CAHIERS GUTenberg

PRODUCTION DE MÉTADONNÉES MATHML POUR DES ARTICLES DE RECHERCHE EN MATHÉMATIQUES : L'EXPÉRIENCE DU CEDRAM

Thierry BOUCHE

Cahiers GUTenberg, nº 51 (2008), p. 61-76.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_2008___51_61_0>

PRODUCTION DE MÉTADONNÉES MATHML POUR DES ARTICLES DE RECHERCHE EN MATHÉMATIQUES : L'EXPÉRIENCE DU CEDRAM

Thierry BOUCHE

RÉSUMÉ. — On décrit CEDRICS, un système de production automatisée de revues scientifiques entièrement piloté par des formats LaTeX. Après un survol rapide du contexte et des principes qui ont guidé sa mise en œuvre, on s'efforce de montrer que le système est particulièrement efficace grâce à la possibilité qu'offre LaTeX de paramétrer simultanément une typographie de qualité et des métadonnées fidèles et complètes. Cela a été possible en combinant deux interpréteurs de code LaTeX de vocations très différentes mais qui embarquent un processeur de macro TeX complet : PdfLaTeX de Hàn Thế Thành, et Tralics de José Grimm.

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.

1. INTRODUCTION

1.1. LE PROJET CEDRAM

Le projet CEDRAM de la Cellule MathDoc (unité mixte de services du CNRS et de l'université Joseph-Fourier à Grenoble) a été lancé pour soutenir les éditeurs indépendants de journaux mathématiques en France.

Cahiers GUTenberg n° 51 — Octobre 2008, p. 61-76.

La forme visible de ce projet est un portail rassemblant l'édition électronique d'un ensemble de revues (`www.cedram.org`). Le soutien passe par le développement d'un environnement de production performant constitué d'une boite à outils modulaire, et par la mise en ligne des articles. Il en existe aussi une forme invisible : les *Cahiers GUTenberg* sont produits et diffusés avec les outils du CEDRAM ; ces outils ont aussi été adoptés par des collègues tchèques pour la production de revues comme *Archivum Mathematicum*.

Une caractéristique notable de l'outil de préparation des fascicules est qu'il est entièrement écrit en LaTeX. Ce choix peu commun est justifié par l'observation que seul TeX est en mesure d'interpréter le contenu d'un article écrit en TeX, et partant d'en extraire des métadonnées exactes [1].

1.2. Les outils

Quatre modules principaux ont été développés dans le cadre de ce projet, qui sont en production depuis quelques années déjà :

1. RUCHE, un outil de gestion des flux rédactionnels qui permet à une revue de superviser tout le processus éditorial de la soumission des articles, l'arbitrage du comité de rédaction jusqu'à la préparation du volume à paraître, le tout au travers d'une interface web [3].

2. CEDRICS, un système de production de revues en LaTeX qui automatise tout ce qui peut l'être (folios, génération de métadonnées…). Il produit le PDF des pages intérieures et la couverture pour l'imprimeur, le PDF écran de chaque article, les métadonnées XML détaillées du volume dans un format dual (le texte est en Unicode dans une structure voisine de la TEI-Lite, les expressions mathématiques sont disponibles à la fois en MathML de présentation et en pseudo code LaTeX).

3. EDBM, un système complet d'indexation et de fouille utilisant ces métadonnées.

1. Dans ce texte, le terme *métadonnées* revêt son sens usuel dans le domaine de la documentation : il fait référence à toutes les informations décrivant le contenu d'un article comme son titre, le nom de ses auteurs, sa pagination, les informations bibliographiques de parution, mais aussi les mots clés, les résumés ou la bibliographie. Une version XML du texte intégral d'un article est aussi considérée comme une métadonnée si c'est le contenu du PDF qui fait référence.

4. Une interface web exploitant cette base de données, qui génère les pages dynamiques permettant de naviguer dans les sites des revues.

L'objet de cet article est de donner quelques précisions sur le deuxième outil.

2. CEDRICS

2.1. La classe Cedram

J'ai écrit au cours de l'été 2005 une première version des outils LaTeX du CEDRAM, qui était pour l'essentiel une version modifiée de la classe amsart, comportant quelques innovations pour automatiser la production et la collecte de métadonnées exactes [1]. Les deux nouveautés (relatives) du moteur PdfTeX qui permirent cela étaient la possibilité de lancer des commandes en cours de compilation (par `\write18`) d'une part, et d'inclure des PDF multipages d'autre part.

Le principe gouvernant ce système est très simple : une *revue* (scientifique) est un ensemble dénombrable de *volumes physiques* utilisant un même maquette (parfois numérotés à la file comme les *Cahiers*, mais souvent publiés en plusieurs livraisons — ou fascicules — regroupées par tomes annuels) dont l'essentiel du contenu est une liste ordonnée d'*articles*. Pour éviter les incohérences entre plusieurs supports (revue imprimée, sommaire en ligne, métadonnées exportées pour des partenaires…) et se prémunir contre les changements de dernière minute, il faut avoir une seule source pour chaque information, et en dériver toutes ses utilisations à l'aide de programmes.

Un exemple d'information qui appartient à la revue et à elle seule est son titre, son ISSN : les articles qui affichent ces informations dans leur maquette doivent les hériter de la revue, c'est en général réalisé en utilisant une classe spécifique ; pour nous ce sera une option de la classe cedram. La tomaison (tome, fascicule, mois et année de parution…) est une information pertinente au niveau volume : les articles doivent donc l'hériter du volume dans lequel ils paraissent. Si on copie ces informations dans les articles eux-mêmes, on court le risque de ne pas les modifier lorsque, pour une raison ou une autre, l'article change de volume (et surtout d'oublier d'en répercuter les conséquences sur les autres articles). Enfin, les informations sur un article comme le titre, les noms d'auteurs, le résumé font évidemment partie de l'article. En revanche, si l'on y réfléchit un peu, on s'aperçoit que les numéros de page, à part

le premier d'entre eux dans le cas des revues qui poursuivent la numérotation des pages sur un tome paraissant en plusieurs fascicules, est le résultat d'un ensemble de paramètres (le folio de la première page du volume, le matériau folioté précédent l'article, notamment tous les articles qui le précèdent et leurs longueurs, des options de maquette comme le fait de démarrer ou non un article en belle page, etc.). Ce numéro (singulièrement le folio de la dernière page d'un article) est très souvent faux lorsqu'il est copié à la main (au sommaire, notamment). On voit ainsi assez souvent des articles démarrant en page 19 à la suite d'un article qui finit en page 19 ou 20, car quelques commentaires ont dû être ajoutés sur épreuves ! Par suite, les folios doivent être calculés : le sommaire d'un volume est finalement un produit assez sophistiqué qui ne peut être produit qu'a posteriori, en prenant en compte ces deux niveaux d'informations que sont, outre la définition de la revue, le volume et chacun des ses constituants.

Au CEDRAM, la production d'un volume de revue est donc entièrement contrôlée par un système hiérarchisé de fichiers LaTeX.

— Les articles sont placés dans des répertoires séparés et sont autosuffisants à l'exception des informations provenant du volume dans lequel ils seront publiés (tomaison, titre éventuel…). Cela signifie par exemple que deux auteurs peuvent utiliser chacun un jeu de macros personnelles appelé `input.tex`, des versions incompatibles d'extensions standard, sans qu'il y ait de conflit.

— Un volume est pour l'essentiel une collection ordonnée d'articles : c'est un répertoire contenant les répertoires de tous les articles à inclure et le fichier qui définit dans une syntaxe LaTeX *ad hoc* tous les paramètres de ce volume.

— Dans ce schéma, l'unité de production étant le volume, il n'a pas semblé nécessaire de continuer à empiler des répertoires au-dessus des volumes : la revue et ses constantes (structure d'un fascicule, maquette, pages de titre, ours, présentation des sommaires, de la couverture, etc.) sont traitées comme des fichiers de style, et activées par une option de la classe cedram.

La compilation du fichier volume lance les compilations individuelles de tous les articles et les inclue dans la foulée, produisant en une passe le PDF des pages intérieures. Du coup, LaTeX connaît les informations

```
\let~\catcode~‘76~‘A13~‘F1~‘j00~‘P2jdefA71F~‘7113jdefPALLF
PA’’FwPA;;FPAZZFLaLPA//71F71iPAHHFLPAzzFenPASSFthP;A$$FevP
A@@FfPARR717273F737271P;ADDFRgniPAWW71FPATTFvePA**FstRsamP
AGGFRruoPAqq71.72.F717271PAYY7172F727171PA??Fi*LmPA&&71jfi
Fjfi71PAVVFjbigskipRPWGAUU71727374 75,76Fjpar71727375Djifx
:76jelse&U76jfiPLAKK7172F7117271PAXX71FVLnOSeL71SLRyadR@oL
RrhC?yLRurtKFeLPFovPgaTLtReRomL;PABB71 72,73:Fjif.73.jelse
B73:jfiXF71PU71 72,73:PWs;AMM71F71diPAJJFRdriPAQQFRsreLPAI
I71Fo71dPA!!FRgiePBt’el@ lTLqdrYmu.Q.,Ke;vz vzLqpip.Q.,tz;
;Lql.IrsZ.eap,qn.i. i.eLlMaesLdRcna,;!;h htLqm.MRasZ.ilk,%
s$;z zLqs’.ansZ.Ymi,/sx ;LYegseZRyal,@i;@ TLRlogdLrDsW,@;G
LcYlaDLbJsW,SWXJW ree @rzchLhzsW,;WERcesInW qt.’oL.Rtrul;e
doTsW,Wk;Rri@stW aHAHHFndZPpqar.tridgeLinZpe.LtYer.W,:jbye
```

On the first day of Christmas my true love gave to me
a partridge in a pear tree.
On the second day of Christmas my true love gave to me
two turtle doves
and a partridge in a pear tree.

Figure 1: Un source TEX dû à David Carlisle dont la compilation produit un texte parfaitement lisible de 446 mots sur 90 lignes (au-dessous, les cinq premières).

provenant du volume en cours de compilation (dont le numéro de la première page du prochain article à traiter), et peut les passer à l'article en question (à l'aide d'un fichier de configuration écrit dans son répertoire) avant de lancer sa compilation. Comme toutes les informations sur le volume sont à la disposition de LATEX lors d'une seconde passe, il produit au passage tous les fichiers annexes nécessaires à la publication de ce volume comme les sommaires, la couverture, les métadonnées.

L'idée maîtresse qui a conduit à cette architecture est que le moteur TEX est le seul logiciel qui soit en mesure de faire des prédictions (donc d'écrire des métadonnées) sur le résultat de la compilation d'un fichier source TEX. Le grand classique rappelé en figure 1 donne l'intégralité d'un code TEX qui, compilé, étale une comptine en parfait anglais sur deux pages… Pourtant, tout analyseur syntaxique naïf le décrirait plutôt comme un message crypté à l'aide d'une variante d'UUencode ou Binhex !

L'émulation du processeur de macros TEX ou le recours à des heuristiques pour déduire (a priori aussi bien qu'a posteriori) des informations sur la forme et le contenu d'un fichier TEX est une source inévitable d'erreurs lorsque l'on ne contrôle pas *tout* le processus, en particulier les macros utilisées par les auteurs. Il est donc beaucoup plus sûr de structurer les informations nécessaires dans le fichier source pertinent, et de demander à TEX de les interpréter. Les solutions existantes que j'ai pu regarder avant de me résoudre à inventer un nouveau système avaient toutes des problèmes car elles n'assuraient pas une synchronisation parfaite entre ce qui allait être mis sur le web et imprimé.

Les éléments de base de l'interface utilisateur et les principes de fonctionnement du système n'ont pas évolué depuis sa première version décrite dans [1]. L'organisation des fichiers sources, la structuration et la répartition des métadonnées au sein de ces fichiers, les modifications de la classe amsart pour la rendre indépendante de la maquette, ainsi que les mécanismes de base permettant l'export de métadonnées ont prouvé leur solidité sur une dizaine de revues et plus d'un millier d'articles.

Mais la première version avait une limitation majeure, car elle essayait de faire écrire directement par TEX les métadonnées XML, qu'il fallait ensuite retravailler plus ou moins au petit bonheur.

Lorsque TEX écrit dans un fichier auxiliaire un morceau de code LATEX stocké en mémoire dans une macro, il est assez difficile de s'assurer que le résultat soit du XML valide (il faut développer certaines macros, en protéger d'autres, ne pas interpréter les formules de maths, contrôler le codage des caractères utilisés en sortie, éviter les caractères qui sont spéciaux en XML, etc.). La fonction d'écriture des signets incluse dans l'extension hyperref donne une idée de ce que l'on peut faire dans ce domaine, sans aller jusqu'à fournir un convertisseur LATEX vers XML complet écrit en TEX ! Certains champs comme les pages étaient donc calculés et écrits par TEX dans leur forme définitive, tandis que pour d'autres comme les résumés c'est le code brut présent dans le source qui était enregistré ; les noms d'auteurs subissaient un traitement intermédiaire, avec le risque de ne pas toujours gérer parfaitement certains caractères exotiques… Au final, ce pseudo-XML devait être retraité pour être exploitable, ce qui était lourd et inefficace, pour un résultat médiocre (utilisation de latex2html !).

Par ailleurs, du fait que les métadonnées étaient exportées par TEX lui-même lors de la compilation d'un article, une structuration plus fine de certains éléments vus par TEX était hors de portée. Ceci nous a conduit à imposer l'utilisation de BibTEX et produire les bibliographies XML à l'extérieur de la chaîne de traitement principale, contrairement aux principes précédemment énoncés !

2.2. Tralics

L'idée de produire le XML en utilisant un traducteur dédié s'est alors imposée : au lieu de demander à TEX lui-même de chercher à fabriquer du XML approximatif tout en compilant le PDF de l'article, une modification mineure du système d'export permettait d'écrire un fichier de métadonnées entièrement codé en TEX, la structure étant dictée par la classe cedram et plus proche de la DTD cible, le contenu des champs textuels étant le code brut provenant de l'article sans aucune interprétation de la part de TEX.

Après quelques tests, Tralics [2] s'est avéré être une solution prometteuse. La conception de Tralics comme outil de production d'un XML pivot destiné à produire par la suite des formats variés en fait en effet un outil parfait pour capturer le contenu et le conserver dans une forme générique.

L'article de José Grimm dans ce volume donne le contexte de son développement à l'INRIA et quelques exemples de son fonctionnement. Je vais ici rendre compte de l'expérience assez extraordinaire que ça aura été pour moi de découvrir peu à peu le potentiel de ce logiciel, de l'apprivoiser malgré la forme un peu rebutante de la documentation existante, de voir les nouvelles fonctionnalités dont j'avais besoin introduites très rapidement, et de pouvoir implémenter immédiatement celles que je pouvais coder directement en TEX, pour disposer au final d'un système qui marche.

Lorsque j'ai pris contact avec José, la principale application de Tralics, sinon la seule, était la production du rapport d'activité de l'INRIA. Quand on considère le nombre d'équipes et à la diversité de profils de chercheurs de cet institut, on se dit que ce doit être l'épreuve du feu pour un tel logiciel. Ce d'autant plus que la décision de passer toutes les sources en XML, et de produire les pages XHTML comme le PDF à partir du XML produit par Tralics impose qu'aucune information ne

soit perdue. De fait, lors de mes premiers essais, en juin 2006, la conversion de la structure des classes LaTeX, des tables, des références bibliographiques, et des formules mathématiques était bonne. En revanche, le XML produit n'était jamais exploité directement, mais soit retravaillé par un script Perl (qui d'une certaine façon faisait à partir du XML de Tralics ce que latex2html fait à partir d'un source LaTeX) pour la version XHTML, soit interprété par xmltex (qui n'est pas le processeur XML/MathML le plus strict du marché) pour le PDF.

Tralics comporte

— un interpréteur complet de macros TeX ;

— un traducteur puissant de caractères codés à la TeX (si l'on peut appeler ça un code !) vers Unicode ;

— un traducteur de formules mathématiques TeX en MathML de présentation (il est possible de produire du MathML de contenu, mais alors il faut l'écrire soi-même) ;

— un parseur de fichiers BibTeX ;

— la traduction d'un grand nombre de constructions LaTeX standard ;

— la traduction d'un grand nombre de constructions définies par des extensions populaires de LaTeX ;

— un ensemble de commandes spécifiques pour gérer les éléments et les attributs crées dans le XML cible ;

— un mécanisme pour définir ou paramétrer certaines commandes à l'aide d'options de ligne de commande ou de fichiers de configuration.

Au cours des 6 mois de développement de CEDRICS, Tralics a reçu quelques améliorations :

— un support plus complet et robuste pour les expressions mathématiques courantes, y compris celles définies par les extensions de l'AMS ;

— un nouveau mécanisme pour contrôler l'effet des changements de police mathématique en MathML ;

— un mécanisme qui permet de « rembobiner » le contenu d'un environnement pour le faire interpréter deux fois (sans avoir figé les catcodes du contenu, par exemple), ce qui permet d'avoir en une passe l'équivalent de deux passes sur un même texte avec des options complètement différentes ;

— un traitement typographiquement correct des guillemets et apostrophes !

Tralics est donc un outil redoutable pour convertir un article LaTeX standard avec son éventuel BibTeX vers une structure XML pivot, mais il traite aussi bien du code TeX de bas niveau et peut servir à écrire un fichier XML paramétré finement en TeX. Dans la plupart des cas, il fait exactement ce qu'il faut sans configuration particulière. Il oublie les instructions LaTeX n'ayant de sens que pour la représentation visuelle du contenu et stocke les informations dans une structure arborescente d'éléments.

2.3. CEDRICS : la rencontre de Cedram et Tralics

Le nouveau système de production du CEDRAM est désormais entièrement bâti à partir de deux composants : PdfTeX et Tralics. La configuration de ces composants pour nos besoins un peu spécifiques représente de l'ordre de 9 000 lignes de code LaTeX (dont la plus grande part provient d'amsart) et 1 900 lignes pour Tralics (dont 1 800 lignes en syntaxe TeX : j'utilise finalement très peu les possibilités offertes par le fichier de configuration).

Comme notre cœur de métier est la publication d'articles de recherche en mathématiques, monter une chaîne de production comme celle du rapport de l'INRIA en utilisant du XML comme pivot pour tous les traitements est à ce jour totalement hors de portée : beaucoup de nos articles comportent des constructions qui sont difficilement représentées en MathML (surtout si l'on souhaite en préserver la sémantique), et les ajustements fins de la mise en page sont très importants pour la clarté et la lisibilité des textes. La version de référence qui fait foi [2] est donc le PDF produit par PdfTeX. C'est en général ce qui est vérifié par l'auteur et le personnel de la revue. Le défi est donc de produire des métadonnées représentant aussi fidèlement que XML le permet le contenu du PDF. La technique qui consiste à écrire des métadonnées LaTeX pendant la compilation du volume, puis à les traduire le plus directement avec Tralics répond parfaitement à ce besoin.

Une conséquence intéressante de ce traitement en deux étapes est que nous avons maintenant *deux* agents intelligents qui passent sur du

2. Cette notion est très importante pour les textes de maths qui, une fois validés scientifiquement, peuvent servir de référence pour de nouvelles avancées à tout moment ultérieurement.

code LaTeX pour produire les métadonnées. La première passe (LaTeX) se contente désormais de sauver dans un *token* l'argument de certaines macros ou le contenu de quelques environnements et de l'écrire littéralement dans le fichier auxiliaire à l'intérieur d'une structure spécifique. Sur l'exemple d'un auteur (figure 2), on peut noter que cette première passe a déjà modifié la structure par rapport à celle du source LaTeX, qui est parfaitement plate, en plaçant les informations relatives à un auteur à l'intérieur d'un même environnement — on pourra aussi noter qu'à l'exception de la macro `\killparcode` explicitée ci-dessous, toutes les macros qui apparaissent sont connues de Tralics et correctement traitées sans aucune configuration. Pour les numéros de pages et les autres métadonnées *calculées* par LaTeX au fil de la compilation, le mécanisme est similaire si ce n'est que ça n'est pas `\the\c@page` qui est écrit dans le fichier pour Tralics, mais son développement.

La seconde passe (Tralics) prend pour source le fichier écrit lors de la première passe. Comme il s'agit d'un fichier généré, sa structure est connue par avance. En fonction de la nature des métadonnées capturées, les traitements sont adaptés en utilisant des variantes de `\xbox` et `xmlelement`. Par exemple, comme un résumé est du texte non contraint pouvant contenir des listes ou des formules centrées, Tralics conserve un certain nombre d'instructions permettant de les afficher (comme l'élément `<p>`), tandis qu'elles seront effacées si elles se trouvent dans un nom d'auteur ou dans une adresse. Voici la définition de la macro `\killparcode` donnée à Tralics pour ignorer tout saut de ligne (utile notamment dans la présentation de l'adresse dans l'article).

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

On note que tralics connaît aussi quelques commandes internes de LaTeX comme `\@ifstar`.

LaTeX original (structure cedram.cls)

```
\title{Sur $z^p$ comme somme}
\author{\firstname{Pierre} \vonname{de} \lastname{Fermat}}
\address{%
  Laboratoire de descente infinie\\
  Université de Toulouse\\
  ...}
\email{p.fermat@univ-toulouse.fr}
\urladdr{http://www.maths.univ-toulouse.fr/~fermat/}
```

Code LaTeX pour Tralics (sortie de compilation avec cedram.cls)

```
{\killparcode\begin{XTAelement}[fr]{titre} Sur $z^p$
  comme somme\end{XTAelement}}
  \begin{xmlelement}{auteur}
     \xbox{prenom}{Pierre}
     \xbox{particule}{de}
     \xbox{nom}{Fermat}
    {\killparcode\begin{xmlelement}{adresse}Laboratoire de
        descente infinie\\ Université [...] \end{xmlelement}}
     \xbox{mel}{p.fermat@univ-toulouse.fr}
     \xbox{url}{\url{http://www.maths.univ-toulouse.fr/~fermat/}}
  \end{xmlelement}
```

XML produit par Tralics (cedramarticle.dtd)

```
<titre xml:lang='fr'>Sur <formula type='inline'><math>
  <msup><mi>z</mi> <mi>p</mi></msup></math></formula> comme somme</titre>
<TeXtitre xml:lang='fr'>Sur <texmath textype='inline' type='inline'>
  z^p</texmath> comme somme</TeXtitre>
<auteur>
  <prenom>Pierre</prenom>
  <particule>de</particule>
  <nom>Fermat</nom>
  <adresse>Laboratoire de descente infinie Université
         [...] </adresse>
  <mel>p.fermat@univ-toulouse.fr</mel>
  <url>http://www.maths.univ-toulouse.fr/~fermat/</url>
</auteur>
```

Figure 2: Métadonnées titre et auteur du CEDRAM : structures LaTeX et XML.

LaTeX original

```
\begin{thebibliography}{99}
  \bibitem{BoSo} W. \textsc{Borchers}, H. \textsc{Sohr}\pointir
    ‘‘On the equation $\rot v = g$ and $\div u = f$
    with zero boundary conditions’’.
    \textit{Hokkaido Math J.}, \textbf{19} (1990), p. 67-87.
\end{thebibliography}
```

Code LaTeX pour Tralics

```
\xmlbibcitation{BoSo}
\xmlbibcite{BoSo}{1}
{\killparcode\begin{biblio}
  \bibitem{BoSo} W. \textsc{Borchers}, H. \textsc{Sohr}\pointir
  ‘‘On the equation $\rot v = g$ and $\div u = f$ with zero boundary
  conditions’’. \textit{Hokkaido Math J.}, \textbf{19} (1990) p. 67-87.
\end{biblio}}
```

XML produit par Tralics (versions MathML|pseudo-TeX)

```
<biblio type='flat'>
 <bib_entry user-id='BoSo' id='bid0' doctype='none'>
  <reference>1</reference>
  <bibitemdata>W. <hi rend='sc'>Borchers</hi>, H. <hi rend='sc'>Sohr</hi>.&#x2014;
   &#x201C;On the equation
    <formula type='inline'><math xmlns='http://www.w3.org/1998/Math/MathML'>
     <mrow>
      <mo form='prefix'>rot</mo>
      <mi>v</mi>
      <mo>=</mo>
      <mi>g</mi>
     </mrow>
    </math>
   </formula>
   with zero boundary conditions&#x201D;. [...]
   <hi rend='it'>Hokkaido Math J.</hi>, <hi rend='bold'>19</hi> (1990) p. 67-87.
  </bibitemdata>
 </bib_entry>
</biblio>
```

```
<texmath textype='inline' type='inline'>
   \operatorname{rot}v = g
</texmath>
```

Figure 3: Une bibliographie du CEDRAM : structures LaTeX et XML.

D'autres exemples des bénéfices de cette double passe sont les bibliographies. Lorsqu'elles n'utilisent pas BibTEX, LATEX sauve dans le fichier auxiliaire le contenu intégral de l'environnement sans pouvoir en modifier la structure comme dans le cas des métadonnées auteur. Tralics, en passant sur cet environnement, convertit chaque `\bibitem` en élément `<bibitem>` (figure 3). Dans le cas de bibliographies en BibTEX, je m'en remets à peu près intégralement à Tralics qui parse le fichier .bib et traduit sa structure en une base de données XML. Dans tous les cas, LATEX fournit à Tralics les éléments qu'il ne pourrait pas recalculer lui-même sans risque d'introduire des différences : la liste des références citées et les étiquettes utilisées par LATEX pour ces références (les commandes `\citation` et `\bibcite`, traditionnellement enregistrées dans le .aux, sont ici fournies à Tralics sous la forme visible en figure 3).

Enfin, j'ai évoqué cette possibilité pour Tralics de « rembobiner » le contenu d'un environnement après l'avoir lu une première fois. Cela revient en fait à laisser la possibilité d'une troisième passe Tralics sur certaines métadonnées. Elle est utilisée pour tout ce qui peut contenir des formules de mathématiques, car nous produisons alors une version duale des métadonnées : le texte est toujours en Unicode, mais les maths sont stockées en MathML et en TEX (qui est proposé sur le serveur pour les navigateurs qui ne peuvent pas afficher correctement du MathML). En fait, ce « TEX » est produit par Tralics, qui génère donc toujours du XML valide, ce qui nous permet aussi de le simplifier en interceptant un certain nombre de commandes. Des exemples sont montrés pour le titre d'un article ou d'un article cité dans les figures 2 et 3.

2.4. MathML

L'utilisation de Tralics a été l'occasion de tester des conversions relativement massives (plus de 1000 articles) de métadonnées riches en mathématiques. Lors des premiers essais, j'étais un peu dubitatif.

— D'une part car le support MathML des navigateurs restait un peu primitif, et imposait des conditions mutuellement exclusives entre MathPlayer et Mozilla. Par exemple les caractères « gras de tableau noir » (à double barre) ou gothiques, peuvent être spécifiés de trois façons distinctes en MathML (ℝ : caractère Unicode U+211D, entité `&Ropf ;` ou `<mi mathvariant='double-struck'>R</mi>`), mais aucune ne marche dans les deux navigateurs (au moins la dernière affiche

toujours un R !). Dans le même ordre d'idée, il y avait un pataquès avec les déclarations de type MIME nécessaires pour que MathPlayer s'active, mais qui provoquaient l'affichage du code source XML par Firefox…

— D'autre part car je tombais très souvent sur des formules dont l'affichage était inadapté, qu'un caractère Unicode affiche un carré (avec une certaine version de Firefox, cela concernait toutes les parenthèses !), ou que les alignements ou le placement des différents éléments d'une formule un peu complexe se trouvent si mal placés qu'on ne pouvait pas la déchiffrer.

Mais après s'être affranchi de latex2html et ses images, il ne me semblait plus possible de revenir en arrière. C'est pourquoi j'ai mis un peu d'énergie dans la création de métadonnées duales TEX/MathML, de façon à avoir une bouée de sauvetage dans le cas où le MathML ne serait pas satisfaisant, en attendant que le support s'améliore, tant dans Tralics que dans les navigateurs.

Finalement, José Grimm a mis dans Tralics tous les mécanismes de base qui permettent de faire un MathML de présentation raisonnable à partir d'AMS-LATEX. Le support des navigateurs et la couverture mathématique des polices Unicode se sont sensiblement améliorés, et le site du CEDRAM en tire largement parti (voir figure 4 et le site lui-même : `http://www.cedram.org/`).

3. Convertir aussi les articles ?

Étant donné le niveau atteint actuellement par les traducteurs, on est tenté de pousser la conversion un peu plus loin, c'est-à-dire de produire une version du texte intégral des articles en XML. On peut imaginer qu'une telle version, proprement structurée, serait la métadonnée ultime pour faire des recherches efficaces sur ce corpus (comme, par exemple, pouvoir rechercher un mot dans une définition, dans un énoncé, transformer les références à un théorème en liens inter-articles, etc.). Étant donné le niveau de support atteint par les navigateurs, on pourrait même envisager de publier une version XML des articles en plus du PDF. De fait, il me semble qu'un format XHTML dans lequel les formules seraient en MathML et les illustrations en SVG commence à être crédible pour diffuser des textes scientifiques sur internet.

Le principal obstacle réside pour nous dans le caractère hautement visuel de nombreux articles de recherche en maths, qui utilisent des

Alfred Rényi
Sur un théorème général de probabilité
Annales de l'institut Fourier, 1 (1949), p. 43-52
Article PDF | Analyses MR 14,886d | Zbl 0036.08703

Résumé - Abstract

L'auteur généralise un théorème qu'il a déjà donné (J. de Math. 28 (949)). Envisageant un champ de probabilités au sens de Kolmogoroff, il élargit puis étudie la notion de discrépance, en introduisant la discrépance $D_y(x)$ d'une variable aléatoire x par rapport à une autre variable aléatoire y ; elle se réduit au coefficient de corrélation si x et y sont des variables caractéristiques. Il introduit aussi la notion de suite de variables aléatoires "presque indépendantes deux à deux", avec un coefficient Δ dit module de dépendance. Il donne alors essentiellement pour une telle suite x_n l'inégalité

$$\sum_1^{\infty} D_y^2(x_n) \le \left(1+\Delta\right)\left[1+\left(\frac{e(y)}{\sigma(y)}\right)^2\right]$$

où $e(y)$ est valeur probable, σ, écart moyen.

(a) MathML : Internet Explorer 8 avec MathPlayer 2.1d

Alfred Rényi
Sur un théorème général de probabilité
Annales de l'institut Fourier, 1 (1949), p. 43-52
Article PDF | Analyses MR 14,886d | Zbl 0036.08703

Résumé - Abstract

L'auteur généralise un théorème qu'il a déjà donné (J. de Math. 28 (949)). Envisageant un champ de probabilités au sens de Kolmogoroff, il élargit puis étudie la notion de discrépance, en introduisant la discrépance $D_y(x)$ d'une variable aléatoire x par rapport à une autre variable aléatoire y ; elle se réduit au coefficient de corrélation si x et y sont des variables caractéristiques. Il introduit aussi la notion de suite de variables aléatoires "presque indépendantes deux à deux", avec un coefficient Δ dit module de dépendance. Il donne alors essentiellement pour une telle suite x_n l'inégalité

$$\sum_1^{\infty} D_y^2(x_n) \le \left(1+\Delta\right)\left[1+\left(\frac{e(y)}{\sigma(y)}\right)^2\right]$$

où $e(y)$ est valeur probable, σ, écart moyen.

(b) MathML : Firefox 3.6 avec les polices STIX

Alfred Rényi
Sur un théorème général de probabilité
Annales de l'institut Fourier, 1 (1949), p. 43-52
Article PDF | Analyses MR 14,886d | Zbl 0036.08703

Résumé - Abstract

L'auteur généralise un théorème qu'il a déjà donné (J. de Math. 28 (949)). Envisageant un champ de probabilités au sens de Kolmogoroff, il élargit puis étudie la notion de discrépance, en introduisant la discrépance $D_y(x)$ d'une variable aléatoire x par rapport à une autre variable aléatoire y ; elle se réduit au coefficient de corrélation si x et y sont des variables caractéristiques. Il introduit aussi la notion de suite de variables aléatoires ``presque indépendantes deux à deux'', avec un coefficient $\Delta $ dit module de dépendance. Il donne alors essentiellement pour une telle suite x_n l'inégalité

```
$$\sum ^\infty _1 D^2_y(x_n)\le (1+\Delta ) \big [1+\big ({e(y)\over \sigma (y)}\big )^2\big ]$$

où $e(y)$ est valeur probable, $\sigma $, écart moyen.
```

(c) TeX

Figure 4: Trois présentations d'un même article…

symboles inventés ou détournés, des fontes spéciales, des figures et des diagrammes. Un premier pas serait de disposer d'outils de conversion vers SVG pour un certain nombre d'extensions à caractère graphique de LaTeX, comme XY-pic ou PGF. Comme c'est a priori tout à fait faisable dans Tralics, il ne reste plus qu'à susciter des bonnes volontés !

À l'heure actuelle, la quantité de travail nécessaire pour produire une version XHTML à partir d'un article sous sa forme finale en LaTeX reste dissuasive : le nombre de constructions bizarres employées par les auteurs, de code fossile copié de sources TeX antédiluviens, d'extensions ou de macros non supportées — voire dont la traduction en XHTML+MathML est vraiment problématique — est trop important.

Bibliographie

1. T. Bouche, « A pdfLaTeX-based automated journal production system », *TUGboat* **27** (2006), nº 1, p. 45-50.

2. J. Grimm, « Tralics, a LaTeX to XML Translator », *TUGboat* **24** (2003), nº 3, See `http://www-sop.inria.fr/apics/tralics/`.

3. P. Jacquier-Roux, « RUCHE, an editorial flow management tool », `http://ruchedemo.cedram.org/`, 2006.

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