

SVG を使ったすかしの例

ドキュメントを XML 化する目的は、Web ページを作成したり、電子的に配布するデータを作り出す、電子的なマニュアルを作成する、など、いろいろあります。しかし、どのような意図で XML ドキュメントを作成するにしても綺麗にレイアウトして「紙」に印刷することを欠かすことができません。

しかし、ドキュメントを SGML や XML で作成した時、それを綺麗に組版して印刷するのは技術的に難しい課題です。SGML の時代からドキュメントを SGML 化することで、ワン・ソース・マルチユースが実現できると言われてきましたが、現実には紙に印刷することさえ簡単には実現できませんでした。

SGML、XML ドキュメントを印刷する方法として、今までよく使われてきたのは次のような方法です。

- XML を Perl などのスクリプトで Microsoft Word の外部ファイル形式である RTF に変換する。RTF を Word などの RTF を読み込むアプリケーションで読み込んでレイアウトを調整して印刷する。レイアウト調整済み文書は RTF 形式になる。
- FrameMaker + SGML のスタイルを指定して、FrameMaker + SGML で読み込み、レイアウトを調整してから印刷する。この場合、レイアウト調整済み文書は FrameMaker の独自形式となる。
- 他の対話式組版ソフトや DTP ソフトの形式に変換し、組版ソフトや DTP ソフトでレイアウト調整して印刷する。レイアウト調整済み文書は、組版ソフトや DTP ソフトの形式となる。

しかし、これらの方法には問題があります。一番大きな問題は、XML コンテンツをアプリケーション独自の形式に変換してしまうことです。変換後のファイルに対してレイアウト調整を行うわけですが、レイアウト調整済み文書は元の XML コンテンツとは切り離されたアプリケーション独自のものになります。もし、ソースの XML コンテンツを変更したら、変換からレイアウトまでの操作をもう一度繰り返さなければなりません。

ドキュメントのソースを作ってから、レイアウト指定と調整を行い印刷するまでの流れは一サイクルで済むことはむしろ珍しく、普通は、校正などでソース変更とレイアウト調整のプロセスを何サイクルも通ります。従って、XML ソースを修正する都度、アプリケーションに読み込んでレイアウト調整が必要になるので、余計に手間がかかり、折角 XML でコンテンツを作った意味が薄くなってしまいます。そうかといって、レイアウト調整するアプリケーション上で文書の内容を修正したのでは、XML ソースと最終印刷物の内容がずれてしまいます。

こういったことを考えますと、コンテンツ XML 化の効果を上げるには **XML コンテンツ作成、レイアウト調整、組版・印刷まで一連のフローを確立する**のが理想的です。

Extensible Stylesheet Language 仕様は、Web の標準を定める団体である World Wide Web Consortium (W3C) で、「XML を綺麗にレイアウトして組版するための仕様」として、策定が進んでいるものです。この仕様の正式な略称は XSL ですが、現在、XSL という言葉は大変混乱していますので、本稿では XSL-FO と略記します⁽¹⁾。

現在、Internet Explorer 5 を使えば XML を表示したり印刷したりすることができるようになっています。しかし、これは、実際は内部で HTML に変換して表示しています。IE5 のようなブラウザはパソコンの画面で表示することを主たる目的として設計されています。このため、ブラウザではページレイアウトを指定するのが難しくなります。画面で表示してみるならブラウザが良いですが、「紙」への印刷では物足りません。

⁽¹⁾脚注文章。末尾の付録「XSL 仕様小史」を参照。