Making BachoTEX Proceedings

Jean-Michel Hufflen

TUG 2024

20th July 2024

The origins

Fall 2020 $\Longleftarrow$ deep reorganisation within the French-speaking group GUTenberg.

The origins

Fall 2020 $\Longleftarrow$ deep reorganisation within the French-speaking group GUTenberg.
January 2021 $\Longleftarrow$ I became the new editor of the French journal *Cahiers GUTenberg*.

The origins

Fall 2020 $\Longleftarrow$ deep reorganisation within the French-speaking group GUTenberg.
January 2021 $\Longleftarrow$ I became the new editor of the French journal *Cahiers GUTenberg*.
Spring 2022 $\Longleftarrow$ ten years after the previous issue (#57), a new one (#58) is distributed to all the members of GUTenberg.

The origins

Fall 2020 $\Longleftarrow$ deep reorganisation within the French-speaking group GUTenberg.
January 2021 $\Longleftarrow$ I became the new editor of the French journal *Cahiers GUTenberg*.
Spring 2022 $\Longleftarrow$ ten years after the previous issue (#57), a new one (#58) is distributed to all the members of GUTenberg.
Tu put #58 into action, we developed some tools in order to automate the making of the whole issue, from separate articles. Some points shown at TUG 2022.

Going further

BachoTEX 2018, 2019, without proceedings.

Going further

BachoTEX 2018, 2019, without proceedings.
BachoTEX 2023 $\Longleftarrow$ I proposed to make these proceedings into a 3-part volume.

Going further

BachoTEX 2018, 2019, without proceedings.
BachoTEX 2023 $\Longleftarrow$ I proposed to make these proceedings into a 3-part volume.
Planning some extensions, e.g., two tables of contents in Polish and English.

Going further

BachoTEX 2018, 2019, without proceedings.
BachoTEX 2023 $\Longleftarrow$ I proposed to make these proceedings into a 3-part volume.
Planning some extensions, e.g., two tables of contents in Polish and English.
Expected for the summer's end.

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.
Compiling with pdfLaTeX, XeLaTeX, LuaLaTeX...

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.
Compiling with pdfLaTeX, XeLaTeX, LuaLaTeX...
BachoTeX $\Longleftarrow$ ConTeXt, too.

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.
Compiling with pdfLaTeX, XeLaTeX, LuaLaTeX...
BachoTeX $\Longleftarrow$ ConTeXt, too.
ConTeXt group? $\Longleftarrow$ doing the whole of proceedings, but only from articles written using ConTeXt.

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.
Compiling with pdfLaTeX, XeLaTeX, LuaLaTeX...
BachoTeX $\Longleftarrow$ ConTeXt, too.
ConTeXt group? $\Longleftarrow$ doing the whole of proceedings, but only from articles written using ConTeXt.
LC2 experience $\Longleftarrow$ need *automated generation* of a whole issue's parts.

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.
Compiling with pdfLaTeX, XeLaTeX, LuaLaTeX...
BachoTeX $\Longleftarrow$ ConTeXt, too.
ConTeXt group? $\Longleftarrow$ doing the whole of proceedings, but only from articles written using ConTeXt.
LC2 experience $\Longleftarrow$ need *automated generation* of a whole issue's parts.
Check programs, e.g., page number succession.

The situation

Cahiers GUTenberg $\Longleftarrow$ a kind of *legacy* with some articles ready or close to be such.
Compiling with pdfLaTeX, XeLaTeX, LuaLaTeX...
BachoTeX $\Longleftarrow$ ConTeXt, too.
ConTeXt group? $\Longleftarrow$ doing the whole of proceedings, but only from articles written using ConTeXt.
LC2 experience $\Longleftarrow$ need *automated generation* of a whole issue's parts.
Check programs, e.g., page number succession.
Avoiding *information redundancy*. For example, tables of contents should be generated automatically.

How?

Assembling PDF files? $\Longrightarrow$ pdfpages package.

How?

Assembling PDF files? $\Longrightarrow$ pdfpages package.
Recompile (LA)TEX source files? $\Longrightarrow$ Makefile.

How?

Assembling PDF files? $\Longrightarrow$ pdfpages package.
Recompile (LA)TEX source files? $\Longrightarrow$ Makefile.

```
target: dependency1 dependency2 ...
    command
```

Task related to programming

Extracting the title, the author(s), and the first page number.

Task related to programming

Extracting the title, the author(s), and the first page number.
Last page number $\Longleftarrow$ lastpage package.

Task related to programming

Extracting the title, the author(s), and the first page number.
Last page number $\Longleftarrow$ lastpage package.
Getting the other informations with (LA)TEX?

Task related to programming

Extracting the title, the author(s), and the first page number.
Last page number $\Longleftarrow$ lastpage package.
Getting the other informations with (LA)TEX? Possible theoretically, but better: using a TEX-parser written in the Scheme programming language.

Task related to programming

Extracting the title, the author(s), and the first page number.
Last page number $\Longleftarrow$ lastpage package.
Getting the other informations with (LA)TEX? Possible theoretically, but better: using a TEX-parser written in the Scheme programming language.

```
(g-retain-command "title" 2
                  #t ; The first argument is optional.
                  #t ; Top level only.
                  #t ; Recursively.
                  #t ; Within the preamble only.
                  1  ; Only one occurrence, then give up.
                  (lambda (ignored-s title-s) ...))
```

Polish/English

Title in English for the table of contents in English .

Polish/English

Title in English (resp. Polish) for the table of contents in English (resp. Polish).

Polish/English

Title in English (resp. Polish) for the table of contents in English (resp. Polish).

```
\hbsettitleinenglish{Making an Issue of \cgut}
\hbsettitleinpolish{Skład biuletynu \cgut}

\title{\hbgettitleinenglish\thanks{Title in Polish:
 \foreignlanguage{polish}{\emph{\hbgettitleinpolish}}.}
```

Polish/English

Title in English (resp. Polish) for the table of contents in English (resp. Polish).

```
%%%--smlbibtex-plus--[
\newcommand{\cgut}{%
 \foreignlanguage{french}{\emph{Cahiers GUTenberg}}}
%%%--]

\hbsettitleinenglish{Making an Issue of \cgut}
\hbsettitleinpolish{Skład biuletynu \cgut}

\title{\hbgettitleinenglish\thanks{Title in Polish:
 \foreignlanguage{polish}{\emph{\hbgettitleinpolish}}.}
```

Generating *variants*

E.g., printed or animated versions $\Longrightarrow$ additional option.

Generating *variants*

E.g., printed or animated versions $\Longrightarrow$ additional option.
Generating a new source file by means of *regular expressions* $\Longleftarrow$ Ruby.

How to proceed?

Write a *master file*.

How to proceed?

Write a *master file*.

```
\cgincludearticleplus{pdflatex}{edito-h}{edito-h}

\cgincludearticleplus{pdflatex}{knuth-eg}{knuth-eg}

\cgincludearticleplus{%
 pdflatex}{flynn-bideault}{flynn-bideault}

\cgincludearticleplus{xelatex}{poulain}{poulain-plus}

\cgincludearticleplus[%
 animated]{lualatex}{chupin}{chupin-plus}

\cgincludearticleplus{%
 xelatex}{accidentals-french}{accidentals-french}
```

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).

```
+include deps
```

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).

`+include deps`

Build the deps file:

```
poulain/poulain-plus.pdf: poulain/poulain-plus.tex
        cd poulain ; xelatex poulain-plus.tex
```

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).

`+include deps`

Build the deps file:

```
poulain/poulain-plus.pdf: poulain/poulain-plus.tex
        cd poulain ; xelatex poulain-plus.tex
```

Generate separate articles.

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).
`+include deps`
Build the deps file:

```
poulain/poulain-plus.pdf: poulain/poulain-plus.tex
        cd poulain ; xelatex poulain-plus.tex
```

Generate separate articles.
Checking the successive page number ranges and build the table(s) of contents.

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).
`+include deps`
Build the deps file:

```
poulain/poulain-plus.pdf: poulain/poulain-plus.tex
        cd poulain ; xelatex poulain-plus.tex
```

Generate separate articles.
Checking the successive page number ranges and build the table(s) of contents.
Compile the master file (you can use a suitable *target*).

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).
`+include deps`
Build the deps file:

```
poulain/poulain-plus.pdf: poulain/poulain-plus.tex
        cd poulain ; xelatex poulain-plus.tex
```

Generate separate articles.
Checking the successive page number ranges and build the table(s) of contents.
Compile the master file (you can use a suitable *target*).
On request, generate the variants.

Next stages

Skeleton file $\Longrightarrow$ Makefile (basic).
`+include deps`
Build the deps file:

```
poulain/poulain-plus.pdf: poulain/poulain-plus.tex
        cd poulain ; xelatex poulain-plus.tex
```

Generate separate articles.
Checking the successive page number ranges and build the table(s) of contents.
Compile the master file (you can use a suitable *target*).
On request, generate the variants.
Cleaning useless files $\Longleftarrow$ `make clean`.

Miscellaneous

more-abstracts package $\Longleftarrow$ *several* abstracts can be given.

Miscellaneous

more-abstracts package $\Longleftarrow$ *several* abstracts can be given.
antispam package:

`tug2024@tug.org` $\Longrightarrow$ `tug2024 (at) tug (dot) org`

How articles are changed

Putting/checking commands for titles.

How articles are changed

Putting/checking commands for titles.
Specifying the starting page number.

How articles are changed

Putting/checking commands for titles.
Specifying the starting page number.
Checking/adapting abstract(s).

How articles are changed

Putting/checking commands for titles.
Specifying the starting page number.
Checking/adapting abstract(s).
Other points related to compiling.

In other ConTEXt/contexts?

New module: `tugboat-hbachotex`, in preparation.

In other ConTEXt/contexts?

New module: `tugboat-hbachotex`, in preparation.
Using the `\author` depend on classes.

In other ConTEXt/contexts?

New module: `tugboat-hbachotex`, in preparation.
Using the `\author` depend on classes.
Dealing with possibly multilingual aspects:
babel/polyglossia/other packages *ad hoc*.

In other ConTEXt/contexts?

New module: `tugboat-hbachotex`, in preparation.
Using the `\author` depend on classes.
Dealing with possibly multilingual aspects:
babel/polyglossia/other packages *ad hoc*.
Using a way to determine the *last page number* $\Longleftarrow$ the lastpage package

In other ConTEXt/contexts?

New module: `tugboat-hbachotex`, in preparation.
Using the `\author` depend on classes.
Dealing with possibly multilingual aspects:
babel/polyglossia/other packages *ad hoc*.
Using a way to determine the *last page number* $\Longleftarrow$ the lastpage package (although some programs can compute the number of the pages of a PDF file).

Good — less good

Dependencies among files: OK, but currently only at the top level

Good — less good

Dependencies among files: OK, but currently only at the top level
(not for included files, in particular *graphic* files).

Good — less good

Dependencies among files: OK, but currently only at the top level
(not for included files, in particular *graphic* files).

Future

Before fall $\Longleftarrow$ *Cahiers GUTenberg* #59 and BachoTEX proceedings.

...

Conclusion

My system is:

open some typesetting systems can coexist, some languages, too;

Conclusion

My system is:

open some typesetting systems can coexist, some languages, too;

limited some operations require some experience in programming, some basic knowledge in Scheme;

Conclusion

My system is:

- **open** some typesetting systems can coexist, some languages, too;
- **limited** some operations require some experience in programming, some basic knowledge in Scheme;

but I am ready to:

Conclusion

My system is:

open some typesetting systems can coexist, some languages, too;

limited some operations require some experience in programming, some basic knowledge in Scheme;

but I am ready to:

- perform demonstrations for interested people;

Conclusion

My system is:

open some typesetting systems can coexist, some languages, too;

limited some operations require some experience in programming, some basic knowledge in Scheme;

but I am ready to:

- perform demonstrations for interested people;
- help end-users if need be.