Programming Bibliographies

Jean-Michel Hufflen

TUG

8 August 2021

An Original View

Basic Idea $\Longleftarrow$ a text processor runs on a *computer*. That is, it is a *program*, but this program's end-users are not directly related to its programming, to the successive statements of this program. Example: interactive word processors such as Microsoft Word. They are customisable, but by means of a graphical interface (most often by interactive menus).

What about LaTeX?

Commands allow end-users to customise its behaviour. These commands are produced by a *programming language*.
Can a text processed by LaTeX take advantage of programming features?
OK if these features are (quite) easily programmable with TeX's language.
Counter-example: sort an array with LaTeX before formatting it using LaTeX. (Try to program a sort procedure using TeX's commands.)

Now

LuaTEX!
Example: perform numeric calculations before formatting data.

Building 'References' sections

(i) *Searching* bibliography databases.

(ii) *Sorting* extracted resources.

(iii) *Format* each reference.

(i), (ii), (iii) were often done by BibTeX.
Nowadays, the biblatex package is quite often used, so (iii) uses commands belonging to this package and 'actual' formatting is deferred to LaTeX's next pass. (i) & (ii) can be delegated to BibTeX, but biber is preferred.

With BibTeX

Still used... at least by some conference submission tools...
The language for bibliography styles is old-fashioned, more suitable for small changes than programming a new style.
However, this language expresses *algorithms*.
BibTeX cannot deal with *numeric* sorts!

Sorting

In BibTeX: very limited, as mentioned above.
Using biblatex: many specific fields allows special orders to be expressed, e.g.:

`sortname` `sorttitle` `sortyear`

many schemes are predefined, specified by means of *mnemonics*, but:

- as far as I know, month names are not considered,
- difficult cases are supposed to be addressed by the `\DeclareSortingTemplate` command, but practically, many complicated situations are solved by the last key `sortkey`.

Number of names

. . . for authors, editors, etc.
In BibTeX $\Longleftarrow$ limited since a string is generated as a sort key.
With biblatex $\Longleftarrow$ customisable, but not unbounded.

Reference keys

In `alpha` styles, suffixes are added in case of ambiguity. But how?

[Rob 1964a] Kenneth Robeson. *The Man of Bronze*. No. 1 in *Doc Savage Series*. Bantam Books, 1964.

[Rob 1964b] Kenneth Robeson. *The Thousand-Headed Man*. No. 2 in *Doc Savage Series*. Bantam Books, 1964.

or:

[Rob 1965] Kenneth Robeson. *The Polar Treasure*. No. 4 in *Doc Savage Series*. Bantam Books, 1965.

[Rob 1965a] Kenneth Robeson. *Brand of the Werewolf*. No. 5 in *Doc Savage Series*. Bantam Books, 1965.

More related to programming

Bibliographic module of ConTEXt, a significant part is programmed in Lua.
MlBibTEX, programmed in Scheme, Version 1.4 deals with *encodings*.

Version 1.3 is working, but the repository has been closed.
I personally escaped Covid-19, but an additional amount of work has slowed down the final part of Version 1.4 (different installation procedure).

What is (about to be) provided by MlBibTeX

An *open* format for bibliographies, which can be reached by XML tools.
A better interface with Scheme, for some important procedures, e.g., sort.
A compatibility for some additional fields of biblatex, even if some standard styles are used.
A strong type checking for some important fields.

Executable files provided

`mlbibtex` analogous to `bibtex`;

`mlbiblatex` analogous to `biber`, in the sense that generated files are suitable for the biblatex package.

I am a *programmer*

I prefer to deal with *open formats* and bibliographical references used within LaTeX should be such (possibly using a description language based on xml).
MlBibTeX is a wild animal recognising only its master...
Less true for the new version, it should recognise any master who will express orders in Scheme...

Thanks for your attention

. . . and be ready for MlBibTeX's new version!