Managing printed and online versions of large educational documents

Jean-Michel Hufflen

Abstract

We have developed a LaTeX 2$_\varepsilon$ package, pfa-macros, usable for both presentational education, concerning 'classical' students, and distance education, where most of a curriculum is performed by means of online documents. First, we explain why requirements for educational documents are not the same for these two ways of teaching. Then we show why our package allows us to manage two versions — printed and online — of the same textbook.

Keywords Presentational education, distance education, course text, online course, case study, LaTeX, PDF, pdfLaTeX, pfa-macros package.

1 Introduction

The Internet has revived *correspondence education*: now many network tools are widely used within this field: electronic mail, mailing lists, forums, online documents available *via* the Web, etc. The term 'correspondence education' seems to be quite old, since it appears to be related to 'classical' letters sent and delivered by post, so nowadays the term '*distance education*' is preferred. As result of greater and greater interest in distance education, most universities in the world have increased such offerings. An example of a French academic institution delivering distance education is the CTU,[1] part of the University of Franche-Comté, located at Besançon. The CTU allows students to get all the units required for a master in Computer Science. Of course, the University of Franche-Comté still provides curricula in presentational education — for students who physically attend 'classical' lectures, exercises and lab classes — which remains the 'traditional' way of teaching. Obviously some teaching units are common to the two curricula of presentational and distance education.

In this article, we show how some new LaTeX commands allow us to manage the different parts of a single document's body, for presentational students as well as distance ones. In fact, these parts have been initially written as chapters and appendices of a textbook for presentational students. Later, they have been reused and maintained as we explain in Section 2. Then Section 3 goes thoroughly into requirements about educational documents and shows that requirements for textbooks for presentational students and online documents for distance students are not the same. Our commands have been grouped into a package pfa-macros:[2] Section 4 describes the broad outlines of it. Finally, Section 5 discusses some alternative solutions.

A report about this work has already been published as [7], but within a general conference about computer-aided education, so there we reduced technical details about LaTeX's features as far as possible. The present article gives a bit more detailed description of our package's functionalities.[3] However, reading it only requires knowledge of LaTeX as an end user.

2 History

One of the teaching units we are in charge of is devoted to *functional programming*.[4] In fact, it is entitled *Advanced Functional Programming*, PFA for short,[5] since it is attended by graduate students — 4th-year university degree in computer science — that is, students who already have experience in programming. The 'philosophy' and contents of this unit are described in [6]. Let us just recall briefly that students actually practise only one programming language within this unit — Scheme [20] — but alternative implementations of functional programming concepts are exemplified using other programming languages, such as Common Lisp[6] [21], Standard ML[7] [16], CAML[8] [12], and Haskell[9] [17]. Other comparisons with modern object-oriented languages such as Java [9], C++ [22], and C# [13] are also given. In addition, we show in [6] that some examples are demonstrated using TeX's language. As a consequence, a textbook based on what is taught within this unit should include many excerpts of programs using various languages. Setting up this teaching unit PFA began in spring 1997 and the first version of our printed document [4] came out in August 1997, with a pre-version of a short additional document [5] devoted to an introduction to the λ-calculus [1], the common root of functional programming languages.

[1] ***C**entre de **T**élé-enseignement **U**niversitaire*, that is, *University Centre for Teleteaching.*

[2] Available online: `http://lifc.univ-fcomte.fr/home/~jmhufflen/latex-etc/pfa-macros.sty`.

[3] An extended version [8], more technical, is given in the proceedings of the 2010 conference of the GuIT (***G**ruppo **U**tilizzatori **I**taliani di **T**eX*), the Italian-speaking users group.

[4] *Functional programming* emphasises application of functions, whereas *imperative programming* — the paradigm implemented within more 'traditional' languages such as Pascal [25] or C [11] — emphasises changes in state.

[5] ***P**rogrammation **F**onctionnelle **A**vancée*, in French. Our package's name — pfa-macros — originates from this acronym.

[6] 'Lisp' stands for **LIS**t **P**rocessor.

[7] 'ML' stands for **M**eta**L**anguage.

[8] **C**ategorical **A**bstract **M**achine **L**anguage.

[9] Named after Haskell Brooks Curry (1900–1982).

When the master's for distance students was launched, for the academic year 2004–2005, its curriculum obviously resembled the master's in presentational education. But a unit common to these two curricula was not necessarily handled by the same teacher. Concerning us, we have been in charge of the PFA unit within both presentational and distance education, but this arrangement does not hold true for all the units. So we were interested in a method that would allow us to derive the two versions — printed and online — from the same source files. Such a *modus operandi* would ease the maintenance of our documents. For example, some slight mistakes should be fixed once, and we wished to add more examples. More ambitiously, the version of standard Scheme changed, from [2] to [10], so we ought to adapt some existing texts and examples.[10]

3 Different requirements

The document [4] consists of six chapters. Each chapter includes exercises, given with model solutions. These chapters are followed by several appendices — making precise some extra information or devoted to lab class exercises done by students — and a rich 'Bibliography' section. The whole document is approximately 400 pages long. It can be viewed as a textbook, even if its dissemination is limited to this unit's students. The students are progressively given the successive parts of this document, but it is organised as a whole, with precise architecture: cross-references are widely used throughout it. Of course, it contains not only text — in the sense of successive paragraphs — but also many examples of programs and some mathematical formulas, even if it is not really a textbook in mathematics.

3.1 Requirements about typography

When the first teaching units were launched in distance education, teachers were obviously asked to install online documents on the Web. Some teachers wrote documents using HTML.[11] However, such a choice seemed to us unsuitable for scientific documents: the look of resulting Web pages depends on the browser used; in addition, formatting mathematical formulas and program fragments often results in poor-quality output. We could have used some converters from LaTeX source texts to HTML pages,[12] which may use *images* to insert fragments whose conversion to HTML is difficult, e.g., mathematical formulas. However, even if these converters allow the output's quality to be improved, in comparison with direct writing in HTML, authors have to adapt source texts in order for the conversion to work properly. In other words, it may be difficult to do such a task for a large document already written and formatted.

Concerning the insertion of program fragments, let us recall that this point was essential, especially about the fragments given in languages other than Scheme. We could perform some demonstrations during the lab classes of presentational students, so they could observe these other programs' behaviour. The same *modus operandi* was impossible for distant students, and it was difficult to ask them to install many compilers or interpreters. So the solution was to ask them for exercises only in Scheme — as done for presentational students — but the examples given throughout our text must be explicit, in order for these students to understand without running them. In addition, we paid much attention to the indentation of these programs and inserted some comments throughout them using special effects — e.g., slanted fonts — so they do not use `verbatim`-like environments, but are built by means of `tabbing` environments.

From our point of view, only PDF[13] [3, Ch. 2] offers some sufficient warranty about the quality of texts displayed on the Web. This point is also related to *communication*: when a teacher writes some formulas onto a blackboard, students see the result exactly as the teacher formats it. The same warranty is given by PDF files, not by HTML pages. So we decided to systematically use PDF files, generated by the pdfLaTeX program [3, § 2.4]. In addition, if the hyperref package [3, § 2.3] is used, PDF files produced by pdfLaTeX can support hyperlinks, as in HTML. Let us now come to the organisational differences between texts for presentational and distance students.

3.2 Presentational vs. distance education

Of course, distance students could not be given a single document as a huge PDF file. It is preferable for distance students to get separate medium-sized files, according to the steps of their planning. Besides, let us not forget that these files are downloaded: students cannot be asked to download a huge file again if only some typing mistakes have just been fixed. Splitting this big document into separate files induces a precise organisation of cross-reference links throughout the original version. Information redundancy should be avoided, so all the parts should point to the same 'Bibliography' section, as a separate file.

[10] Later, in 2007, another change occurs, from [10] to [19].

[11] **H**yper**T**ext **M**arkup **L**anguage. A good introduction to it is [15].

[12] Some are described in [3, Ch. 3–4].

[13] **P**ortable **D**ocument **F**ormat.

Model solutions can be given after each exercise for presentational students, especially if this exercise has already been proposed in class. That cannot be the same for a document devoted to distance education: model solutions should be grouped at the end of each chapter, or provided in separate files.

4 The pfa-macros package

Let us assume that the chapters, sections, etc. of the two versions — printed and online — are numbered identically. Besides, LaTeX allows each chapter of a document to be associated with its own auxiliary (.aux) file, containing information solving cross-references. So we can compile a chapter for the online version by using the auxiliary files of the document's other chapters of the 'presentational' version. A cross-reference written by LaTeX's `\ref` command is implemented in pdfLaTeX as an internal hyperlink, which is fine for cross-references within the same chapter. For external references, we define a new command:

```
\pfaexternalref[chapter-file]{label0}
```

If the big document for presentational education is generated, this works like `\ref{label0}`. If the chapter is generated as part of the online text, a link to the file *chapter-file*.pdf is put. In both cases, the same text is displayed or enlighted by a hyperlink. If the complete version has already been put on the site, it can be searched. Otherwise, it is a kind of *stub* whose contents reads 'This chapter will be put later' and when the complete version is put, the hyperlink will remain the same.[14] We use similar technique for cross-references to footnotes belonging to another chapter (commands `\pfacite`, `\pfaexternalref`, `\pfaexternalfootnotref`, etc.).

5 Other methods

There exists some work allowing a LaTeX document to refer to a label belonging to an external document. A first example is given by some commands of the html package [3, § 3.5.3], unsuitable for us, since this package is only interesting if you want to derive HTML pages. A second implementation of external references using hyperlinks is given by the xr-hyper package [14, § 2.4.6]. Nevertheless, this package has two drawbacks for us. First, it does not deal with bibliographical citations (`\cite` commands). Second, it cannot refer to an external label that will be defined later. To explain that, let us consider that the first chapter refers to a section of the second chapter. As long as the second chapter is replaced by a stub, the hyperlink will fail; it will work only as soon as this chapter's complete text is made public.[15] Within our system, the hyperlink always points to the second chapter's PDF file, a stub or the complete text.[16]

If we had started from scratch, that is, if both the presentational and distance unit were launched at the same time, an interesting method could have been to specify our input files using XML,[17] and XSLT[18] [24] could have been used to derive texts for LaTeX, or in XSL-FO[19] [23]

6 Conclusion

A first sketch of the present article was initially designed for the EuroTeX 2010 conference. The Web page announcing this event mentioned that LaTeX is still widely used, but 'the landscape is changing', and other word processors continue to emerge.

From our point of view, the present work shows that LaTeX is still unrivalled to 'intelligently' process texts for several purposes. As mentioned above, the first version of our course text came out in 1997. Then it has evolved deeply — chapters and appendices have been wholly revised — and continuously, since we have applied some changes each year. We did it successfully — in particular when we had to be conformant with new revisions of standard Scheme — so we can think that our system is reliable.

7 Acknowledgements

I am grateful to the distance education students who addressed me very constructive criticisms; year after year, they indirectly helped me improve my tools. Thanks to Karl Berry and Barbara Beeton, who kindly proofread this article.

◇ Jean-Michel Hufflen
LIFC (EA CNRS 6942)
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`

[14] Of course, when we started this task, such a choice led us to look for all the occurrences of the `\ref` command and change some into `\pfaexternalref` ones. In practice, that was not difficult, because a good technique is to prefix labels' name by an identifier for the corresponding chapter. So the file name to be put was not difficult to supply.

[15] From a pedagogical point of view, such a *forward reference* is often viewed as bad. But it can occur within a footnote, or a fragment that can be skipped at first reading.

[16] That could be improved in a future version: if the external label exists, the hyperlink directly points to the corresponding resource, if not, it points to a stub.

[17] e**X**tensible **M**arkup **L**anguage. [18] is a good introduction to this meta-language.

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

[19] e**X**tensible **S**tylesheet **L**anguage — **F**ormatting **O**bjects.

References

[1] Alonzo Church: *The Calculi of Lambda-Conversion.* Princeton University Press. 1941.

[2] 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 Juan Rozas, Guy Lewis Steele, Jr., Gerald Jay Sussman and Mitchell Wand: "Revised Report^4 on the Algorithmic Language Scheme". *ACM Lisp Pointers*, Vol. 4, no. 3. July 1991.

[3] Michel Goossens and Sebastian Rahtz, with Eitan M. Gurari, Ross Moore and Robert S. Sutor: *The LaTeX Web Companion.* Addison-Wesley Longman, Inc., Reading, Massachusetts. May 1999.

[4] Jean-Michel Hufflen : *Programmation fonctionnelle avancée. Notes de cours et exercices.* Polycopié. Besançon. Juillet 1997.

[5] Jean-Michel Hufflen : *Introduction au λ-calcul (version révisée et étendue).* Polycopié. Besançon. Février 1998.

[6] Jean-Michel Hufflen: "Using TeX's Language within a Course about Functional Programming". *MAPS*, Vol. 39, pp. 92–98. In EuroTeX 2009 conference. August 2009.

[7] Jean-Michel Hufflen: "Recycling Previous Documents for Distance Education". In: *Proc. CSEDU 2010*, Vol. 1, pp. 469–472. Valencia, Spain. April 2010.

[8] Jean-Michel Hufflen: "Managing Printed and On-Line Versions of Large Educational Documents". *ArsTeXnica.* To appear. November 2010.

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

[10] 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.

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

[12] Xavier Leroy, Damien Doligez, Jacques Garrigue, Didier Rémy and Jérôme Vouillon: *The Objective Caml System. Release 3.12 Documentation and User's Manual.* 2010. `http://caml.inria.fr/pub/docs/manual-ocaml/index.html`.

[13] Microsoft Corporation: *Microsoft C# Specifications.* Microsoft Press. 2001.

[14] 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.

[15] Chuck Musciano and Bill Kennedy: *HTML & XHTML: The Definitive Guide.* 5th edition. O'Reilly & Associates, Inc. August 2002.

[16] Lawrence C. Paulson: *ML for the Working Programmer.* 2nd edition. Cambridge University Press. 1996.

[17] Simon Peyton Jones, ed.: *Haskell 98 Language and Libraries. The Revised Report.* Cambridge University Press. April 2003.

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

[19] Michael Sperber, William Clinger, R. Kent Dybvig, Matthew Flatt and Anton van Straaten, with Richard Kelsey, Jonathan Rees, Robert Bruce Findler and Jacob Matthews: *Revised$^{5.97}$ Report on the Algorithmic Language Scheme.* June 2007. `http://www.r6rs.org`.

[20] George Springer and Daniel P. Friedman: *Scheme and the Art of Programming.* The MIT Press, McGraw-Hill Book Company. 1989.

[21] Guy Lewis Steele, Jr., with Scott E. Fahlman, Richard P. Gabriel, David A. Moon, Daniel L. Weinreb, Daniel Gureasko Bobrow, Linda G. DeMichiel, Sonya E. Keene, Gregor Kiczales, Crispin Perdue, Kent M. Pitman, Richard Waters and Jon L White: *Common Lisp. The Language. Second Edition.* Digital Press, 1990. `http://www.cs.cmu.edu/Groups/AI/html/cltl/cltl2.html`.

[22] Bjarne Stroustrup: *The C++ Programming Language.* 2nd edition. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts. 1991.

[23] W3C: *Extensible Stylesheet Language (XSL). Version 1.1.* W3C Recommendation. Edited by Anders Berglund. December 2006. `http://www.w3.org/TR/2006/REC-xsl11-20061205/`.

[24] W3C: *XSL Transformations (XSLT). Version 2.0.* W3C Recommendation. Edited by Michael H. Kay. January 2007. `http://www.w3.org/TR/2007/WD-xslt20-20070123`.

[25] Niklaus Wirth: "The Programming Language Pascal". *Acta Informatica*, Vol. 1, no. 1, pp. 35–63. 1971.