BibT_EX's language for bibliography styles [Patashnik, 1988b], old-fashioned and based on handling a stack. This replacement process is long, but more and more L^AT_EX users build their documents' bibliographies with the biblatex package [Lehman, 2014]. The MIBibT_EX program [Hufflen, 2003] is less widespread but some big-sized projects using it have succeeded.

2.1 The biblatex package

Let us recall that when the biblatex package [Lehman, 2014] is used, formatting 'References' sections is deferred to L^AT_EX⁸. So the files generated by BibT_EX in this case contain texts marked up with L^AT_EX commands customised when this package is loaded⁹. A new bibliography processor, biber [Kime and Charette, 2014], is able to build files suitable for this package, and aims to replace BibT_EX within this framework. When it works, biber is able to deal with more fields. For example, a SORTYEAR field can take precedence over the YEAR field when entries are sorted, even if only YEAR fields' contents appear within references¹⁰. For our purpose, these two fields YEAR and SORTYEAR might be used as follows:

YEAR = {154?}, SORTYEAR = 1549

that is, the SORTYEAR field allows such an entry to be ranked at the end of the entries originating from the same decade (1540's), whereas the YEAR field expresses that the unit digit is not really known. However, biber—like BibT_EX—uses only string sorts, that is, negative years are not handled correctly. Besides, no syntax for authors' uncertain identity is provided.

2.2 MIBibT_EX

Initially, MIBibT_EX [Hufflen, 2003] was designed in order to provide a 'better' BibT_EX, with particular

example, BibT_EX's standard bibliography styles do not use MONTH fields during the sorting step. As a consequence, works of the same authors and same years of publication cannot be sorted according to this month information.

8. The biblatex package has been introduced in Italian in Pantieri [2009].

9. A comparable approach exists within ConT_EXt's bib module [CONTEXTGARDEN, 2012].

10. Let us mention that the biblatex package allows a *complete* date—that is, year, month, and day—to be specified by means of a DATE field, taking precedence over BibT_EX's standard fields YEAR and MONTH. In practice, this DATE field is more and more used, but does not really apply for inexact dates.

focus on multilingual features. Later, it integrated other services, as detailed in Hufflen [2012b]. When BibT_EX's users experiment MIBibT_EX, they immediately notice that the latter is less permissive than the former. A particular case of this feature: only non-zero integers—including negative ones¹¹—are allowed as values associated with YEAR fields. This kind of *type-checking* has some advantages: for example, whenever field names are unknown, warning messages are emitted. On the contrary, BibT_EX would have silently ignored such fields if they are optional. As another advantage, MIBibT_EX can perform numerical sorts on numerical data, such as years¹². At first glance, this feature runs counter to the introduction of more expressive power. But within MIBibT_EX's source files, associating a field name with a new definition of the corresponding parsing function is easy. In fact, we developed MIBibT_EX's parser in order for syntactical extensions to be added easily.

3 MIBibT_EX's inexact extension

The features we present hereafter belong to the *inexact* extension. This extension can be used as an option of the commands `mlbibtex` and `mlbibtex2xml`¹³:

`mlbibtex [-inexact] job-name`

where '[...]' stands for an optional part and '*job-name*' refers to an auxiliary (.aux) file, as in 'old' BibT_EX. The *inexact* extension increases expressive power about inexact years and uncertain identity, while performing strict type-checking for some specialised fields such as YEAR, AUTHOR, or EDITOR.

3.1 Inexact dates

Syntactically, a value associated with the YEAR field must be either an exact year, that is, an integer beginning with a non-zero digit and possibly preceded by the '-' sign, or an inexact one. If the prefix 'ca' (for '*circa*') is used before an exact year, this prefix means 'around this date':

YEAR = {ca1492} YEAR = {ca-429}

Other inexact years may be specified in means of *patterns*, in which case '?' is for an unknown digit. Examples:

'154?' means a year between 1540 and 1549
'15??' 1500 ... 1599

An imprecise digit cannot be followed by a precise one. In other words, the question mark '?' cannot

11. But not zero, because there was no Year zero, Year 1 immediately succeeded Year -1.

12. Let us recall that on the contrary, BibT_EX and biber only run lexicographic sorts.

13. See Hufflen [2012b] about the commands provided by MIBibT_EX.

be followed a digit from 0 to 9, so ‘15?5’ is an incorrect value for a YEAR field. The zero digit is allowed at the leftmost position if it is immediately followed by a question mark, e.g., ‘0?’ is for a year between 1 AD and 9 AD, ‘-0?’ is for a year between 9 BC and 1 BC.

The rules for MIBIBTEX’s increasing sort operation are organised as follows:

- an inexact year is ranked *after* an exact year being the same value; e.g.:

1492 < ca1492 -429 < ca-429

- a pattern for an inexact year is ranked *after* the greatest year it can replace;
- inexact years are sorted according to *imprecision levels*, that is, the number of occurrences of the ‘?’ sign. For example:

1599 < ca1599 < 159? < 15??

because the imprecision levels of the last three—inexact—years are respectively 0, 1, and 2. If years before Christ are considered, applying the same rules yields:

-1500 < -ca1500 < -150? < -15??

Let us go back to syntactical rules about years and denote them by means of *regular expressions*, as allowed in programming languages like Perl¹⁴ or taxonomy description languages like XML Schema [W3C, 2008]. Exact years can be specified by the following regular expression¹⁵:

-?[1-9][0-9]*

Both exact and inexact years are described by the deterministic regular expression:

ca-?[1-9][0-9]* | -?[1-9][0-9]*\?* |
-?0\?+

3.2 Uncertain identity

An unknown author is specified by the ‘??’ sequence. Using this sequence before an author name means that this identity is uncertain.

AUTHOR = {?? and Mersenne Marin}

is for two authors, the first is unknown, the second is Mersenne Marin. The notation:

AUTHOR = {?? Mersenne Marin}

means that the author’s identity is surmised. You can also write:

14. **Practical Extraction and Report Language** [see Wall et al., 2000 for an introductory book about this language].

15. Readers unfamiliar with regular expressions can consult Stubblebine [2007] for a good introduction.

AUTHOR = {?? Marin, Mersenne}

or put this ‘??’ notation before MIBIBTEX’s keywords introducing a person name’s parts [Hufflen, 2003]:

AUTHOR = {?? first => Mersenne,
last => Marin}

It is well-known that an organisation name can be used an author’s name, provided that additional braces allow this organisation name to be viewed as the *Last* part of a name handled by BIBTEX [Patashnik, 1988b]:

AUTHOR = {{Gruppo Utilizzatori Italiani
di \TeX}}

If the identity of such a name is uncertain, do not put the ‘??’ sequence between braces—it would lose its special meaning—but proceed as follows:

AUTHOR = {?? {Gruppo Utilizzatori
Italiani di \TeX}}

or use MIBIBTEX’s **org** keyword:

AUTHOR = {?? org => Gruppo Utilizzatori
Italiani di \TeX}

The rules for increasing sort operation are:

- an unknown author is ranked *before* any known one; e.g.:

?? < Claudio Beccari

- an author with uncertain identity is ranked *after* the same author when its identity is sure; e.g.:

Mersenne Marin < ?? Mersenne Marin

4 References’ look

Let us recall that MIBIBTEX is written using the Scheme programming language [Kelsey et al., 1998] and running our parser on bibliography database files results in a structure that may be viewed as an XML¹⁶ tree. Technical details related to our XML format fall out of the scope of this paper, but we mention that this format has been enlarged in order to include markers related to inexact years and uncertain identity. Likewise, MIBIBTEX’s initial library now includes Scheme functions able to deal with such data. For example, given an inexact year, such a function allows us to know its imprecision level. As a consequence, it is easy to write a function that displays the corresponding year or century with a question mark if the imprecision level is respectively 1 or 2. Let us recall some previous examples, applying this function when a ‘References’ section is built results in:

16. **eXtensible Markup Language**.

YEAR = {154?} $\implies$ 154?
 YEAR = {15??} $\implies$ 16th century
 YEAR = {-7??} $\implies$ 8th century BC

Likewise, the full power of a programming language easily allows us to display a negative year—`YEAR = {-753}`—as such or as a year BC:

−753 753 BC

At the present time, `MLBIBTEX`'s `inexact` extension is experimental. Since `BIBTEX`'s bibliography styles cannot deal with these new features, it is impossible to use the compatibility mode allowing such 'old' styles to be applied. So only 'new' styles using the `nbst`¹⁷ language [Hufflen, 2003] could be updated. Another solution consists of bibliography styles wholly written using Scheme, the only style available now being an extension of `plain`, that is, references are sorted—as we showed above—and labelled by numbers¹⁸. The only way to customise this style is to redefine some Scheme functions¹⁹. Of course, we plan to go on with developing new styles and interfaces if users are interested. Criticisms and suggestions to do that are welcome.

5 Conclusion

On the one hand, we think that the features introduced by the `inexact` extension of `MLBIBTEX` should remain optional. As mentioned above, most bibliographies—in particular, for recent works—are composed of items with precise metadata about dates and authors, and `MLBIBTEX`'s default type-checking procedures allow us to be ensured about that. On the other hand, the results we got within our experiments of this extension are encouraging, but we only tried basic examples. We could certainly improve our 'inexact' features, but we need further experiment in collaboration with people involved in ancient studies in order to go on with this direction.

A Sorting with `MLBIBTEX`

In this annex, we do not detail all the features of the sort procedures used within `MLBIBTEX`, we just aim to show how the framework described in Hufflen [2012b] has been extended.

Let us recall that when `MLBIBTEX` processes an `.aux` files, it builds a list of citation keys, according to the order of first citations using these keys throughout the text. If the bibliography to be built is unsorted, we use this list as it is, as

17. New Bibliography `ST`yles.

18. At the present time, if you use both the `inexact` extension and a `\bibliographystyle` command within your `LATEX` document, a warning message is emitted, and this command is useless.

19. That can be done by means of Scheme expressions stored into a `.smlbibtex` file belonging to the current user's home directory.

soon as corresponding entries are associated with each member. Otherwise, this list of citation keys is *sorted*. All the sorts performed by `MLBIBTEX` are *stable*²⁰; all the lexicographic sorts are language-dependent, the default language being English. The generation of such sorts is explained in Hufflen [2007] and additional details concerning person names are given in Hufflen [2008a].

As shown in Hufflen [2012b], all the order relations used by our sort procedures have a *thunk*²¹ as an additional argument representing what is to be run when the two data are equal. This feature allows us to chain the use of several sort keys easily: if two data are viewed as equal by a sort key, the comparison based on the next sort key is immediately called. Our numerical order relations are written within this framework, so do our new comparison procedures including `inexact` dates. Lexicographic order relations have additional optional arguments controlling the sort's direction and the case comparisons [Hufflen, 2012b]. The `<authors<?` function, allowing us to compare bibliographical items regarding authors or editors has been extended:

`((<authors<? rel?) i0 i1 k0 f0 f1)`

where i_0 and i_1 are the two items to be compared, *rel?* the language-dependent order relation used to compare strings, k_0 the thunk that it called if i_0 and i_1 are equivalent. The new optional arguments f_0 and f_1 are respectively called when comparisons involve unknown authors and uncertain identity. Default values for f_0 and f_1 behave as we shown in § 3.2, but other functions can be used. In other words, processing such cases can be customised.

Often the order relation used by the sort procedure is closely related to the bibliography style used and does not appear explicitly when the bibliography processor is invoked. Practically, bibliographies are usually increasingly sorted with respect to authors' or editors' names, then dates, then titles. As shown in Hufflen [2012a,b], the `mlbiblatex` program, a variant of `MLBIBTEX` that builds bibliographies for the `biblatex` package, uses another approach: the order relation used for sorting bibliographical items is given within the command line, by means of mnemonics. In a future version, we could do the same for the order relations used to sort `inexact` metadata.

Acknowledgements

I thank Claudio Beccari for his patience when he was waiting for this article, and for his Italian translations of the abstract and keywords. Many

20. That is, the relative order of items with equal sorting keys is maintained.

21. A zero-argument function, with respect to Scheme's terminology.

thanks also to Gianluca Pignalberi who proof-read a first version.

References

- CONTEXTGARDEN. *Bibliographies in MkIV*. http://wiki.contextgarden.net/Bibliography_mkiv, July 2012.
- Bruno Delprat and Stepan Orevkov. MayaPS: Maya hyeroglyphics with L^AT_EX. *TUGboat*, 33(3):289–294, 2012.
- Hesiod. *Theogony — Works and Days* [Θεογονία — Ἔργα καὶ ἡμέραι]. Oxford University Press, 1988. Translation from ancient Greek by Martin Lichtfield West.
- Jean-Michel Huffle. MIBiB_TE_X’s version 1.3. *TUGboat*, 24(2):249–262, July 2003.
- Jean-Michel Huffle. Managing order relations in MIBiB_TE_X. *TUGboat*, 29(1):101–108, 2007. EuroBach_TE_X 2007 proceedings.
- Jean-Michel Huffle. Revisiting lexicographic order relations on person names. In *Proc. Bach_TE_X 2008 Conference*, pages 82–90, April 2008a.
- Jean-Michel Huffle. Specifying translated works in bibliographies. *ArsTeXnica*, 6:93–97, October 2008b. In Proc. GUIT 2008 meeting.
- Jean-Michel Huffle. A comparative study of methods for bibliographies. *TUGboat*, 32(3):289–301, October 2011. Proc. TUG 2011 conference, Trivandrum, India.
- Jean-Michel Huffle. MIBiB_TE_X and its new extensions. In *Proc. 6th ConT_EXt Meeting & EuroT_EX 2012*, pages 82–91, Breskens, The Netherlands, October 2012a.
- Jean-Michel Huffle. Beyond BiB_TE_X. *ArsTeXnica*, 14:7–14, October 2012b. In Proc. GUIT meeting 2012.
- 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. Revised⁵ report on the algorithmic language Scheme. *HOSC*, 11(1):7–105, August 1998.
- Philip Kime and François Charette. *biber. A Backend Bibliography Processor for biblatex. Version biber 1.9 (biblatex 2.9)*. <http://ftp.oleane.net/pub/CTAN/biblio/biber/documentation/biber.pdf>, May 2014.
- Philipp Lehman, with Philip Kime, Audrey Boruvka, and Joseph Wright. *The biblatex Package. Programmable Bibliographies and Citations. Version 2.9a*. <http://ctan.mirrorcatalogs.com/macros/latex/contrib/biblatex/doc/biblatex.pdf>, June 2014.
- Frank Mittelbach and Michel Goossens, with Johannes Braams, David Carlisle, Chris A. Rowley, Christine Detig, and Joachim Schrod. *The L^AT_EX Companion*. Addison-Wesley Publishing Company, Reading, Massachusetts, 2 edition, August 2004.
- Lorenzo Pantieri. L’arte di gestire la bibliographia con biblatex. *ArsTeXnica*, 8:48–60, October 2009.
- Oren Patashnik. *BiB_TE_Xing*. Part of the BiB_TE_X distribution, February 1988a.
- Oren Patashnik. *Designing BiB_TE_X Styles*. Part of the BiB_TE_X distribution, February 1988b.
- Gianluca Pignalberi, Salvatore Schirone, and Jerónimo Leal. Introduzione agli strumenti per grecisti classici. *ArsTeXnica*, 11:15–30, April 2011.
- Gianluca Pignalberi, Salvatore Schirone, and Jerónimo Leal. Ancora sugli strumenti per grecisti classici. *ArsTeXnica*, 13:11–16, April 2012.
- Tony Stubblebine. *Regular Expression Pocket Reference*. O’Reilly, 2 edition, July 2007.
- W3C. XML *Schema*. <http://www.w3.org/XML/Schema>, December 2008.
- Larry Wall, Tom Christiansen, and Jon Orwant. *Programming Perl*. O’Reilly & Associates, Inc., 3 edition, July 2000.

▷ Jean-Michel Huffle
FEMTO-ST (UMR CNRS 6174) & University of Franche-Comté,
16, route de Gray,
25030 Besançon CEDEX
France
jmhuffle@femto-st.fr