

XML を用いた技術文書の電子出版

3Y-03

塚本享治、花園徹太郎*、菊地修一、畑朝章、鎌田悠佑 (東京工科大学)

はじめに

XML は文書だけでなく、画像、映像、アニメ、音楽など多様なメディアの共通的な記述形式になりうる可能性を秘めている。そこで、学部内で必要となる多様なメディア関連文書の電子出版/アーカイブシステム化を手始めに XML 活用分野の開拓を始めた。この報告では、これまでに実現したシステムと、学会誌記事を題材に行った評価について述べる。

システム概要

システムの設計にあたり、XML 形式で記述されたコンテンツや変換後のコンテンツをネットワーク経由で共有、管理、公開することを前提とした。システムは Apache Jakarta Tomcat をベースに図 1 のように実現することにした。

利用者には、WWW ブラウザ以外の特殊なソフトを用いなくても良いように、サーバ側で XML ファイルのエディット機能までを含むものとした。サーバ側の Tomcat で、DB アクセス、及び、XSLT を使用した、変換処理を行う。

ユーザーはあらかじめ準備した DTD に従って XML ファイルを作成する。DTD は用途別に複数準備しておく。必要ならば、数式を記述する MathML 形式の文書ファイル、Jpeg 等の画像フ

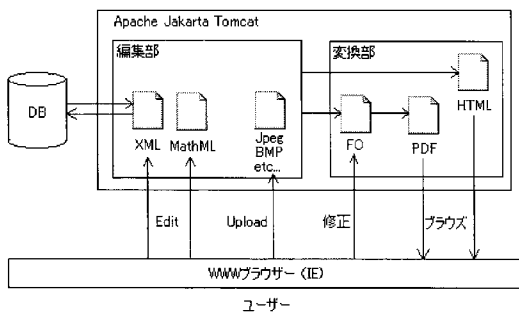


図1・システム概要

ァイルをアップロードする。そのうち Tomcat が HTML・PDF 形式の二種類のファイルに変換する。

作成した文書は DB に蓄積され、誰でも検索出来る。作者であるならば、修正をすることも可能である。

変換プロセス

XML ファイルは、いったん JepaX 形式のファイルに変換し、JepaX 形式を HTML ファイルや PDF ファイルに変換する。JepaX は入れ子構造が多く記述が面倒ではあるが、標準的な規格なので、他のソフトとの互換性があるからである。XML ファイルが用途別に存在するよう、JepaX 変換スタイルシートも複数ある。

この JepaX ファイルを中心とした変換プロセスを図にしたものが図 2 に掲げる。

- ① MathML ファイルを、MML2SVG.xsl によって SVG ファイルに変換する。
- ② XML ファイルを、XML2JepaX.xsl で JepaX ファイルに変換する。
- ③ XML ファイルを、XML2HTML.xsl で HTML ファイルに変換する。この HTML ファイルはブラウザによって検証できる。
- ④ JepaX ファイルを、JepaX2FO.xsl で FO ファイルに変換する。
- ⑤ FO ファイルは、リンクで参照された画像ファイルや SVG ファイルと一緒に、

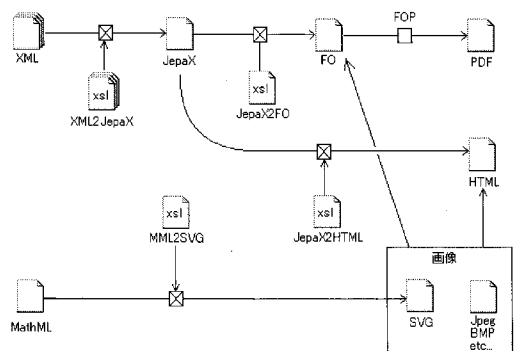


図2・変換プロセス

Technical document publishing based on XML technology.
 Michiharu Tsukamoto, Tetsutaro Hanazono,
 Syuichi Kikuchi, Tomoaki Hata, Yusuke Kamata
 Tokyo University of Technology

Apache.org のアプリケーション FOP によって、PDF ファイルに変換する。この PDF ファイルもブラウザによって検証できる。

- ⑥ 図表や数式を適切な位置にレイアウトする作業を自動化するのは難しい。そこで、当面は PDF ファイルをブラウザして、FO ファイルを編集することによって解決することとした。XML ファイルを修正するのはコンテンツにレイアウト情報が埋め込まれてコンテンツとしての意義が半減するからである。

変換スクリプト

変換の中心となる部分は下記の二つである。これらは XSLT スタイルシートであるが、変換ルールを再帰的に適用するだけにとどまらず、DOM ツリーや外部情報を用いた計算処理を行っている。そこで以下ではスクリプトと呼ぶ。

(1) JepaX → FO 変換スクリプト

これは、JepaX ファイルを FO ファイルに変換するためのスクリプトである。HTML で表現できるようなタグは、XSLT と XPath 関数を使ってほぼすべてを実装した。具体的には、

- 複数のコラム
- 画像の貼り付け
- リスト (順序付きリスト・定義リスト・二段リスト)
- テーブル

(2) MathML → SVG 変換スクリプト

これは、MathML ファイルを SVG ファイルに変換するものである。第 1 版では XSLT のみで変換を試みた。XSLT は変数の更新ができない関数型言語であり、戻り値に制約があるために、特殊なプログラミングを行うこととなった。しかし、加減乗除・平方根・分数・積分・行列まで、変換することが可能となった。

美しい数式を出力するにはフォントメトリックスを利用することが必要となる。そこで、第 2 版では BSF(Beans Scripting Framework)を使って外部 Java プログラムを呼び出し、Java プログラムで DOM ツリーの操作や外部情報を使った計算を行い、XSLT 側ではその戻り値を使って SVG ファイルを作り出す方式のものを現在開発中である。

評価実験

情報処理学会インタラクティブエッセイ Vol.39, No.4(1998.4)を使って評価実験を行った。第 1 版のスタイルシートはすべて XSLT で記述し、XML2HTML.xml が約 290 行、XML2JepaX.xml が約 140 行、JepaX2FO.xml が約 390 行、MML2SVG.xml が約 200 行であった。その変換処理時間は、約 1 秒で終了したが、FOP で 8 ページの PDF ファイルに変換するのに 10 秒を要した。なお、テスト環境は、Pentium3、1GHz、256MB、Windows2000 であった。

残された課題

課題は 2 種類に分けられる。

(1) 利用する規格・パッケージの問題点

XSL を実装した現状の ApacheFOP では、複数著者の記事をページ内での連結、コラム数の変更などが不可能である。また、縦書きなども未実装である。

しかし横書きの単一記事ならば、図表、数式もふくめて、まずまず納得いくものが出力できた。

(2) システム開発の問題点

開発中のシステムは汎用的なレイアウトを選択・処理という文書作成に留まっており、以下のような課題が残されている。

- 細かいレベルでレイアウトを変更することの出来る DTD
- 継承できるような XSLT モジュール化
- データベースアクセスの簡略化
- DTD 対応の XML エディタの作成

XSLT は強力な変換言語であるが多様なメディアを扱うにはそれだけでは不十分であり、他言語で補強することが必要なことが明らかとなった。

参考資料

- 1)XSL1.0: <http://www.w3.org/TR/xsl>
- 2)XSLT1.0: <http://www.infoteria.com/jp/REC-xslt-19991116-jpn.htm>
- 3)XPath1.0: <http://www.infoteria.com/jp/REC-xpath-19991116-jpn.htm>
- 4)SVG1.0: <http://www.w3.org/TR/SVG>
- 5)MathML2.0: <http://www.w3.org/TR/MathML2/>
- 6)JepaX0.9: <http://x.jepa.or.jp/jepax/>