DML 2010

Michal Růžička; Petr Sojka

Data Enhancements in a Digital Mathematical Library

In: Petr Sojka (ed.): Towards a Digital Mathematics Library. Paris, France, July 7-8th, 2010. Masaryk University Press, Brno, Czech Republic, 2010. pp. 69--76.

Persistent URL: `http://dml.cz/dmlcz/702575`

Data Enhancements in a Digital Mathematical Library

Michal Růžička and Petr Sojka

Masaryk University, Faculty of Informatics,
Botanická 68a, 602 00 Brno, Czech Republic,
mruzicka@mail.muni.cz, sojka@fi.muni.cz

Abstract. The quality of digital mathematical library depends on the formats and quality of data it offers. We show several enhancements of (meta)data of the Czech Digital Mathematics Library DML-CZ. We discuss possible minimalist modification of regular LaTeX documents that would simplify generating basic metadata that describes the article in an XML/MathML format. We also show a proof of concept of a method that enables us to include LaTeX source code of mathematical expressions into pdfTeX-generated PDFs in such a way that the reader can Copy & Paste the code from his PDF viewer. This code, hidden in the PDF file, can also be used for LaTeX math indexing.

Key words: metadata generation, XML, MathML, PDF, copy-math

1 Introduction

Since 2005, a digital mathematics library has been under development in the Czech Republic. The goal of the Czech Digital Mathematics Library project (DML-CZ) [3] is the preservation in digital form of the contents of the major part of mathematical literature ever published in the Czech lands, and to provide free and public access to the digital content and bibliographical data.

The DML-CZ development was officially completed at the end of the 2009. The aim of this article is to give a short summary of some of the techniques that facilitated the success of this project.

A LaTeX document workflow consists of several steps, some of them can be reworked to enhance the final versions of documents that are stored in a digital repository. Besides postprocessing final PDF files [5], we can modify the processing of the document that a journal editor typically does, (can be seen in Section 2) and enrich the document source code itself (Section 3).

In this article we intend to show how a slight modification to regular LaTeX documents and classes enabled us to prepare DML-CZ metadata with only slight modification to the current workflow of the editors of mathematical journals involved.

The EuDML project [4] has already been launched and it is hoped that the DML-CZ results can be applied to it. Despite being officially finished, the result of—the Czech Digital Mathematics Library—project is here and

Petr Sojka (editor): DML 2010, Towards a Digital Mathematics Library, pp. 69–76.
ISBN 978-80-210-5242-0

we intend to continue developing in further. One possible contribution could be our method of including LaTeX source code of mathematical expressions into pdfTeX-generated PDFs in such a way that the reader can Copy & Paste it directly from his PDF viewer. A PDF file of this kind could also be used for LaTeX math indexing.

Proof of concept of this technique is shown in the second part of this article.

2 Minimalist XML Metadata Extraction

Although the greater part of the DML-CZ project was retro-digitization—which involved scanning, OCR and finally processing the paper-only documents for the digital format—, future developments of the library depend on how the new issues of the mathematical journals are processed. With this in mind, it has been necessary to prepare appropriate software support for the mathematical journals involved that will enable editors to prepare DML-CZ data easily.

The first approach was a complex system inspired by the French CEDRAM project [8,2]. It automates many of the standard procedures of the journal issue preparation [10]. Although the French system is used by the editors at the Archivum Mathematicum [1], not everyone there was willing to adopt such a complex system which seriously disrupted their current workflow.

We therefore prepared a minimalist set of LaTeX macros in the form of a LaTeX macro package. This package can be easily customized to meet needs of a particular journal document class / style file. The LaTeX macro package itself does not transform the LaTeX source code to XML. Rather, this package literally exports selected parts of the LaTeX document to an external file in such a way that it forms a simple LaTeX document. This occurs without any expansion of the LaTeX code; TeX toks registers are used (using the standard LaTeX output system—`\newwrite`, `\openout`, `\write`, `\closeout`). This file is subsequently processed by a journal-independent Tralics-based procedure, which is described in the next section.

The Tralics program [9,7]—a LaTeX to XML translator—has proved itself an adequate translator of the LaTeX code to XML.[1] Use of Tralics is the most indispensable part of the system. Its engine is able to process regular LaTeX code which obviates converting the LaTeX code to plain text directly; nor do we have to deal with the LaTeX macro expansion or the complexity of its syntax. Tralics outputs a UTF-8 encoded XML file.

This output is finally processed by the XLST processor furnishing DML-CZ metadata in its final form. A schema of the process can be seen in Figure 1.

At the same time as the final PDF document is created, the metadata is automatically generated based on the same source code. Thus, we can be sure the metadata is correct and up-to-date unlike the situation in which the editors prepare metadata 'by hand' or generate it asynchronously.

Even if the editor used another incarnation of TeX, instead of LaTeX it should still be possible to export the necessary data in such a way that the result

[1] Tralics is also used in the complex system of the Archivum Mathematicum journal.

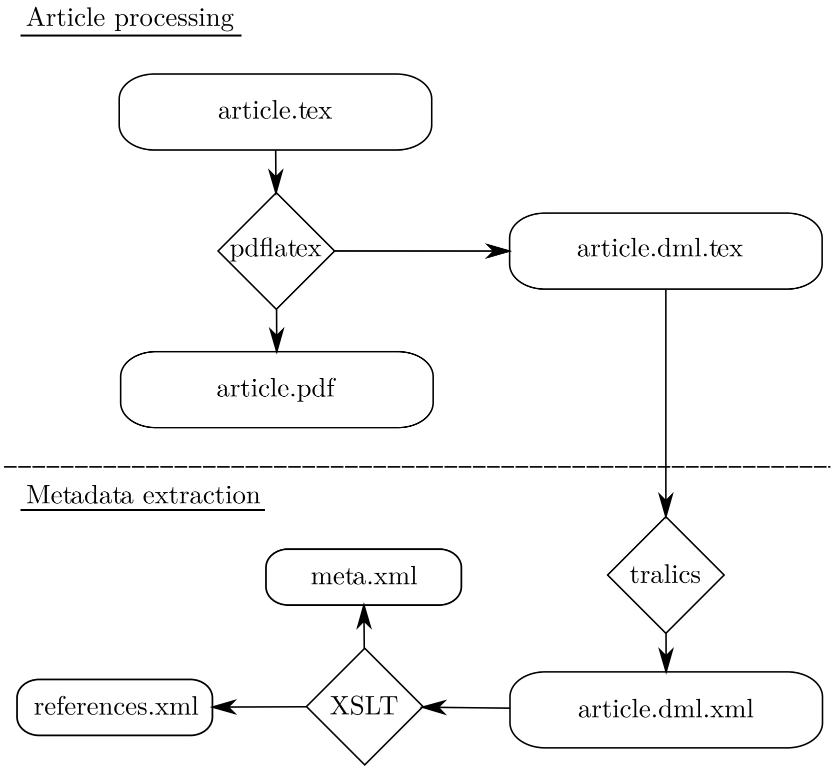


Fig. 1. Schema of work of the minimalist metadata extraction system

would be acceptable as Tralics input. (This data is a small subset of the whole document and includes just the title, author names, abstract, keywords etc.)

This solution is also platform independent—both the TEX and Tralics are multi-platform programs.[2] The XSLT processors are also available for all standard operating systems.

The minimalist extension of the current editorial workflow does not use BibTEX for references processing since not all the editors are willing to ues it. This is despite the fact that it is supported directly by the Tralics program. A special set of macros is used instead to mark up the structure of each bibliography record giving them more flexibility in bibliography formatting.

Since Tralics supports MathML thus we are able to translate mathematical expressions from the input LATEX notation to this XML language. Because we decided to support just a controlled subset of the 'well known' LATEX macros in the DML-CZ metadata, it is easier to achieve a correct MathML translation.

[2] The Tralics program uses dynamic libraries of the Cygwin project (`http://www.cygwin.com/`) to run on the MS Windows operating system.

3 Copy Math—a Proof of Concept

The DML-CZ project stores full texts of the articles as PDF files as do many other digital libraries. PDF is widely adopted and very often used for electronic publications. Thanks to PdfTEX, PDF is also the *de facto* standard output format of the modern TEX distributions.

Being capable of high quality mathematical typesetting, TEX is widely used. LATEX mathematical notation is well known, effective, and used not only in LATEX documents, but also in a variety of other projects, such as Wikipedia.

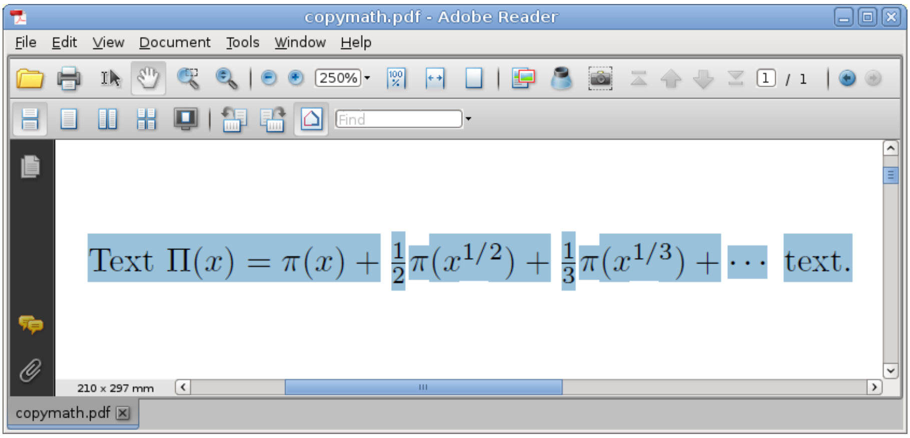
Thus, LATEX source code is usually a good choice for plain text representation of mathematical expressions. Users and maintainers of repositories of digital documents themselves demand plain text for the content of PDF documents—in Japan, regular PDF documents are processed using OCR (optical character recognition) techniques to obtain plain text representation of math from PDFs [6,11].

Unfortunately, PdfTEX-produced PDF documents do not provide their readers with this kind of output if they use Copy & Paste functions of their preferred PDF reader.

A LATEX document with a following body part has the PdfTEX generated PDF as shown in Figure 2.

```
\begin{document}
Text
$\Pi(x) = \pi(x) + \frac{1}{2}\pi(x^{1/2}) +
 \frac{1}{3}\pi(x^{1/3}) + \cdots$
text.
\end{document}
```

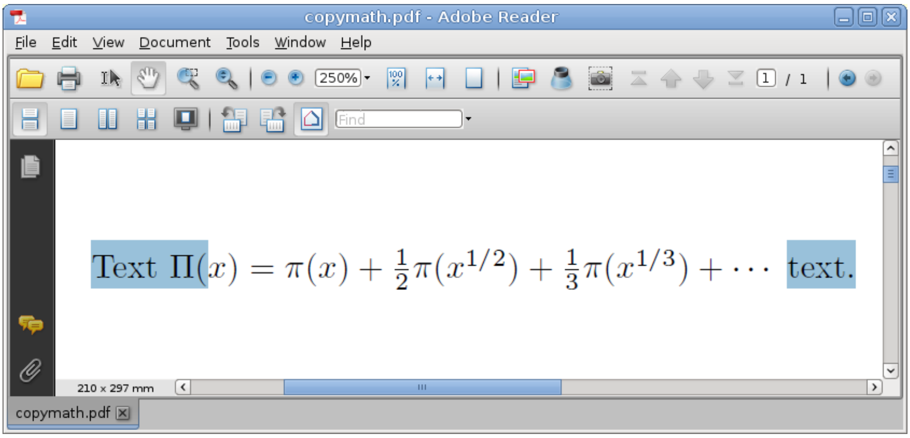
The content of the document is selected properly but the result of the Copy operation is malformed mixture of unicode characters. To address this



copymath.pdf - Adobe Reader
File Edit View Document Tools Window Help
250%
1 / 1
Find
210 x 297 mm
copymath.pdf

Text $\Pi(x) = \pi(x) + \frac{1}{2}\pi(x^{1/2}) + \frac{1}{3}\pi(x^{1/3}) + \cdots$ text.

Fig. 2. CopyMath disabled PDF document


copymath.pdf - Adobe Reader
File Edit View Document Tools Window Help
250%
1 / 1
Find
210 x 297 mm
copymath.pdf

Text $\Pi(x) = \pi(x) + \frac{1}{2}\pi(x^{1/2}) + \frac{1}{3}\pi(x^{1/3}) + \cdots$ text.

Fig. 3. CopyMath enabled PDF document

inconvenience we decided to use the `ActualText` command of the PDF language to mark the region of the mathematical expression inside the PDF document and allow PDF readers to provide their users with the LaTeX source code of the expression. Figure 3 shows the PDF file that resulted from the same document with our experimental CopyMath macro package switched on.

Mathematical expressions are not selected visually; the result of the Copy operation is the original LaTeX source code itself:

```
Text $\Pi (x) = \pi (x) + \frac {1}{2}\pi (x^{1/2}) +
\frac {1}{3}\pi (x^{1/3}) + \cdots $ text.
```

The implementation is not easy because we want the package to be as user friendly as possible—users should not be forced to modify their mathematical expressions in any way, `\usepackage{copymath}` should cater for all their needs. However, this requires nonstandard modifications of the LaTeX mathematical environments.

To implement CopyMath we need to add `\pdfliteral` at the beginning and end of every mathematical environment. The dollar sign ($) is activated (`\catcode‘$=13`) and redefined. It is necessary to keep track of nested mathematical environments (e.g. `$a\mbox{$b$}c$`), and double-dollar display-math syntax (`$$a + b$$`) adds another layer of complication.

To redefine LaTeX mathematical environments (`\begin{math}...\end{math}`, `\begin{eqnarray}...\end{eqnarray}` etc.) we keep the original definition of their opening (`\let\normalequation\equation`) and closing commands. The environment is consequently redefined using our auxiliary macros. The opening command is substituted for a macro that scans tokens until the closing command of the mathematical environment is achieved. And we must never lose sight of nested environments. The scanned content of the mathematical environment is used to prepare a `\pdfliteral` code. The `\pdfliteral` code and the original content of the mathematical environment are used by another auxiliary macro that is used instead of the closing command of the original mathematical environment.

Here is an example of CopyMath macro definitions:

```
%% Auxiliary macros.
\newcounter{nestedmath} \setcounter{nestedmath}{0}
%
\newtoks\copymath@envgetbuffera
\newtoks\copymath@envgetbufferb
%
\long\def\copymath@envget#1#2\end #3{%
    \copymath@envgetbuffera=\expandafter{\copymathenvput}%
    \def\copymath@envtempa{#3}\def\copymath@envtempb{#1}%
            \ifx\copymath@envtempa\copymath@envtempb%
        \copymath@envgetbufferb={#2}%
        \def\copymath@envgetnext{\end{#1}}%
    \else%
        \copymath@envgetbufferb={#2\end{#3}}%
        \def\copymath@envgetnext{\copymath@envget{#1}}%
    \fi%
    \global\edef\copymathenvput{%
        \the\copymath@envgetbuffera \the\copymath@envgetbufferb}%
    \copymath@envgetnext}
%
\long\def\copymathenvget#1{%
    \gdef\copymathenvput{}\copymath@envget{#1}}
%

%% $
\let\@origensuredmath=\@ensuredmath
%
\def\normalinlinemath#1{%
 \ifnum\value{nestedmath}>0 \@origensuredmath{#1}%
 \else%
    \addtocounter{nestedmath}{1}%
    \pdfliteral{/Span << /ActualText<\pdfescapehex{\detokenize{$#1$}%
                }> >> BDC}%
    $#1$%
    \addtocounter{nestedmath}{-1}%
    \pdfliteral{EMC}%
 \fi}
%
\let\@ensuredmath\normalinlinemath
%
\catcode‘$=13

%% \begin{equation}...\end{equation}
\let\normalequation\equation
\let\normalendequation\endequation
```

```
\renewenvironment{equation}%
    {\copymathenvget{equation}}%
    {\ifnum\value{nestedmath}>0 \message{You cannot nest equation}%
     \else%
       \normalequation%
            \addtocounter{nestedmath}{1}%
            \pdfliteral{/Spanx << /ActualText<\pdfescapehex{%
                \detokenize{\begin{equation}}\copymathenvput\detokenize{%
                \end{equation}}}> >> BDC}%
            \copymathenvput%
            \addtocounter{nestedmath}{-1}%
            \pdfliteral{EMC}%
        \normalendequation%
     \fi}
```

Unfortunately, it seems that this approach is not as universal as expected. For example, it is not possible to directly use this kind of macro redefinition for $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-LaTeX mathematical environments and this has necessitated a complex macro redefinition. Another possible solution should be preprocessing of the source code using an external tool. This approach, however, would need to deal with the complexity of the LaTeX syntax.

4 Conclusions

Minimalist modifications of the current editorial workflow proved to be an easy way of moving mathematical journal editors to a digital-library-friendly state. Tralics provides us with sufficient functionality to perform this easily and with platform independence.

The CopyMath macro package shows an alternative route to improving pdfTeX-generated PDFs, but the proper redefinition of all possible mathematical environments cannot be expected to be easy.

References

1. Archivum Mathematicum. [online], http://www.emis.de/journals/AM/, Masaryk University, Brno, Czech Republic. Last modified December 18, 2009. [cit. 2010-04-25].
2. Centre de diffusion de revues académiques mathématiques. [online], http://www.cedram.org/, [Center for diffusion of mathematic journals]. [cit. 2008-05-25].
3. Czech Digital Mathematics Library. [online], http://dml.cz/, [cit. 2010-04-24].
4. EuDML: The European Digital Mathematics Library. [online], http://www.eudml.eu/, This page was last modified on 20 January 2010, at 08:09. [cit. 2010-04-25].
5. Hatlapatka, R., Sojka, P.: PDF Enhancements Tools for a Digital Library. In: Sojka, P. (ed.) Proceedings of DML 2010, pp. 69–76. Masaryk University Press, Paris, France (Jul 2010).
6. Infty Project: Research Project on Mathematical Information Processing. [online], http://www.inftyproject.org/en/, [cit. 2010-06-02].

7. Tralics: a LaTeX to XML translator. [online], `http://www-sop.inria.fr/apics/tralics/`, Last modified $Date: 2009/11/24 17:17:03 $ [cit. 2010-04-24].
8. Bouche, T.: A PdfLaTeX-based automated journal production system. TUGboat 27(1), 45–50 (2006), In Proceedings of EuroTeX 2006.
9. Grimm, J.: Tralics, a LaTeX to XML Translator. TUGboat 24(3), 377–388 (2003), In Proceedings of EuroTeX.
10. Růžička, M.: Automated Processing of TeX-Typeset Articles for a Digital Library. In: Sojka, P. (ed.) DML 2008 – Towards Digital Mathematics Library. pp. 167–176 (2008), Birmingham, UK, July 27th, 2008.
11. Suzuki, M., Kanahori, T., Ohtake, N., Yamaguchi, K.: An Integrated OCR Software for mathematical Documents and Its Output with Accessibility. In: Computers Helping people with Special Needs. Lecture Notes in Computer Sciences, vol. 3119, pp. 648–655. Springer (2004), 9th International Conference ICCHP 2004, Paris, July 2004.