Thierry Bouche
CEDRICS: When CEDRAM Meets Tralics

In: Petr Sojka (ed.): Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008. Masaryk University, Brno, 2008. pp. 153--165.

Persistent URL: http://dml.cz/dmlcz/702544

CEDRICS: When CEDRAM Meets Tralics

Thierry Bouche

Université de Grenoble I & CNRS
Institut Fourier (UMR 5582) & Cellule Mathdoc (UMS 5638)
BP 74, 38402 St-Martin-d'Hères Cedex, France
E-mail: `thierry.bouche@ujf-grenoble.fr`
URL: `http://www-fourier.ujf-grenoble.fr/~bouche/`

Abstract. We describe CEDRICS, a general purpose system for automated journal production entirely based on a LaTeX input format. We show how the very basic ideas that initiated the whole effort turned into an efficient system because of the ability of LaTeX markup to parametrise simultaneously and without compromise high typographical quality for the PDF output as well as accurate XML metadata with (presentation) MathML formulas. This was made possible by the availability of two entirely independent LaTeX source processors with specific focus but full TeX-macro language support: PdfLaTeX by Hàn Thế Thành, and Tralics by José Grimm.

Key words: Tralics, CEDRICS, CEDRAM, metadata generation and conversion, LaTeX, MathML

1 CEDRAM

1.1 The project

The CEDRAM project at Cellule MathDoc in Grenoble, France, is an innovative support of academic mathematical serials joining forces by presenting their electronic editions on a unique portal in order to gain more visibility, and a professional production environment [1]. It had been discussed since 2003, formally launched in 2005, with first visible results in January 2006 when 3 journals got a CEDRAM website. A steering committee monitors the management of the project, and helps select scientifically robust serials with a rigorous editorial policy and long term development objectives. Of course, as the mathematical section of CNRS was one of the founding partners in the venture, journals positively evaluated and supported by CNRS are at the heart of the project.

The support is provided through

- a common portal (located at `http://www.cedram.org/`);
- an electronic publication platform with dedicated instances for each hosted serial (which get a website in the form `http://journal.cedram.org/`);

Petr Sojka (editor): DML 2008, Towards Digital Mathematics Library, pp. 153–165.
© Masaryk University, 2008 ISBN 978-80-210-4658-0

- a set of modular productivity tools that can be inserted at some crucial steps in each partner's production workflow, in order to get a uniform quality for the output without imposing a uniform interface for all the partners;
- an import tool from NUMDAM in order to present the whole run of journals at their CEDRAM website;
- an export to NUMDAM in order to secure long term access;
- archiving production source files for long term preservation of the mathematical content.

A broad description of the first CEDRAM phase, covering its first two years of existence, is to be found in our recently published chapter [2], as well as some material archived at the project's website. A paper focusing on the TEXnical innovations from that initial period was also published in *TUGboat* [3].

1.2 The results

At the time of writing, three main software modules have been developed or adapted to the project, where they have been routinely used since early 2007:

1. RUCHE, a software for managing the editorial process: from paper submission and refereeing through a web-based interface, to the preparation of published volumes [4].
2. CEDRICS, a LATEX driven journal issue production environment that automates whatever can be (page numbers, metadata generation...). It starts from LATEX sources and outputs the print PDF for the whole issue and its cover, the screen PDF for each article, XML metadata prepared in a dual format (text is in plain Unicode with TEI-like structure, mathematical expressions are encoded both as presentation MathML and as XMLised LATEX).
3. EDBM, a database manager taking care of metadata ingesting, indexing, interlinking, and generating the user interface to the collections (essentially the same system as in NUMDAM, which has been made more customisable).

In this report, we will focus on step 2, which has been developed during the second semester of 2006, and has allowed us to launch the second phase of the CEDRAM production workflow on March, 1st 2007.

1.3 The scope

Currently, 5 journals and 3 seminar proceedings are hosted on the CEDRAM platform. The seminars and two journals are open access, one journal has a 2 years moving wall while the remaining ones opted for a 5 years moving wall.

Let us mention that 2 more journals are produced with CEDRICS, but exploited separately: *Cahiers GUTenberg* by Association GUTenberg, and *Archivum Mathematicum* by DML-CZ team, see [5].

More surprisingly, instances of CEDRICS have proven very useful as an effective way to produce NUMDAM compatible metadata for born-digital content with incompatible native metadata. Some 1,500 articles from various sources have been processed to date, to support acquisition of post-digitisation content in NUMDAM.

2 CEDRAM, Phase II

At the end of its first year of existence, the CEDRAM was a rather fragile project. On the scientific side because some partners who were instrumental in the definition of the project eventually did not join it. On the technical side because the production workflow which had been set up in a hurry was discouragingly inefficient, so much so that an apparent success like the proposal from 2 more journals to join could have killed it, given the resources needed to customise the system to deal with a new serial, and to produce each new volume.

The CEDRAM LaTeX+BibTex system [3] that had been conceived and implemented during summer 2005 introduced interesting new methods where the entire publication process is controlled from a hierarchical directory of LaTeX source files. The two recent features that helped implement this into reality were the ability of the PdfTeX programme to launch external commands and to include multipage PDFs, so that one could compile a whole journal issue in one run, recompiling each constituent article and including the resulting PDF at once. The idea supporting this operational schema is that the TeX engine itself is the only software that can make sound predictions (i.e. generate metadata) from a TeX encoded source file (if you don't believe me, have a look at [6]). Using emulation (like latex2html, e.g.) and heuristics is error prone: it is much safer to have TeX write down all relevant information to an auxiliary file after it has been fully interpreted. Having now processed more than two thousand articles, we can report that it just works.

However, the first version of the system had two drawbacks that made it somewhat impractical for our task:

1. In order to get NUMDAM-like features, and compatibility with NUMDAM metadata schema, we imposed to store any bibliographical information in BibTex files. Although it is a neat way to insure that a journal-wide bibliographic style is obeyed by all published articles, it is not something that can afford very small publications run by academics themselves. Moreover, authors or editors are not always able to build correct fielding of the bibliographies, which often results in poor (incorrect) data. Even pulling BibTex data from reviewing databases is not a working fallback, as they reflect much more faithfully the databases they are coming from than the articles references they are supposed to identify. Also, when you try to input articles that have been already printed, you face constructs that simply cannot be properly stored in a BibTex database (merged references are a frequent example).

2. Because of the subtleties of TEX's macro expansion, it is not an easy task to use TEX itself to write textual metadata to an auxiliary file in a form suitable for generating a valid XML file. Most of CEDRAM's metadata magics rely on storing some valid TEX code in a macro, that will be executed in different contexts with local definitions. If a typeset title is going to end up in an XML element, many issues have to be cared of like on-the-fly reencoding or escaping special characters. In fact, what is really wanted is a general converter from some TEX dialect to a clean XML structure, which is not the kind of software you'd want to write using TEX macros!

The CEDRAM environment provided all the infrastructure needed to store in a structured way fragments of TEX code from its LATEX sources. Its first version was writing an approximative XML code whose content was obtained through TEX's interpretation of those fragments of LATEX source code read in the actual sources of the articles (usually keyed by the authors themselves, thus hardly under control). We modified it slightly so that it would write a new LATEX structure (the innovation was thus mostly to write out LATEX environments instead of XML elements as grouping boundaries in the auxiliary document bearing metadata) whose content would be the literal (uninterpreted) TEX code for textual data. In fact, this was an easy task, and this now raised the entirely new problem of finding software able to transform a structured LATEX file (non-conforming to a predefined structure) into a valid XML file with whatever DTD would fit our project.

After evaluating many possible solutions, we began to test José Grimm's Tralics [7], and decided to adopt it as it not only provided solutions to the above issues, but opened new perspectives that soon became reality thanks to the high responsiveness of its author.

2.1 Enters Tralics

Tralics is a program that reads LATEX source code and outputs an XML file. Its first application area being the production of the yearly activity report of INRIA, a busy institute with more than 2,500 researchers forming more than 140 project teams, it is meant as a productivity tool that captures the structure and content from LATEX individual contributions, and creates uniform XML source files from which the multimedia facets of the report are generated (XHTML website, printable PDF, etc.). Given the wide range of activities at INRIA, and the wide range of researcher profiles, one can safely bet that most documented and undocumented features of LATEX have to be supported if lossless information is to be found in the final versions of the report, which are generated from the derived XML only. When we first tried it out, around June 2006, the support for tables, illustrations, bibliographic references, mathematical formulas was good. The design of Tralics, oriented toward the production an XML *source* file for further processing, makes it a perfect tool for a project like CEDRAM, because we aim at capturing content in an abstract structure, storing all relevant information in the structure, and discarding all irrelevant details like typographical adjustments in the conversion process.

Tralics includes

- a full TEX macros interpreter;
- a versatile translator from TEX-encoded text character to Unicode;
- a translator of mathematical formulas into presentation MathML;
- a BibTex file parser;
- a number of LATEX standard commands;
- a number of commands defined by popular LATEX packages;
- a mechanism to define or refine commands and behaviour through command-line options and various configuration files.

During the 6 months of development of CEDRICS, it gained

- a much more comprehensive and robust support of mathematical constructs including the AMS packages;
- a new mechanism for controlling the way mathematical font changes are expressed in MathML;
- a mechanism for rewinding input and interpreting it twice (this is useful if you want to process the same input with two different options, so that, for instance, you can store in two XML elements the (mostly) uninterpreted source code and its converted version);
- a proper treatment of typographical quotation marks, at last.

Tralics is thus very good in translating a LATEX (and possibly BibTex) structure into an XML structure. In most cases, it just does the right thing, forgetting print oriented LATEX heritage, replacing boxes and glue by rigidly nested elements. But Tralics is not a full replacement for LATEX in the electronic edition paradigm, as long as other output formats are expected: most of the formatting notions in LATEX are unimplemented—this goes as far as numbers: page numbers are clearly meaningless as everything goes onto the same XML page, but equation numbers, figure numbers, theorem numbers are lost! In INRIA's report, cross references are computed by a Perl script using Tralics provided identifiers. In our case, the reference source still being that of articles' PDFs compiled with LATEX, the XML version must be synchronised with them, not the other way around.

2.2 CEDRICS

The new CEDRAM production system is thus entirely built on top of two components: PdfTEX, with shell extensions enabled; and Tralics. The configuration of these components amounts to 9,000 lines of LATEX code, and 1,900 lines for Tralics (which means 1,800 lines using TEX syntax). The overall result got the nickname CEDRICS, which will be used throughout this report.

As we are dealing with mathematical research articles, using XML and Tralics as it is done at INRIA seemed completely out of scope: many mathematical articles contain constructs that do not have a MathML counterpart, and some routine structures such as a cross reference from an equation to another are not currently supported. We thus keep the LaTeX source and its PdfTeX's produced PDF as ultimate references for the mathematical content we publish. This implies that the metadata has to be compatible with these. We achieve this by using an intermediate auxiliary "LaTeX" file which contains all the metadata we want to capture. The metadata like page numbers, bibliographical references labels, which have been computed and used by LaTeX while compiling the article, are exported in interpreted form, while textual, static metadata is exported as the literal TeX string that has been keyed in the article's source file. Tralics runs on this auxiliary file and produces the definitive XML metadata for the article.

For good reasons I am not going to repeat here, the whole operation is always performed on a completed *issue* or *physical volume*. It is only a matter of reshuffling fields and rearranging articles to get the XML file for the issue created conforming to the CEDRAM DTD. Finally, we get the best of both worlds, without compromise:

1. authors and most of our fellow editors, who know LaTeX and have no clue about XML, prepare, edit, check and correct proofs with the formats they are used to (those who use the amsart or smfart LaTeX classes are almost compatible with our production tools at once);
2. this yields the validated paper and electronic versions of the issues to be published;
3. the LaTeX system extracts the content that is exploited as metadata in our system, and generates the dynamic metadata as well;
4. Tralics derives an abstract version, which is imported in the CEDRAM database.

Compared to the previous system, the general picture has not changed. In fact, the input format did not change at all on this occasion, so that journals that had joined us at an early stage didn't notice the move. What has really changed is the versatility of this new system, and the productivity gains thanks to the fact that Tralics speaks TeX natively, so that once configured for proper action on legal TeX code, this code is supported in our production workflow. Moreover, as the configuration only requires basic skills in TeX macro programming, it is easily managed: I implemented the `\cite` command in ten minutes when it appeared in an abstract.

The output format (the CEDRAM XML DTD) changed only slightly: support for MathML, some new elements recording font selection, a new structure for bibliographies, so that the migration of the publishing platform was relatively straightforward. A large part of the effort was indeed consumed for testing the MathML support in various browsers and operating systems, and finding reasonable fallbacks. Tralics had to be changed internally many times before it would output MathML code behaving as expected with the two environments we wanted to support (Internet explorer with MathPlayer plugin, and Gecko based browsers with MathML enabled and specific mathematical fonts installed). For instance, it is currently impossible to use a uniform treatment for all mathematical font changes that can be represented either through the mathvariant attribute, a MathML entity for each character, or the (theoretically equivalent) Unicode character. Moreover, one can find cases where a MathML-ready browser cannot display an XHTML page with MathML unless a very specific list of conditions is met (where the MIME-type of served files has an important role), which can make it impossible to read the same page with another MathML-ready browser with conflicting requirements.

2.3 New modes of operation, or: *Toward a LaTeX metadata editor?*

A side effect of using Tralics is that we have now *two* intelligent agents passing over our content in order to produce the metadata.

The first pass (LaTeX) selects metadata to be exported or computed, and places it according to an *ad hoc* structure. This already requires some intelligence when you know how author's informations are stored in amsart, thus in cedram (see Table 1).

The second pass exploits the file written out in the first pass, using alternative definitions of many macros so that they produce better XML. For instance, as an abstract is expected to be presented on the website, we allow Tralics to translate some formatting instructions (like <p> elements) to the XML. but this is forbidden and stripped out of author's names or addresses. Here is the definition of the `killparcode` macro, e.g.:

```
\newcommand\spaceop[1][]{\space}%
\def\killparcode{%
   \def\\{\@ifstar{\spaceop}{\spaceop}}
   \let\par\space
   \let\newline\space
   \ignorespaces
}
```

which supports the full syntax of \\ and converts any such instruction an author might input into a single space.

Another consequence is that we now have two major modes of operation, and two optional formats for entering bibliographies into the system.

The two operative modes are

1. The native CEDRAM mode, which means that articles use the cedram LaTeX class, and the system produces the metadata while typesetting the PDF.
2. A metadata-only mode, developed for the recovery of born-digital editions with non-exploitable metadata, which means that a standard issue preparation tree is presented to CEDRICS, where articles use the cedram LaTeX class with the `Recup` option: The system produces the metadata from what it finds in the article file, but does not typeset the PDF except for the first "cover" page, that is generated on this occasion. To the CEDRICS system, this mode does not differ from the native one, because the same process produces the same files in the same formats, which can be readily

Table 1. CEDRAM author information: LaTeX and XML structures

Original LaTeX (cedram.cls structure)

```
  \author{\firstname{Ben} \lastname{Green}}
  \address{%
    School of Mathematics\\
    University of Bristol\\
    ...}
  \email{b.j.green@bristol.ac.uk}
  \urladdr{http://www.maths.bris.ac.uk/~mabjg/}
  \author{\firstname{Terence} \lastname{Tao}}
  \address{%
    Department of Mathematics\\
    University of California at Los Angeles\\
    Los Angeles CA 90095, USA}
  \email{tao@math.ucla.edu}
  \urladdr{http://www.math.ucla.edu/~tao/}
```

Pseudo-LaTeX code for Tralics (PdfLaTeX output with cedram.cls)

```
 \begin{xmlelement}{auteur}
     \xbox{prenom}{Ben}
     \xbox{nom}{Green}
{\killparcode\begin{xmlelement}{adresse}School
          of Mathematics\\ University ...\end{xmlelement}}
     \xbox{mel}{b.j.green@bristol.ac.uk}
     \xbox{url}{\url{http://www.maths.bris.ac.uk/~mabjg/}}
 \end{xmlelement}
 \begin{xmlelement}{auteur}
     \xbox{prenom}{Terence}
     \xbox{nom}{Tao}
{\killparcode\begin{xmlelement}{adresse}Department
           of Mathematics\\  University ... \end{xmlelement}}
     \xbox{mel}{tao@math.ucla.edu}
     \xbox{url}{\url{http://www.math.ucla.edu/~tao/}}
 \end{xmlelement}
```

XML output from Tralics (cedramarticle.dtd)

```
<auteur>
  <prenom>Ben</prenom>
  <nom>Green</nom>
  <adresse>School of Mathematics  University of ...</adresse>
  <mel>b.j.green@bristol.ac.uk</mel>
  <url>http://www.maths.bris.ac.uk/~mabjg/</url>
</auteur>
<auteur>
  <prenom>Terence</prenom>
  <nom>Tao</nom>
  <adresse>Department of Mathematics  University of ...</adresse>
  <mel>tao@math.ucla.edu</mel>
  <url>http://www.math.ucla.edu/~tao/</url>
</auteur>
```

input in the CEDRAM database. It requires some manual editing, typically done by copy-pasting from the original source file when available, but provides also an environment for keying metadata in CEDRAM format. When you're more familiar with LaTeX and emacs than with XML and MathML, this is quite a powerful way to do so!

Regarding bibliographies, the novelty is that BibTex is no more required, which lowers a lot the barrier into the project. When no BibTex file is used, the LaTeX pass detects it and copies thebibliography environment verbatim into the auxiliary file. It also sets a boolean that Tralics will recognise: it will thus parse the environment, and capture each bibliographical reference into a flat element (with formatting instructions). Compared to metadata generated from BibTex, the change is hardly noticeable to our users, as display and interlinking perform just as well.

2.4 Production summary

So, how is it to produce a nice electronic edition with high-end features for some academic journals? This proved to be kind of a nightmare until we had the current tools that make it at last a reasonable task. Table 2 gives an overview of the number of different strategies and techniques we tried out in order to attain our goals. In the end, all the metadata that could be engineered through Tralics enjoyed it, and that's it. All our mathematical metadata is in MathML, and it is pretty well handled by current browsers as long as they are properly configured. In any case, we also provide an HTMLised TeX version for our users who cannot understand MathML.

The NUMDAM years were easy: the metadata from that period is sufficient, it is relatively cheap to produce and easy to manage, it has drawbacks (like bibliographies somewhat arbitrarily tagged, and flatten out to sometimes improper Unicode strings) but none of them prevent comfortable usage.

The native CEDRAM years were fun: although finding the right way of doing things and the right people to do them was a hard job, once running everything was smoother than expected. Even the possibility of enhancing the metadata after the issues were published by just taking advantage of Tralics improvements because the source files were compatible with whatever new features we might come up with was a joy.

The inbetween period that has been described as "retro-born-digital" was a pain. Sometimes, we stopped NUMDAM digitisation at an early date because supposedly knowledgeable and responsible people told us to do so. Then we had to face electronic collections from the prehistoric digital age, with no usable source files, with unusable PS or PDF files, with metadata reduced to a bare minimum that did not meet any of our expectations. We tried out many different strategies for these legacy digital collections, usually unrecoverable. The only one I would advise to a newcomer is to digitalise this all anew, starting from scratch bitmaps, as if it were paper. Every other path is much too much time and brain consuming.

Table 2. The CEDRAM production segments chart

	Retrodigitisaion				*Born-digital recovery*		*Future-ready born-digital*		
	1997	2000	2001	2002	2003	2004	2005	2006	2007
AIF	*NUMDAM* 1,811 art. 51,034 p.		*Mixed strategies* 359 art. 11,056 p.					*CEDRICS (+B)* 218 art. 6,910 p.	
AFST	*NUMDAM* 1,109 art. 38,400 p.				*NUMDAM again*			*CEDRICS (R, +B)* 26 art. 836 p.	*CEDRICS (R, −B)* 29 art. 736 p.
AMBP	*NUMDAM* 177 art. 1,632 p.				*CEDRICS (R, +B)* 45 art. 922 p.			*CEDRICS (+B)* 32 art. 870 p.	
JTNB	*NUMDAM* 431 art. 7,700 p.					*CEDRICS (R, −B)* 98 art. 1,818 p.		*CEDRICS (−B)* 82 art. 1,550 p.	
JEDP	*NUMDAM* 469 art. 5,580 p.		*Mixed strategies* 45 art. 400 p.			*CEDRICS (R, −B)* 13 art. 204 p.	*CEDRICS (−B)* 26 art. 510 p.		
TSG	*NUMDAM* 266 art. 3,800 p.						*CEDRICS (R, −B)* 13 art. 260 p.		*CEDRICS (−B)* 15 art. 240 p.
SEDP	*NUMDAM* 640 art. 9,400 p.	*CEDRICS (R, −B)* 251 art. 3,650 p.							
Total	*NUMDAM metadata* 4,903 art. 117,546 p.				*CEDRICS metadata* 848 art. 18,506 p.				

Notes:

1. All production from year 2006, that was initially performed with CEDRAM, v. 1, has been regenerated with CEDRICS. All variants of the CEDRICS system appear in the table: an R means "recovery mode" (i.e. only metadata is generated, the preexisting PDF is not typeset anew); +B means that the bibliographic references are tagged using BibTex; while −B means that they are generated from flat LaTeX "thebibliography" environment.
2. In fact, all TeX metadata available for collections predating CEDRICS has been reengineered with special versions of it, so that all titles and abstracts are available with MathML formulas when applicable (AIF since 1949, JTNB since 1989).
3. The period 2001-2005 of AIF results of successive attempts with varying success: as the journal had already a website, metadata and PDFs were first taken there. It appeared soon that both metadata and PDFs had unrecoverable weaknesses (like wrong title, authors missing, wrong page numbers; wrong PDF with missing pictures, special characters, etc.). The bibliographies were extracted, with a lot of manual editing, from the plain TeX source files to the extent possible. Finally, quality was so uneven that years 2001-2004 were digitised, but some hand-made metadata was still used after another round of manual checking!

As “NUMDAM boss”, I choosed year 2000 as the last digitised year for most of the journals we dealt with. The rationale was that, as they were typically produced digitally since 1990, it should be reliable as of 2001. This was a vastly wrong assumption. Even some exceptional journals distributed by big commercial companies proved to have no metadata up to year 2003. In the worst cases, we have used our recovery mode to produce brand new metadata by extracting text from PDFs (possibly translated from PS with distiller, or even DVI files with dvipdfm) and turning this into proper TEX code, thereafter generating clean MathML.

3 Full Text Experiments

Given the advances in browsers, it seems to us that XHTML, with MathML and SVG enabled, is a credible publication format which could easily be generated on-the-fly from a structured XML source. The main obstacle in this direction is the heavy graphic dimension of many mathematical research articles, making use of invented symbols, special fonts, figures and diagrams. One could hope that a limited set of LATEX supported graphical languages like METAPOST, XY-pic, PGF, be supported by Tralics and produce pure SVG. This would open the door to more accessible mathematical research, more usable for persons with disabilities, but probably also more easily “understood” by automated agents and parsers.

We would like to come up with some sort of structured full-text XML format that could serve as a better metadata (for structure-tuned, math-aware searching or ranking) than the flat text extracted from the PDF we currently use. For instance, one can imagine that the weight of a word used in definitions or theorems should be high. In digitised texts, this could be obtained through font recognition (italics are highlighted items); in born-digital texts, the structure should provide this information.

The ultimate goal of our experiments would be to produce some XML form of the full text that could be simultaneously used for all purposes (from metadata to end-user consumption, in a hopefully accessible manner). While software exists that can convert any unsupported construct to a bitmap image, it seems completely out of sight to convert everything in a current article to pure XML. Nevertheless, we support further investigation in two directions:

1. implementing SVG output from Tralics, so that XHTML with MathML and SVG could be generated from LATEX source;
2. enriching the graphical PDF output with parallel alternative representations stored in additional layers keyed to the pages content in such a manner that one could fallback on a more accessible version of a diagram or formula.

As long as text only is concerned, Tralics is already quite impressive. We developed the Tralics counterpart of the cedram class (800 lines of TEX-like code), so that our full texts could be processed. For instance, we extended Tralics’ theorems in such a way that they use the same counter as LATEX would do, and store its value as an attribute to the `theorem` element. This can handle standard CEDRAM LaTeX source code up to unsupported constructs, which are unfortunately frequent. In many occasions, a solution can be found by redefining a bizarre LaTeX macro to produce some Unicode character string. An example is given by the AMS-LaTeX macro `\rdots`:

```
\newcommand{\rdots}{\mathinner{%
  \mkern1mu\raise1pt\hbox{.}%
  \mkern2mu\raise4pt\hbox{.}%
  \mkern2mu\raise7pt\vbox{\kern7pt\hbox{.}}\mkern1mu}}
```

which is nothing but rising dots to be used in a matrix. This definition has no obvious XML/MathML counterpart, but this is in fact a Unicode character with named entity “utdots”.

A more funny example is given by the following code found in a real article:

```
\[L=\begin{pmatrix}
                 A&\vline&B\\
                 \hline
                 C&\vline&D\\
     \end{pmatrix}\]
```

which yields

$$L = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right).$$

Although a quite a pedagogical presentation, it seems to be absolutely impossible to guess a correct MathML realization of this picture from its actual LaTeX source, i.e. convert some entries into optional presentation attributes...

```
  <mtable columnlines="solid" rowlines="solid">
```

4 Conclusions

Producing or archiving properly an electronic journal of mathematics is not an easy task. Authors write in some dialect of TeX and don’t want to learn any new method until it is sufficiently widespread and easy to use. For academic research in mathematics, LaTeX has no competitor today. Publishers want to have a distinctive layout and quality typesetting, as no generic abstract schema for representating mathematical knowledge (at the level of current research in every field...) is likely to emerge in a foreseeable future. Librarians, content agregators and providers, information professionals, computer scientists want clean metadata that can be seamlessly integrated into general systems which have not been conceived for mathematical texts.

The CEDRICS system we have described provides an unexpected solution to these challenges by using author’s LaTeX sources, minimal edits, and generating XML metadata with presentation MathML (and XML-ized mathematical TeX code). This way, freshly published mathematical articles can be easily published on the Web, or made interoperable with services far removed from the mathematical community using standard protocols such as OAI-PMH.

It has gone as far as being useful in producing metadata for articles that had no LaTeX sources at all, but for which typesetting the abstract in XML with MathML and XML-ized TeX would have been a much more demanding task, at least for the mathematically oriented persons who did this. In this sense, it actually helped a lot to augment the Digital Mathematics Library with new articles from CEDRAM and, more surprisingly, from other sources.

The next step our experiments lead to is to generate the full text of articles in such a way that the whole mathematical content could be exposed directly on the web in a less graphically oriented format than PDF. This would lead to obvious advances in accessibility and retrievability for this content. But, although faithful conversion from mathematician's LaTeX source code to XML with Tralics has proven to be manageable up to the abstract and bibliographical reference lists, a fully exploitable output is currently out of sight, except for special areas where the mathematical vocabulary is under control. By "fully exploitable output", we mean that the XML version could be used as an authoritative reference for the whole article's mathematical content, fully avoiding any artificial use of figures to represent unsupported structures. Nevertheless, we think these experiments should be pursued, at least in order to generate some XML version of the full text (considered as a metadata) that could follow the same standards as the output of mathematical OCRs, yet being derived from the actual sources, thus hopefully more accurate. For this goal, it is likely that a working environment very different from CEDRICS would be needed; alternatives to Tralics should be evaluated (an obvious candidate is LaTeXML [8]), but INRIA's Raweb environement (see [9]) could also be adapted for this purpose.

References

1. Bouche, T., Laurent, Y., Sabbah, C.: L'édition sans drame. Gazette des mathématiciens **108** (April 2006) 86–88.
2. Bouche, T.: Toward a digital mathematics library? In Borwein, J., Rocha, E., Rodrigues, J., eds.: Communicating mathematics in the digital era, AK Peters Ltd (2008) 47–73.
3. Bouche, T.: A PdfLaTeX-based automated journal production system. TUGboat **27**(1) (2006) 45–50.
4. Jacquier-Roux, P.: RUCHE, an editorial flow management tool. (2006). `http://ruchedemo.cedram.org/`.
5. Růžička, M.: Automated Processing of TeX-Typeset Articles for a Digital Library. (2008) In: Sojka Petr (editor): *DML 2008 – Towards Digital Mathematics Library*, Birmingham, UK, July 27th, 2008, pp. 167–176.
6. Carlisle, D.: XII. TUGboat **19**(4) (1998). 348. `http://www.tug.org/TUGboat/Articles/tb19-4/tb61carl.pdf`.
7. Grimm, J.: Tralics, a LaTeX to XML Translator. TUGboat **24**(3) (2003). See `http://www-sop.inria.fr/apics/tralics/`.
8. LaTeXML: A LaTeX to XML Converter `http://dlmf.nist.gov/LaTeXML/`
9. Grimm, J.: Tralics and the Raweb `http://www-sop.inria.fr/miaou/tralics/raweb.html`