Cahiers *GUT*enberg

Real-life Application of TEX and Adobe Acrobat for Electronic Publishing: a Handbook of Algebra and a Journal Archive

Šarūnas BURDULIS and Vytas STATULEVIČIUS

VTEX Typesetting & Electronic Publishing
Akademijos 4, 2600 Vilnius, Lithuania
`vtex@omnitel.net`
and
Institute of Mathematics and Informatics
Akademijos 4, 2600 Vilnius, Lithuania
`sarunas@ktl.mii.lt, vytas@ktl.mii.lt`

Abstract. *A classical way of using TEX in printed typesetting was enhanced for use of the same TEX source to publish electronically. A handbook of algebra and a 4-year journal archive (280 articles) were electronically published using the same TEX source files to produce both the PDF in a form for reading on-screen and a version for printing a hard copy. A package written in plain TEX provided for the mark-up of the logical structure, cross-references, bibliographical references, author names, keywords and symbols. The hypertext contents, index pages and a complete navigation system are also made in PDF and were pre-programmed at the TEX level. Being completely a PDF product the same publications are thus usable on any computer system for which a PDF viewer exists.*

1.   Introduction

"The entry of TEX into the world of hypertext" (as described by Y. Haralambous and S. Rahtz in 1995, see [1]) was probably facilitated by several factors, among which we see the rise of LATEX, the HyperTEX initiative and the emergence of Portable Document Format.

LATEX plays its role as a strong tendency to the more structure-oriented document creation as compared to the traditional TEX typesetting with the emphasis on visual perfection of the physical layout. Once there is a logical structure present in a TEX source, it may be transmitted along the processing line into a device-independent stage and further into the visualisation level. The HyperTEX specification [3] supposes an elegant set of hypertext entities to facilitate such a transmission. Tools like `dvihps` (a version of the famous `dvips` by T. Rokicki hyperised by M. Doyle to support HyperTEX's entities) and `xhdvi` (`xdvi` viewer for X-Windows with HyperTEX capability added by Arthur Smith) honestly follow the HyperTEX specification. However, as far as publication is concerned, these tools do not yet provide a vehicle capable of final widespread delivery of electronic publications to the end-user. HTML (the famous SGML DTD) and PDF (a Portable Document Format from Adobe Systems) have actually been doing widespread delivery for a couple of years. HTML (or SGML in general) and PDF are often seen as competitors in this area and this could be a starting point for another discussion; however we propose to round off here by mentioning that the two formats may successfully work together instead of competing. The `hyperref` package by S. Rahtz (described already in [1]) follows the HyperTEX specification and provides a "hyperised" DVI output from LATEX which may be used by the tools mentioned above. In addition, the package (together with the `repere` utility by Y. Haralambous in earlier versions) provides a "native"-PDF mode which generates a PDF-ready DVI output. Hypertext features are coded here by means of `pdfmark` commands, which may be used by the PDF generation utility like Acrobat Distiller. If PDF is intended as the final output, `hyperref` provides practically an "out-of-the-box" solution. PDF's strength from the TEX user's point of view is that the exact physical layout of the DVI phase is preserved. Together with the ability to carry all the font information as well as, in principle, any exotic fonts themselves, this makes PDF especially attractive for publications from the TEX source containing mathematical expressions.

We present here the making of two electronic publications for which the PDF was chosen as a preferred carrier. Both were published as a traditional hard copy too and a large part of the materials was already typeset in plain TEX when the need for electronic editions surfaced. We didn't use LATEX here, but the `hyperref` approach was partially implemented by preprogramming a specific set of PDF hypertext functions in plain TEX.

2. *Handbook of Incidence Geometry* by Francis Buekenhout in PDF

2.1. The big picture

The *Handbook of Incidence Geometry* [2] was published by Elsevier Science and is a profound work fitted into roughly 1000 pages densely packed with algebraic expressions and diagrams. Indexes by author names, keywords and mathematical symbols are included. The electronic version comprises a set of PDF files (Figure 1). Each chapter is contained in a separate file. Chapters are connected to by hypertext links from an additional file which contains hypertext contents, author index, keyword index and symbol index pages, as well as an opening page and the "Main" menu.
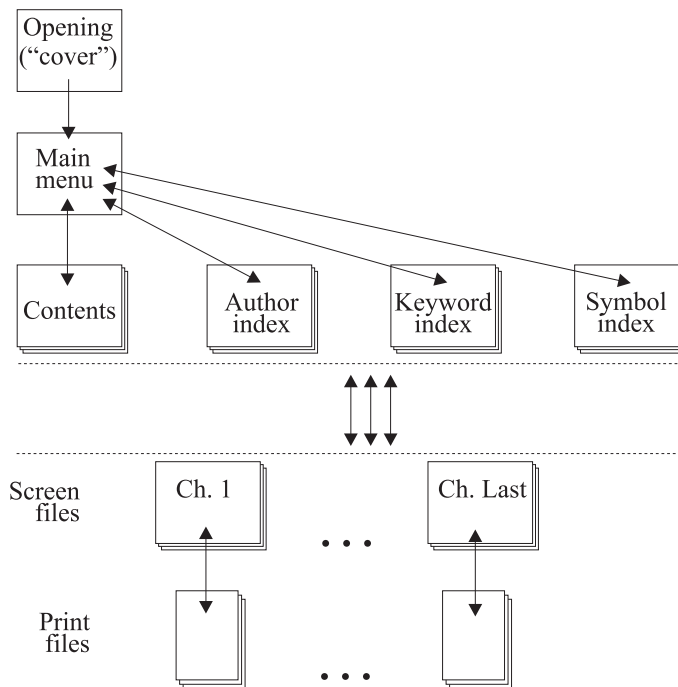

Figure 1 – Schematic layout of the hypertext connections.

First of all there is an opening page which actually is just something nice to be presented to the user when the electronic book is opened. There are two

choices there: either to open the next page or leave (exit). It is more or less a cover page. If the user "opens" it, the so-called "Main" page is displayed. The "Main" page is not much more than a list of links to:

— Preface;
— Contents by chapter titles;
— Author, Keyword and Symbol indexes.

Items listed (with the exception of Preface) are in turn lists of hyperlinks to either the start of a chapter or a particular page where the author's name, keyword or symbol is found.

All the above-mentioned pages are set for reading on screen, i.e. have a relevant aspect ratio and font sizes. In addition to the vertical navigation scheme (contents→chapters→back to contents) the horizontal links are also implemented, so it is possible to move from one index to any other index bypassing the main contents. And though the main materials ("chapters") of the book are presented in "screen" mode, there still is a possibility to print the materials in the original hard copy layout since every chapter has a satellite PDF file, which is produced in the way the book was typeset for hard copy edition. The way this is achieved, the creation of contents and index files and the build-up of the navigation system, is described in more detail in the following subsections.

2.2. Processing the main materials

The handbook was typeset using the package written in plain TEX. The request for the electronic version came later. To produce a hypertext-functional electronic edition, a style package was modified by complementing the structuring macros with functions for producing hypertext entities. Since PDF was chosen as a final form, the abstract HyperTEX notation was not used. Instead, the direct `pdfmark` code was generated and included by way of `\special`s. The `pdfmark` code was added to the following kinds of plain TEX macros:

— Section, subsection, etc. headings (down to the fourth level). Labels were derived from the heading's numbering and a corresponding named destination `pdfmark` was put there. Section heading macros also generated the PDF outline `pdfmark`s which produced the hypertext PDF outline optionally displayed on the left-hand side of the PDF viewer window.
— Originally each chapter had its own detailed contents. The macros responsible for generating such a contents now also included a `pdfmark` code for making a link from the contents to the relevant section as to the named destination.

— Bibliography entry formatting macro produced a relevant named destination in a References section.
— Macros for marking up the references from the text to the bibliography listing produced pdfmarks for hyperlinks into the References section.
— Macros for references like "see Chapter x" or "see Chapter x, Section y" produced action pdfmarks to open the corresponding file at a given section (i.e. named destination in terms of PDF).
— Macros for marking up the text to be included in indexes (author names, keywords, symbols) produced both the output data for indexes and the named destination anchors for hyperlinking back from the index files.
— A footer included macros to produce pdfmarks for the navigation bar which appears on every page. The navigation bar includes links to:

  – "Main" page;
  – chapter contents;
  – author index;
  – keyword index;
  – symbol index;
  – next/previous pages;
  – references section.

The package also provided switching for generating a different layout from the same TeX source: either a "screen" layout with active navigation or a traditional layout for printing out a hard copy corresponding exactly to the one in a printed publication.

2.3. TeX macros producing pdfmark code

In order to efficiently program tools for the above-mentioned mark-up, a set of basic pdfmark generating macros was created. The macros provided TeX commands for the generation of \specials using relevant pdfmarks:

— \PDFDesfineDest{name} defines a sf /DEST pdfmark for the named destination name.
— \PDFBookmarkToDest{text}{name} defines a /OUT pdfmark which adds text to the PDF outline with the link to name destination.
— \PDFGoToDest{text}{name} defines a pdfmark for making a link with the text to the name destination.
— \PDFGoToRDest{text}{file}{name} defines a pdfmark for making a link with the text to the name destination in another PDF file.
— \PDFGoToNext{text} and \PDFGoToPrev{text} define pdfmarks for creating links with the text to the next and previous pages in the same file respectively.

— `\PDFGoToPageNo{text}{pageno}` defines a `pdfmark` for creating a link with the `text` to the `pageno` page in the same file.
— `\PDFLaunch{text}{file}` defines a `pdfmark` for making a link with the `text` to open another `file`. This is also useful for bringing up an already open file at a current location.

The programming here may look quite straightforward. The harder part is to give the right coordinates for the clickable link rectangle. The clickable area and the underlying text are separate objects in PDF and their coordinates must be synchronised. This is achieved by putting 'currentpoint' PostScript operators into pass-through `\special`s at the corners of a T_EX box containing the link text. Retrieved coordinates then were put into the link `pdfmark` by exchanging PostScript variables. The solution works well for links denoted by the text which doesn't run across the line break. In certain cases this may be solved for multiple-line link texts by putting the argument into `\vbox` (as in the case of creating a hyperlinked chapter contents).

2.4. Hypertext indexes

Both paper and electronic editions contain author, keyword and symbol indexes common to all the chapters. Index entries are collected from the main materials by marking them up with designated T_EX mark-up macros. The macros are programmed to produce an auxiliary data file for each chapter. During the composition each marked entry is picked up and:

— a unique label is generated;
— the label is fed into `\PDFDefineDest{}` to produce a named destination;
— a line containing chapter number, section number/title, label and an index entry itself is added into the data file.

Thus after composing all the chapters and concatenating auxiliary data files a textual "database" is obtained for making the indexes. We imported the text database into MS FoxPro. A set of procedures was created to sort, filter and output the index data in a form of T_EX source files. The generated T_EX source contained index entries within the same `\PDF...{}` commands to produce hypertext links in PDF to the corresponding labels (i.e. named destinations) in appropriate chapter files.

When creating T_EX source for the index files a decision has to be made as to what is to be used as a clickable text or object for making a hypertext link. In a traditional paper version a page number (or a list of them) is given. Page numbers do not have that much meaning in the electronic version. In fact, each

chapter has its own numbering. Page numbers are also not very informative if, for example, a keyword is found in a dozen places across the entire publication. We used different linking styles for different indexes. In author and keyword indexes clickable objects are chapter numbers and section titles. In the symbol index, where a lot of entries have a very frequent appearance throughout the handbook, listing section titles would be an overkill. In this case we simply numbered the occurrences of a given index entry and a resulting series of numbers is listed as clickable tokens for each entry.

3. Archive of the *Journal of Adhesion Science and Technology* in PDF

3.1.   Journal articles

Articles for the *Journal of Adhesion Science and Technology* (published by VSP International Science Publishers, `http://www.vsppub.com`) are typeset using a plain TeX package. Originally the style used didn't have any hypertext-seeding features. Therefore with the need to publish electronically the style had to be "enriched". Since the articles are included into archive in their original layout, no effort was made to change the original page size or fonts (Type 1 fonts were already used originally). The changes introduced include:

— `pdfmark`s for filling in standard Document Info fields in PDF (Title, Author, Keywords);
— functions for generating a hypertext outline (bookmarks in PDF);
— functions for outputting data to be used in generating the archive navigation system (contents, indexes);
— colour to emphasize titles and headings.

Filling-in the Document Info was done by adding `\special`s for /DOCINFO `pdfmark`s and fetching the variables from TeX macros used to typeset the title of the article, authors' names and keywords. These macros were also enhanced to produce additional output into external data file. As a result, each time an article is composed an auxiliary data file is produced containing:

— file name (`\jobname`) for that article;
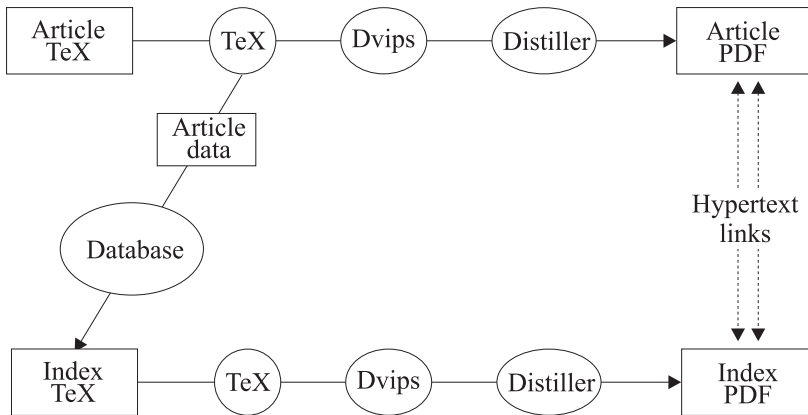— volume/issue;
— authors' names;
— keywords.

Figure 2 – Schematic workflow for generating contents/indexes.

Generating bookmarks in PDF was more involved. In order to produce a two-level deep outline a count of subsections has to be known before writing a pdfmark code for the top-level heading. For this the outline was first written into yet another auxiliary file, which was then composed on a second pass. To produce an auxiliary outline file the section and subsection formatting macros were modified to output the section/subsection title, a depth level and a page number. The main problem encountered in making a PDF outline had already been pointed out in [1]. And indeed, cleaning up the TEX output for the outline can reliably be done by hand only...

3.2. Contents and indexes

Hypertext PDF files for contents and indexes were generated from the data collected during the batch composition of articles. Figure 2 shows the schematic work flow. A plain TEX article style provided for data output into auxiliary files which were then collected into the database and processed (MS FoxPro in our case). By now articles from the journal volumes covering three years of publication are included. The chronological contents of the archive is given by year/volume/issue/article. Author names are processed into an alphabetical index fairly automatically. To make a subject index the articles had to be classified by subject—originally this information was not present in the paper edition, so a help from subject editors was needed here.

The output from the database was a huge TEX file consisting entirely out of the contents/index entries. Another TEX style was written to contain necessary macros to typeset contents/index pages for reading on screen, to create a hy-

pertext navigation bar as a footer and to provide pdfmark generation macros for index entries.

Author and keyword indexes contain a clickable alphabet on their first pages while the contents and subject indexes simply have listings of volumes/issue numbers and subject categories respectively. The footer navigation bar consists of buttons linked to the first page of a given index, next/previous pages and the main menu of the whole archive.

3.3. The final structure of the archive

The structure and the navigation scheme of the journal archive are given in Figure 3. The "Main" page or the main menu contains links to the opening pages of the volume/issue contents and indexes as well as to "Editorial board", "Aims and scope", "Instructions for authors" and Web link to the VSP home page. Except for the original article files all the pages have aspect ratio and font sizes adjusted for on-screen reading. Article files follow the original hard copy style and layout and have an individual hypertext table of contents in the form of a standard PDF outline. As of this writing, the archive includes complete journal materials from the three years of publication—280 articles, 4500 pages.
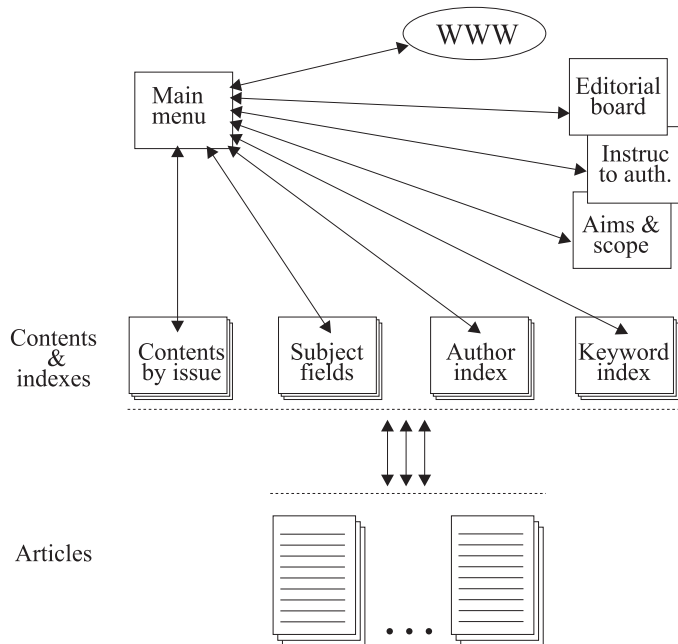
Figure 3 – Schematic layout of the hypertext connections.

4.   Remarks and conclusions

Both projects described above produce electronic publications which are completely in PDF. This means the products may be used on any system that supports a PDF viewer. Up to date, Adobe has provided Acrobat Reader free for Macintosh, Microsoft Windows and a number of UNIX systems. Starting from version 3.0, Acrobat Reader has a search engine which may search through the full-text indexes prepared by the Acrobat Catalog software which comes with the standard Acrobat distribution. Acrobat indexes were built for both publications. It is quite impressive to use such an index to search through the journal archive. On a mid-range Pentium PC the search process takes no more than a couple of seconds for 4500 pages. The query result is a list of article titles with the confidence indicators. The search terms are highlighted when the article file is subsequently open.

Both the handbook and the journal archive are delivered to the end user on CD-ROMs. Beside the publication files themselves and Acrobat indexes, the CD-ROMs also include the installers for all the Acrobat Reader versions provided by Adobe.

We conclude that practical electronic publishing may be successfully done with T_EX as a source for publication materials and a base for a hypertext system. By means of T_EX programming it is possible to publish either for on-screen reading or for printing or both from the same T_EX source. Using PDF as a final carrier format makes the publications widely usable on a number of operating systems without the need to make any special compilation or adjustment.

References

Bibliography

[1] Y. Haralambous, S. Rahtz. L^AT_EX, HTML and PDF, or the entry of T_EX into the world of hypertext. *Proceedings of the Ninth European T_EX Conference*, September 4–8, 1995, Arnhem, The Netherlands, pp. 223–238.

[2] *Handbook of Incidence Geometry*, edited by F. Buekenhout, Elsevier Science, Amsterdam, 1995.

[3] *HyperT_EX FAQ* at `http://xxx.lanl.gov/hypertex`