

XSL の光と影

小林 徳滋
アンテナハウス株式会社
代表取締役

W3C が 2001 年 10 月に勧告した Extensible Stylesheet Language に準拠する組版エンジン XSL Formatter について報告する。XSL FO は、ページメディアを対象に XML コンテンツとレイアウト用スタイルが完全分離できる。一方、スタイルシートの開発がなかなか大変である。仕様も実装も技術的に成熟していないので、継続して改良する努力が必要である。

The brightness and darkness of XSL

Tokushige Kobayashi
President
Antenna House, Inc.

This article reports on XSL Formatter, that conforms to W3C XSL Specification. XSL is not mature technology, future enhancement is expected.

XSL FO の概要

W3C が開発した「Extensible Stylesheet Language」(以下、XSL FO) Version1.0 仕様は、XML をページの概念のあるメディアにレイアウトするためのものである。レイアウトの対象となる Formatting Object、オブジェクトのフォント名などの属性、位置、大きさなどのプロパティを定義している。

開発には非常に長い時間がかかっている。XSL FO 仕様は 1998 年に最初のドラフトが公開された。その後、数回のアップデートを経て、2000 年 11 月 Candidate Recommendation (CR) になった⁽¹⁾。

W3C は、2001 年 2 月から 7 月まで CR を実装した XSL FO 組版エンジンの開発企業・団体の協力により適合性の試験を行い、XSL FO のテストスイートも作成されている⁽²⁾。

こうした実装による評価を経て、**XSL FO 1.0 仕様は 2001 年 10 月 15 日に W3C 勧告となった。**

2000 年から勧告の成立までの動き

⁽¹⁾CR は仕様の実装を推奨する段階である。W3C の勧告は単に仕様を決めるのではなく、実際にそれを実装したソフトウェアがあることが前提になるので CR 段階での実装が重要とされている。

⁽²⁾残念ながら現時点ではテストスイートは公開されていない。

2000年3月27日 Working Draft
2000年10月18日 Working Draft
2000年11月21日 Candidate Recommendation
2001年8月28日 Proposed Recommendation
2001年10月15日 Recommendation

XML インスタンスには、通常ドキュメント構造とコンテンツのみが含まれ、レイアウト指定はなされない。これに対して、レイアウトを指定した XML インスタンスが **Formatting Object** ツリーである。元の XML インスタンスから **Formatting Object** ツリーへの変換には、XSLT プロセサを使うことが多い。もちろん、XSLT プロセサではなく、別の方法で **Formatting Object** ツリーへの変換を行っても良い。

プロパティは、ショートハンドで指定したり、単位の設定は例えばフォントの大きさであれば、ポイント、em、パーセント値などが可能で値の指定方法がさまざまにある。そこで **Formatting Object** ツリーは、リファインメントという処理で各種のプロパティ値の調整を行う。リファインメントの中には、Unicode の **BIDI** アルゴリズムの適用なども含まれる。

レイアウトのモデルはエリアモデルである。エリアモデルは **CSS** を引き継いでいる。各エリアはページの中に位置をもち、そのエリアに何を表示するべきか、背景、パディング、境界をもつ。

XSL FO プロセサ (XSL FO 組版エンジン)

Formatting Object ツリーを組版するソフトウェアが **XSL FO** プロセサである。**XSL FO** プロセサは、(1) **FO** ツリーをリファインメントし、(2) 指定サイズのページに組版してエリアツリーに変換、(3) そして、エリアツリーを出力媒体にレンダリングする。

市販の **XSL FO** プロセサには、**XSL FormaterV2** と **RenderX** の **XEP** がある。オープンソースの運動を行っている **Apache** の **XML** プロジェクトの **FOP** も有名である。**FOP** は無償で公開されているので、普及度が高く、**XSL FO** の普及という面では貢献が大きい。

名前	特徴
Antenna House XSL FormatterV2.2	組版エンジンは Windows の ActiveX コントロール。出力媒体は標準で WindowsGDI だが、 V2 より PDF 出力オプションを留意。 COM コンポーネントとして各種のシステムに簡単に組み込み可能。 CJK に完全対応しているのが特徴。現在、 Linux/Unix 版を開発中。

名前	特徴
FOP	Apache Software Foundation が開発するオープンソースの Java 版の XSL FO 組版エンジン。組版結果は PDF に出力する。日本語フォントも使用できるようになってきている。 Version0.20.3 で W3C 勧告に対応したが、実装のレベルが低い。XML2001 で聞いた話では米国の専門家の間では使えないという評価になっているようだ。現在、レイアウト機能の再設計を行っている。PDF にはフォントのアウトラインを埋め込めないで、パブリッシングという観点では実用にならない。
XEP	ロシア（アルメニア）のベンチャー企業 Render-X の Java 版 XSL FO 組版エンジン。FOP 同様に PDF を出力する。CJK も組版できるが、綺麗ではない。
XFC (XSL Formatting Objects Composer)	米国 IBM の alphaWorks で公開されている試作版 FO プロセッサ。Java 版。発売するかどうかは不明。
UFO	Unicorn 社の XSL FO を TeX のマクロに変換する開発ツール。従って組版は TeX を使用する。製品化されているかどうかは不明。
Passive TeX	XSL FO を処理する TeX のマクロライブラリ。ボランティア・ベースのプログラムで製品化されていない。
Epic Editor	Arbor Text 社の Epic Editor は、XSL FO をサポートしているようだ。但し、Epic Editor の最新版 V4.2 は、2001 年 9 月の発売で、XSL FO 仕様が確定する前なので、サポートしている XSL FO の仕様は Candidate Recommendation 段階のものと思われる（詳細は不明）。

世界の潮流は JAVA による実装に向かっているようだ。Web サービス分野では、JAVA+FOP が非常に普及している。XSL Formatter は公開されている中では唯一の C++による実装である。

XSL Formatter も FOP も組版結果はエリアツリーとなる。エリアツリーをレンダリングして可視化する。XSL Formatter は、現在、WindowsGDI 用のレンダラー、PDF 出力用のレンダラーを用意している。WindowsGDI に出力した場合は、ディスプレイのみでなく、プリンタ・ドライバに出力することができる。

将来はレンダラーを増やすことで出力形式・媒体を拡張していく予定である。現在、エリアツリーを単独で流通させることができないが、将来は、エリアツリーを単独で流通させることができることも考えられる。エリアツリーは XML インスタンスなので PDF よりは取り扱いが容易な筈である。

XSL FO プロセサの市場

XSL FO プロセサの市場は、現在利用例が出てきた段階であって、まだ確立していない。一番多い要望は、XML の表示と印刷であろう。XML を表示する方法として XML から HTML に変換して、HTML をブラウザで表示するという方法が既に普及している。HTML をブラウザで表示すると印刷したときにレイアウトが崩れたり、オブジェクトの印刷位置を指定できないという問題がある。これに代わって XSL FO を使ってレイアウトを指定したいというものである。簡単な帳票、レポートや手紙類の印刷を行うため HTML の代わりに XSL FO を使おうというもの。XSL FO プロセサから紙に出力したり PDF 化する。簡易印刷用途である。

これに対して、XSL FO プロセサで XML のメリットを引き出すには Unicode を使って CJK (中国語、日本語、韓国語)、HAT (タイ語、アラビア語、ヘブライ語) を含む多言語処理を行ったり、例えば航空機のマニュアルなどのような膨大な量のドキュメントの組版などの用途を開拓していく必要があるだろう。またサーバサイドでの完全な自動組版も重要である。

XSL Formatter の利用法には、次の 3 つがある。

- ・ スタンドアロン・アプリケーションとして使用する。エディタで作成した XML ドキュメントにスタイルシートを適用して XSL FO に変換し、組版して出力する用途である。ページ数の大きいテクニカル・マニュアルを組版している例や出荷指示書や注文書をプリントしている例がある。
- ・ サーバ上で XML 自動組版エンジンとして使用する。サーバ上で、データベースから取り出した XML データを組版したり、サーバ上で技術マニュアルを PDF に変換して、各地のショールームに配信する例がある。
- ・ 組込用 XML 表示・PDF 化エンジンとして使われるケースも増えている。例えば、Sweden の Sormans 社では、After Sales Information Systems に組み込んで、XML を PDF 化するのに使っている。米国では医療サービス会社のネットワークに組み込む例も出てきている。

PDF 化

出力形式は PDF が中心になりつつある。出力形式の PDF についていうと、帳票の PDF では高速処理などが必要なのに対し、書籍やマニュアルなどの PDF 化では、フォントの埋め込みを初め、多種類の画像形式への対応、画像の品質の高さ、画像サイズの圧縮処理などより豊富な機能が要求される。

特にパブリッシングのレベルになると高機能が要求されるので、XSL Formatter では、現在のところ Acrobat Distiller に依存せざるを得ない。高度な PDF 処理は相当に敷居が高いが、今後の開発課題としている。

また、タイ語、アラビア語、ヘブライ語などを正しく PDF 化するのは大変に難しい。例えば、これらの言語ではキャラクターのシーケンスによってグリフが変化する。WindowsAPI を使うとタイ語、アラビア語、ヘブライ語のグリフ置換は自動的にやってくれるので画面上はそれなりに表示される。しかし、PDF を自力で作る場合は、WindowsAPI が行っているシーケンスによるグリフ置換を自力で行う必要がある。Windows は、多言語処理機能が強化されているので、Windows の API を使えば簡単にできることも自力で実装していく必要がある。しかし、組版エンジンの多言語化は避けて通れない課題であろう。製造業は既に全世界で製品を販売しているので、多言語マニュアルの作成だけでも相当な需要があるはずだ。

XSL FO の光の部分

コンテンツとスタイルを完全分離することが可能になったことで、ワンソースのマルチレイアウトが簡単にできるようになった。例えば、ドキュメントの自動組版として普及している TeX は、コンテンツとレイアウトが完全分離ではない。

XML を組版できる組版ソフトは増えているが、その多くはインポートと組版処理が二段階になる。すなわち XML をインポートして手動で再組版を行うという方法である。従って、純粋な意味での自動組版ではない。これに対して、XSL FO の仕様は、自動組版を強く意識したものとなっている。プログラムの動作としても、XSLT プロセサと XSL FO プロセサを連続して動かすことで、XML から PDF までを一貫処理できるので、サーバ等で自動組版するのに適切である。

ドキュメントの XML 化というのは、膨大な量、繰り返し使うというニーズがないと意味が薄い。膨大な量、繰り返し使うということは自動処理と不可分である。この面からも、XSL FO は自動処理に用途を見出すべきであろう。但し、XSL Formatter も FOP も数万ページの組版に使うには無理があるためアーキテクチャの見直しが必要だ。第一世代の XSL FO プロセサは、どれもまだ機能的に成熟していないと考えている。

Unicode を使っているので世界標準として市場開拓を行うことができる。多言語処理が可能。Unicode BIDI との関係も整理されていてアラビア語やヘブライ語の処理が簡単にできる。

XSL FO の影

XSL FO は長時間かけて開発した割には仕様の普及度は今ひとつのように思う。注目度、関心度が低いように感じる。その理由としては、Web、HTML の普及、WYSIWYG のワープロに慣れたユーザ、印刷界は DTP 全盛の時代、という時代背景の中で、バッチ志向、自動組版志向の XSL FO が本流にはなりずらいという面がある。

XSLT スタイルシートを作成するのが素人には難しい。特殊なアルゴリズムなので GUI ツールを使って XSLT スタイルシートを自動生成することも難しい。

コンテンツとレイアウトの完全分離が難しい部分もある。例えば、一番分かりやすいのは表の幅。そもそも、コンテンツとレイアウトの完全分離は可能なのか？これは HTML でも同じ問題がある。

FOP という粗悪品のフリー・オープンソースが存在していて、要求水準の低いレベルの需要を FOP が吸収している。このため市販パッケージソフトにお金が流れなくなっている。これは、FOP に限らず、一般的なオープンソース運動の問題点であろう。パッケージソフトを市販するという意味合いでのソフトウェア産業が崩壊する危険がある。受益者負担の原則が崩れている。受益者が製品に対価を支払い、その対価をもって製品のさらなる改良に繋げるという産業の成長の原動力がなくなってしまっている訳だ。Acrobat Reader、IE などと同じだが、無償配布のソフトが普及することで、一種類のソフトウェアの独占になってしまい、競争が阻害され、ユーザの選択の余地がなくなり、産業の停滞に繋がるのではないかと考えている。

XML ドキュメントの自動組版という概念自体が新しく、理解している人が少ない。大部分の人は XSL FO を DTP の延長でしか捕らえていない。しかし、XSL FO は DTP の延長として期待するのは誤りである。このあたりの啓蒙を行わねばならない。

XSLT/XSL FO とも、まだ技術的に成熟していない。XSL FO の仕様はまだ十分とは言えない。CSS も CSS3 の開発の動きがあるので CSS3 を反映して拡張することになるのだろうが、既に一部の仕様はベンダが独自拡張しており、独自拡張部分はベンダ依存になっている。今後、どんどんベンダ依存の部分が増えていき、世界標準の良さがなくなる可能性がある。早期に仕様拡張が必要だ。

XSL FO の仕様は、米国主導で定められており、日本からは参加者がいない。日本語組版について FO の仕様に十分盛り込まれているとはいえないので拡張が必要だろう。日本語組版の要求をどうやって、XSL FO の今後の仕様に繁栄していくかは課題である。