WEB

An application of literate programming: creating a format for the bulletin of the Polish TUG

Włodek Bzyl
& Tomasz Przechlewski

Introduction

On establishing the Polish TEX Users Group the authors were appointed to create its bulletin. One of the first tasks was a creation of a format for the bulletin. We wanted it to be easily maintained and fully documented. The format had to be sufficient to understand, appreciate and later to modify the code by ourselves and/or others.

It was decided that the format should be coded in WEB. Knuth's WEB System of Structured Documentation was thought to fulfill our expectations. We chose FWEB, a multilingual implementation of WEB by John A. Krommes (Princeton University) based on Silvio Levy's CWEB.

WEB programs are easily modified to different environments. Ideally, we should have started with an existing WEB file and then modified it via a change file. However, there were not any formats written in WEB at that moment. To that end we had to translate to WEB the format of our choice—*TUGboat* style. Oddly enough, we became pioneers.

Problem

There are only few sets of macros for typesetting bulletins in the public domain (TTN, *TUGboat*). The *TUGboat* format is the best known, widely used one and can be obtained from almost all archives. *TUGboat* is designed for only one language—English. As our format is bilingual we had to modify *The TEXbook* style. Apart from that the modification was neccesary because:

- CM fonts had to be replaced with Polish fonts which contain Polish diacritical characters;
- some parts of the code had to be changed, some adjusted to get a design that was slightly different from *TUGboat* design;
- the parts of the code unnecessary for our purposes at that moment had to be removed.

Moreover, taking into account the technology of printing the bulletin we decided to use Computer Concrete family of fonts instead of Computer Modern. It yielded decent results.

Template of WEB program

The @ symbol in the WEB language is an 'escape character'. It introduces commands. A WEB file has the form of a sequence of two elements: comments and code. They are separated by WEB commands started with @. The skeleton of a WEB file is shown below:

```
@Lx

@* Title.   ...  Some documentation

@A
TeX macros

@* Title.   ...  more documentation

@A
more TeX macros
etc...

@* Index.
```

The parts of the code introduced by the @* sequence (or @␣, @*1, @*2) are sections (unnamed sections, subsections etc.).

@Lx	switch to TEX language
@@	@ symbol
@␣	start an unnamed section
@*	start a major section
@*1, @*2	start a subsection (subsubsection)
@A	begin TEX code
@O	open new output file

Fig. 1. List of frequently used WEB commands

Converting to WEB

The *TUGboat* format orginally comes in three files: tugboat.cmn, tugboat.sty and ltugboat.sty. All files contain a lot of comments explaining the code. tubguide.tex, which is separately delivered, is a kind of a 'user guide' for authors. It is possible using WEB to combine all the files into one.

Unfortunately, *TUGboat* format had not been converted into WEB. So the first thing to do was the conversion. The process of making a `*.web` file is quite simple. One can do this in the following way:

- Change explanatory comments to sections by removing per cent signs and preceding the whole text with `@` or `@*` symbols (short comments may be left untouched);
- precede macros with `@A`;
- double `@` in macros.

The first step is the most important one as the structure of a document is decided at that moment (sections, subsections, etc.).

What is WEB

The web file is a structured document. It consists of documentation and macros simultaneously. A web file is processed with two preprocessors: TANGLE and WEAVE. TANGLE strips off documentation and reorganizes the code. WEAVE produces documentation in TeX format.

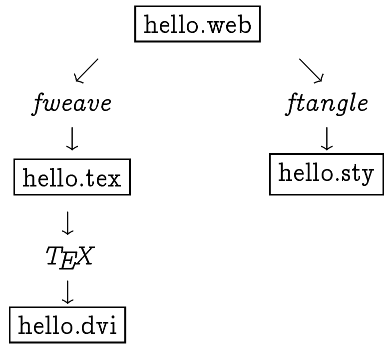

Fig. 2. WEB data flow

Modifying WEB programs

Both processors, WEAVE and TANGLE, can work with two input files: *web* file and *change* file. A change file contains data which override selected portions of *web* file. The structure of a change file has the following form:

```
@x
 ... old lines ...
@y
 ... new lines ...
@z
```

Any text up to the first `@x`, and between `@z` and `@x`, will be bypassed (some additional comments are put there usually).

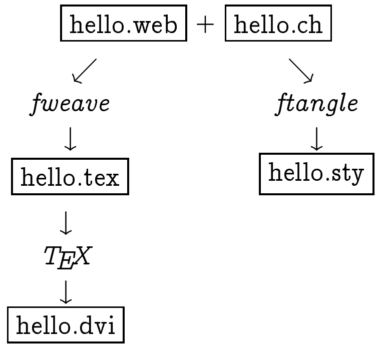

Fig. 3. WEB data flow using a change file

The whole process is illustrated with the following toy example. Let the file `hello.web` contain the following six lines:

```
@Lx

@* First example.
@A
\def\greetings{Hello!}
@* Index.
```

and the change file `hello.ch` adapts it to the Polish language.

```
change to Polish language
@x
\def\greetings{Hello!}
@y
\def\greetings{Cze\'s\'c!}
@z
```

Conclusions

We have found this approach useful in spite of the fact that we did not use all features of WEB. Named modules are supported by FWEB, but we did not use them or the conditional exclusion/inclusion or macro definitions.

FWEB is available via anonymous `ftp` from `ftp.pppl.gov:/pub/fweb`. It runs on IBM-PC's, UNIX machines, and many other systems that provide an ANSI C compiler.

Bibliography

[1] Knuth Donald E., *TeX the program*, Addison-Wesley 1988.

[2] Knuth, Donald E., *Literate Programming*, Center for the Study of Language and Information, Leland Standard Junior University, 1992.

[3] Krommes John A., *The* WEB *System of Structured Software Design and Documentation for C, C++, Fortran, Ratfor, and TEX*. User Manual, 1993.

[4] Sewell Wayne, *Weaving a Program. Literate Programming in* WEB, Van Nostrand Reinhold, NY 1989.

Appendix

Excerpt from the tugboat.web. This will generate two files: tugboat.cmn and tugboat.sty.

```
@z========================================
%%
%% Version 1.0
%%
%% W{\l}odek Bzyl, Tomek Przechlewski
%%
%%=======================================
%% original filename="tugboatc.web",
%% version="1.0",
%% date="8-July-1993",
%% filetype="TeX macros for TUGboat",
%% email="Internet:
%%     matwb@halina.univ.gda.pl,
%%     ekotp@halina.univ.gda.pl",
%% keywords="TUG, tugboat, plain tex",
%% abstract="This composite file contains
%%     the plain-based macros for preparation
%%     of TUGboat converted to WEB".
@x========================================

%% limbo

...

\def\LaMeX{{\rm L\kern-.345em
  \raise.3ex\hbox{\sc a}\kern-.16em
  M\kern-.111em\lower.6ex\hbox{E}
  \kern-.075emX}}

\def\Wtitle{GUST.WEB}

...

%-----------------------------------------

@Lx   @% set the global language to \TeX

@* Identify the version.

@A
@O tugboat.cmn
\def\fileversion{v1.0}
\def\filedate{8 July 1993}

\message{File `TUGBOAT.CMN'
    \fileversion \space\space <\filedate>}

@* Put in the index commands
   with `at' inside.

@f `\@@ 11

@* Helpful shorthand.

@*1 Changes of category.
The following allow for easier changes
of category. These require that the character
be addressed as a control-sequence:
e.g. |\makeescape\/| will
make the |/| an escape character.

@A
\def\makeescape#1{\catcode`#1=0 }
\def\makebgroup#1{\catcode`#1=1 }

...

@O tugboat.sty

...

@* Stop reading this file if
   it's been loaded already.

@A
\ifx\tugstyloaded@@\thistubstyle
  \makeatother\initializearticle
  \endinput
\else
  \let\tugstyloaded@@\thistubstyle
\fi

\message{File `TUGBOAT.STY'
  \fileversion \space\space <\filedate>}

@* Load macros common to \TeX\ and \LaTeX.

@A
\input tugboat.cmn

@* Some things with the same names
   as in, or reiterated from, \AmSTeX.

@A
@% override an \AmSTeX\ convention
\def\document{}

\output{\output@@}

...
```