



**HAL**  
open science

## PolyTeXML Du source XML au document formaté

Stephane Mottelet, Michel Vayssade

► **To cite this version:**

Stephane Mottelet, Michel Vayssade. PolyTeXML Du source XML au document formaté. Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie, Nov 2002, Villeurbanne, France. pp.143-147. edutice-00000651

**HAL Id: edutice-00000651**

**<https://edutice.hal.science/edutice-00000651>**

Submitted on 7 Oct 2004

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# PolyTeXML

## Du source XML au document formaté

Stephane Mottelet<sup>1</sup>, Michel Vayssade

Université de Technologie de Compiègne  
Centre de Recherches de Royallieu, Rue Personne de Roberval, BP 20 529  
60205 Compiègne CEDEX, France

### Résumé.

PolyTeXML est une chaîne éditoriale permettant de produire des cours sur des supports électroniques (fichiers au format PDF pour consultation à l'écran) ou classiques (polycopié de cours sur support papier). Les fichiers sources sont au format XML (eXtended Markup Language) et respectent une DTD (Document Type Definition). Cette chaîne est l'évolution d'une autre chaîne éditoriale : PolyTeX. La chaîne éditoriale PolyTeXML se décompose en : un éditeur adapté pour le format XML (epcedit) et un ensemble de feuilles de style XSL (eXtended Stylesheet Language) permettant de transformer les sources XML en fichiers au format PolyTeX. Les différents choix techniques permettent d'envisager un partage de contenus pédagogiques entre plusieurs personnes au sein d'un groupe de travail, ou bien entre plusieurs sites distants.

**Mots-clés.** XML, XSL, PolyTeX, Chaîne éditoriale, Documents multisupport.

**Keywords.** XML, XSL, PolyTeX, Editorial flow, Multimedia document.

## 1 Présentation rapide de PolyTeXML

### 1.1 Historique du projet

La chaîne éditoriale PolyTeXML est l'aboutissement du projet PolyTeX<sup>2</sup>. Polytex, projet initié par B. Bachimont [1], est un ensemble de macros et de classes LaTeX permettant de construire des polycopiés de cours structurés. Les polycopiés sont structurés en trois niveaux : les chapitres, les sections, les grains. Un grain Polytex est typiquement une partie de texte de 1 à 3 pages, éventuellement accompagné d'exercices, de documents et de transparents destinés à la projection (vidéo ou classique). Un même document peut être :

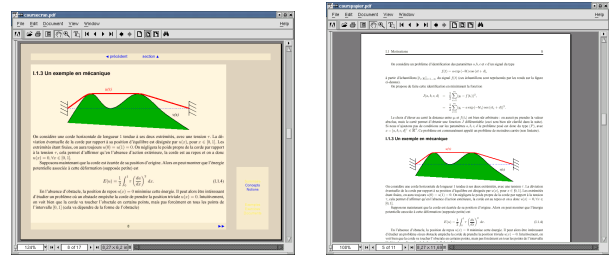


FIG. 1.1 – Un document polytex en version papier ou écran.

1. visualisé sur un écran (image de gauche sur la figure 1.1). La police de caractère utilisée est plus grosse, des liens hypertextes sont présents et le texte est découpé en plusieurs pages écrans accessibles via les liens hypertexte.
2. imprimé (image de droite sur la figure 1.1). La police de caractère est de taille standard, les liens hypertextes disparaissent pour laisser la place à des références.

De plus, PolyTeX permet maintenant de développer assez facilement de nouveaux "thèmes", permettant par exemple d'adapter un document existant à différentes chartes graphiques.

Afin de faciliter le rôle des concepteurs de grain (tout le monde n'est pas censé connaître le langage LaTeX, et de les empêcher d'insérer des commandes de mise en forme dans un grain (ce qui est tout de même possible avec Polytex), nous avons décidé d'ouvrir PolyTeX au monde XML. Un des premiers avantages de cette démarche est de permettre l'utilisation d'un éditeur "intelligent" permettant de valider **a priori** la structure des documents. De plus, le choix des technologies XML nous permet d'envisager plus facilement le partage des contenus pédagogiques.

### 1.2 Principes généraux

PolyTeXML est une chaîne éditoriale permettant de produire des cours matérialisés sur des supports électroniques (écran) ou physiques (papier). La chaîne se constitue de :

<sup>1</sup>e-mail : Stephane.Mottelet@utc.fr, tél : (33) 344234688 , fax : (33) 344234477

<sup>2</sup>http ://www.dma.utc.fr/polytex/

- un **éditeur**, qui peut être quelconque, permettant d’écrire un texte ASCII, ou bien un éditeur XML, permettant d’écrire un document source conforme à la DTD (Définition du type de Document) de PolyTeXML.

L’éditeur de texte n’est pas un traitement de texte : le texte édité n’est pas formaté ; en revanche, il est structuré en utilisant des balises XML, éventuellement paramétrées par des attributs. Le marquage du texte par les balises XML correspond à la structuration du texte.

- un **fichier de commandes**, exécutant la transformation du source XML en un fichier au format LaTeX, utilisant les environnements et commandes du style PolyTeX. Cette conversion est faite via des transformations du source XML, paramétrées par des feuilles de style spécifiant les règles de transformations. Le fichier de commande accepte plusieurs options selon que l’on désire une version imprimée ou électronique (pour consultation à l’écran), un polycopié ou des transparents.

Ce fichier de commandes effectue aussi la compilation de ce fichier au format DVI (DeVice Independant) ou PDF (Portable Document Format), en poursuivant la chaîne de compilation standard de polyTeX.

- un **outil de visualisation**. Le fichier PDF peut être lu grâce à AcrobatReader en exploitant les fonctionnalités de navigation du document. Cependant, durant la phase de mise au point on préférera visualiser le fichier DVI. Dans le cas d’une lecture sur le papier, il suffit d’imprimer le texte construit à travers l’option **paper** à partir d’AcrobatReader, ou bien d’imprimer un fichier **PostScript** produit à partir du fichier DVI.

Le principe essentiel est que la chaîne éditoriale PolyTeXML contient et s’appuie sur la chaîne éditoriale polyTeX (qui a été améliorée et étendue). Il est toujours possible d’utiliser polyTeX directement, mais l’ajout de la couche de conversion à partir d’un fichier au format XML permet de valider **a priori** la structure du document, grâce à un éditeur XML utilisant la DTD.

La distribution de PolyTeXML intègre un utilitaire permettant de convertir les anciens fichiers au format polyTeX en fichiers XML conformes à la DTD de PolyTeXML. Cela permet une transition en douceur pour les utilisateurs actuels de polyTeX.

### 1.3 La DTD de PolyTeXML

La DTD de PolyTeXML est principalement la traduction de la structure implicite des documents PolyTeX,

Cette DTD est un document de 480 lignes environ (8 pages), destinée à être fournie à un éditeur XML permettant de faire de la validation **a priori**, ou bien servant à la validation **a posteriori** avant la transformation vers d’autres

formats. On peut toutefois produire presque tous les documents sans avoir besoin de lire ce fichier. La DTD va donner des indications telles que :

- un document commence et finit par les balises `<polytexml>` et `</polytexml>`
- dans une balise `<grain>` il y a obligatoirement une balise `<title>`
- etc ...

Par exemple voici un modèle minimal de document PolyTeXML, contenant une collection non organisée de grains.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE polytexml SYSTEM
  "/usr/local/PolyTeXML/dtd/polytexml.dtd">
<polytexml>
  <grain>
    <title>Un essai de grain</title>
    <concept>grain</concept>
    <sub-concept>essai</sub-concept>
    <paragraph>
      Ceci est un paragraphe dans le corps
      du grain. On peut par exemple écrire
      des mathématiques au format LaTeX.
      <equation label="eq1">
        y=\frac{x+y}{2}
      </equation>
      et utiliser d'autres types de balises :
      <itemize>
        <item>
          <paragraph>
            des balises de structuration
            dérivées des commandes et
            environnements de polyTeX.
          </paragraph>
        </item>
        <item>
          <paragraph>
            des balises dérivées des
            commandes et
            environnements de LaTeX.
          </paragraph>
        </item>
      </itemize>
    </paragraph>
    ...
  </grain>
  <grain>
    ...
  </grain>
</polytexml>
```

Le document ainsi obtenu est valide et permet de produire deux grains autonomes. Il s’agit d’une première démarche que l’on peut faire lorsque le document final n’est pas encore bien défini. La figure 1.2 montre la première page du document obtenu après compilation.

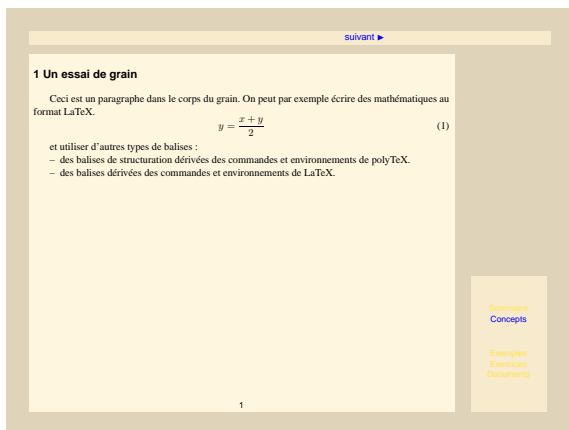


FIG. 1.2 – Allure du premier grain en mode écran

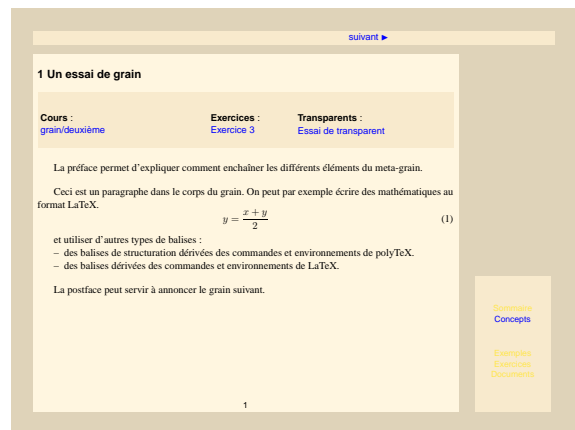


FIG. 1.3 – Allure d'un meta-grain en mode écran

Ce premier exemple de document montre que la DTD de PolyTeXML autorise la saisie de grains isolés. La chaîne de compilation permet aussi de voir rapidement à quoi ressembleront les grains une fois compilés. Nous allons voir maintenant comment un grain peut être associé à d'autres types d'éléments.

## 1.4 Le meta-grain : un élément de scénarisation

On peut associer à un grain, qui contient typiquement l'exposé d'un concept du cours, des exemples, des exercices, des documents, ainsi que des transparents et des références. Cela permet de faire une scénarisation élémentaire. Voici un exemple de meta-grain :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE polytexml SYSTEM
"/usr/local/PolyTeXML/dtd/polytexml.dtd">
<polytexml>
  <meta-grain label="autreMetaGrain">
    ...
  </meta-grain>
  <meta-grain label="exempleMetaGrain">
    <references>
      <refcourse label="autreMetaGrain"/>
    </references>
    <preface>
      La préface permet d'expliquer
      comment aborder les différents
      éléments du meta-grain.
    </preface>
    <grain>
      ...
    </grain>
    <slide>
      ...
    </slide>
    <exercice>
      ...
    </exercice>
    <postface>
      La postface peut servir à
      annoncer le grain suivant.
    </postface>
  </meta-grain>
</polytexml>
```

Sur la figure 1.3 le meta-grain compilé fait apparaître des références, qui sont de deux types. Il y a une référence implicite à l'exercice contenu dans le meta-grain, ainsi que la référence explicite provenant de la balise <references>. L'exercice n'apparaît pas dans le corps du grain. Son emplacement réel dans le document compilé dépendra des éléments de plus haut niveau contenant le meta-grain. Par exemple, pour le source suivant :

```
<polytexml>
  <chapter>
    <section>
      <meta-grain>
        <grain>
          ...
        </grain>
        <exercice>
          ...
        </exercice>
      </meta-grain>
    </section>
  </chapter>
</polytexml>
```

L'exercice apparaîtra dans une section ne contenant que des exercices, à la fin du chapitre.

## 2 PolyTeXML, un outil facilitant le partage des contenus

### 2.1 Partage de grains

Par rapport à PolyTeX, polyTeXML possède une nouvelle fonctionnalité qui est la possibilité d'inclure des grains qui ne sont pas physiquement présents sur la machine où en mise en oeuvre la chaîne de compilation. Voici un exemple de ce qu'il est possible de faire :

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE polytex SYSTEM
"/usr/local/PolyTeXXML/dtd/polytexml.dtd">
<polytexml>
  <meta-grain>
    <references>
      ...
    </references>
    <preface>
      ...
    </preface>
    <include-grain
      href="http://silo.utc.fr/math/grain31415.xml"/>
    <include-exercice
      href="http://silo.utc.fr/math/exo31415-1.xml"/>
    <slide>
      ...
    </slide>
  </meta-grain>
</polytexml>

```

La balise `<include-grain>` vient en lieu et place de la balise `<grain>`, au même titre que la balise `<include-exercice>` pour la balise `<exercice>`. L'attribut `href` précise la localisation du fichier à inclure. Ici, on a donné un exemple où les fichiers se trouvent sur un serveur http, mais on aurait pu aussi utiliser des références purement locales, du type :

```

<include-grain
  href="/home/mottelet/math/grain31415.xml"/>

```

Dans un cas, on isole chaque grain, exercice, document, etc. pour y voir plus clair dans le document principal, dans l'autre, on accède à des ressources pédagogiques éventuellement partagées entre plusieurs établissements distants, ce qui décuple le champ des utilisations possibles.

Dans ce type d'utilisation, les balises `<preface>` et `<postface>` permettent de bien définir où doivent être faits les commentaires spécifiques aux contextes d'utilisation des grains.

**Remarque 2.1** La possibilité de gérer de façon complètement transparente des URL locaux ou de type WWW a été un des critères de choix du processeur XSL `xsltproc` du projet *Gnome*.

### 3 Les aspects techniques de la chaîne éditoriale

Nous allons maintenant préciser quels sont les choix techniques retenus, tout en sachant que ces choix pourront être facilement remis en cause par la suite. On verra par exemple que le choix d'un éditeur XML n'est pas simple, suivant les critères retenus : open source ou payant, adapté pour la saisie d'un grain ou bien d'un document complet, voire les deux, etc.

Pour l'instant, la chaîne éditoriale actuelle a été testée sous Linux, mais elle a priori portable sur tout système d'exploitation supportant une distribution `TeX`, une distribution `PolyTeX`, et l'éditeur `epcedit`.

### 3.1 Les éditeurs XML

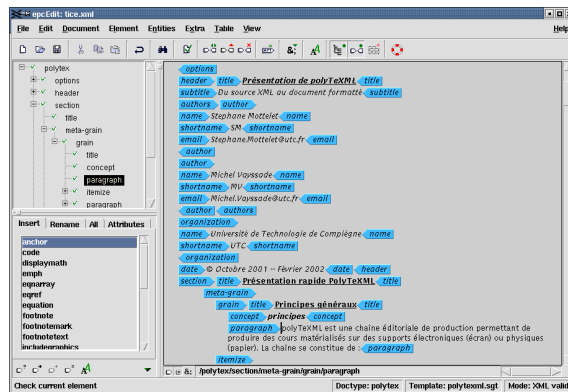


FIG. 3.4 – L'éditeur `epcedit` en action

L'éditeur retenu pour l'instant est l'éditeur `epcedit`<sup>3</sup>. Cet éditeur, écrit en Tcl/Tk, fonctionne sur toutes les plateformes, et présente de nombreuses possibilités de personnalisation, en donnant l'accès à la structure interne du document XML. D'autre part il est assez convivial et permet d'appliquer des styles typographiques pour chaque type de balise. Il permet aussi de naviguer facilement dans la structure arborescente du document (voir sur la figure 3.4 une copie d'écran).

L'intérêt principal de ce type d'éditeur est de proposer une saisie contextuelle du document : à un endroit donné du document il n'est possible d'insérer que ce qui est autorisé par la DTD.

Les possibilités de personnalisation, via des scripts écrits en Tcl/Tk accessibles via un menu, permettent par exemple de lancer la compilation et la visualisation du document sans sortir de l'éditeur.

### 3.2 Les transformations XSL

Pour passer des documents sources (du texte et des balises, dans un ordre et un arrangement conforme à la DTD) au document formaté, on va faire subir au source une série de transformations qui vont conduire à un document "source.tex" pouvant servir d'entrée à "latex" ou "pdflatex".

Pour des raisons liées au fonctionnement des processeurs XSL, cette transformation fichier source XML vers fichier source LaTeX est réalisée en plusieurs étapes enchaînées, avec génération d'un fichier intermédiaire à chaque fois. La réalisation (et la maintenance !) du code XSL en est facilitée.

N'importe quel processeur XSL peut être utilisé : toute la logique de transformation est incluse dans les fichiers

<sup>3</sup><http://www.epcedit.com>

.XSL de la distribution polytexml. Nous utilisons le processeur "xsltproc" du projet GNOME car celui-ci écrit en C et compilé nativement est extrêmement rapide non seulement à l'exécution mais aussi au chargement en mémoire. Ainsi on peut l'invoquer 5 ou 6 fois de suite sans délai gênant en interactif. De plus, il gère correctement de façon transparente les inclusions XML et XSL.

```
xsltproc is a command line tool for applying XSLT
stylesheets to XML documents. It is part of libxslt, the
XSLT C library for GNOME.
```

Project libxslt home page: <http://xmlsoft.org/XSLT/>

### 3.3 le problème des formules mathématiques

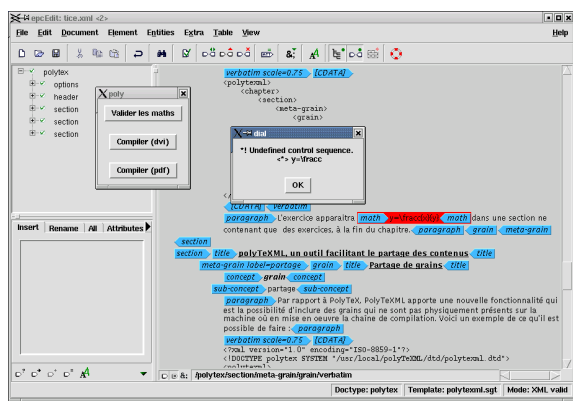


FIG. 3.5 – Validation des formules mathématiques avec epcedit

Il s'agit d'un problème épineux. Bien qu'il soit déjà possible de convertir des formules mathématiques du format LaTeX au format MathML, il est très fastidieux d'écrire ces formules directement "à la main". Il existe des "plugin" pour certains éditeurs XML permettant d'écrire les formules à l'aide d'un éditeur d'équation similaire à celui de Microsoft Word, mais cette façon de procéder n'est admissible que lorsqu'il y a peu de formules à saisir dans un document.

C'est pourquoi, en l'attente d'une solution, nous avons choisi de conserver le format LaTeX pour les formules mathématiques. Ce n'est pas sans poser de nombreux problèmes, en particulier celui de la validation du document. Nous avons trouvé une solution qui nécessite l'utilisation de l'éditeur epcedit. Il s'agit d'un script écrit en Tcl/Tk (le langage dans lequel epcedit a été écrit), qui réalise une compilation séparée des environnements mathématiques, et qui signale les portions de code LaTeX éventuellement incorrectes (voir sur la figure 3.5).

## 4 Conclusion

La chaîne éditoriale PolyTeXML est l'évolution logique de PolyTeX. En ce qui concerne les documents déjà écrits au format PolyTeX, leur conversion ne pose aucun problème particulier, grâce à leur structuration imposée par ce format (un convertisseur automatique a été développé). L'utilisation d'un éditeur "intelligent" devrait permettre de toucher un plus grand nombre d'utilisateurs, grâce à la convivialité indéniable de ce type d'outil.

Les choix techniques (XML, outils "open source", environnement Unix) et le choix de la charte de structuration permettent d'envisager d'utiliser PolyTeXML comme outil "client" d'un broker éditorial (base de données distribuée de contenus pédagogiques).

PolyTeXML est en phase d'évaluation au sein de l'Equipe de Mathématiques Appliquées de l'Université de Technologie de Compiègne, à l'INSA de Lyon et à l'INSA de Rouen (projet INFOREP).

## Bibliographie

- [1] B. Bachimont and J. Charlet. Polytex : un environnement pour l'édition structurée de photocopiés électroniques multisupports. In *EuroTeX'98*, pages 1–16, Saint Malo, France, 1998.