

XSL Formatter V2.3 のご紹介

2002 年 12 月



アンテナハウス株式会社

Table of Contents

XSL の仕様について	3
XSL を使った表示/印刷のステップ	3
XSL 組版エンジン	3
Antenna House XSL Formatter V2.3	4
世界の XSL-FO 組版エンジンの比較	4
XSL 仕様の意義	6
既存の方法は	6
既存ツールの問題点	6
主要ツールの問題	6
スタイルシートの相互運用性の問題	6
XSL のメリット	7
標準仕様である	7
コンテンツとスタイルの分離	7
自動組版に最適な仕様	7
XSL-FO とは	8
XSL-FO の機能説明	8
XSL Formatter の応用例	10
月刊情報誌のページ組版	10
概要	10
運用	10
XSL Formatter 選定の理由	10
After Sales Information System	11
概要	11
Sorman After Sales Information System (SAIS) について	12
XSL Formatter の選定理由	12
MSDS (Material Safty Data Sheet) 発行管理システム	12
概要	12
運用	13
XSL Formatter 選定の理由	13
XSL Formatter による自動組版	14
多言語組版について	15
XML と多言語組版	15
多言語組版の予備知識	15
文字と言語	15
Unicode	15
フォント	15
PDF 技術	17
XML と XSL 技術	17
XML	17
XSL	17
XSL Formatter と多言語組版	18
多言語の混在文書の組版	18
多言語組版の将来	19



XSL の仕様について

XSL (Extensible Stylesheet Language) 仕様は、Web の普及と標準化を進める W3C (World Wide Web Consortium) が、XML 文書を綺麗にレイアウトして組版するための仕様として開発したものです。XSL 仕様の開発は、1998 年から始まりましたが、途中で XSLT や XPath の仕様を切り離して別の仕様として勧告するなど、長期にわたる開発過程を経て **2001 年 10 月 15 日に XSL Version 1.0 が W3C 勧告となりました。**

XML には、普通、**文書内容のレイアウトの情報**がありません。これはコンピュータで自動処理するには良いのですが、人間が内容を理解するためにはあまり向いていません。普通の人間にとっては文章は、見出し、本文などにレイアウトした上で表示して欲しいものです。XML をレイアウトするための標準的な方法はスタイルシートを使うことです。方法は 3 通りあり、この間に関連性が大きく、CSS2 の仕様は XSL にも反映されています。

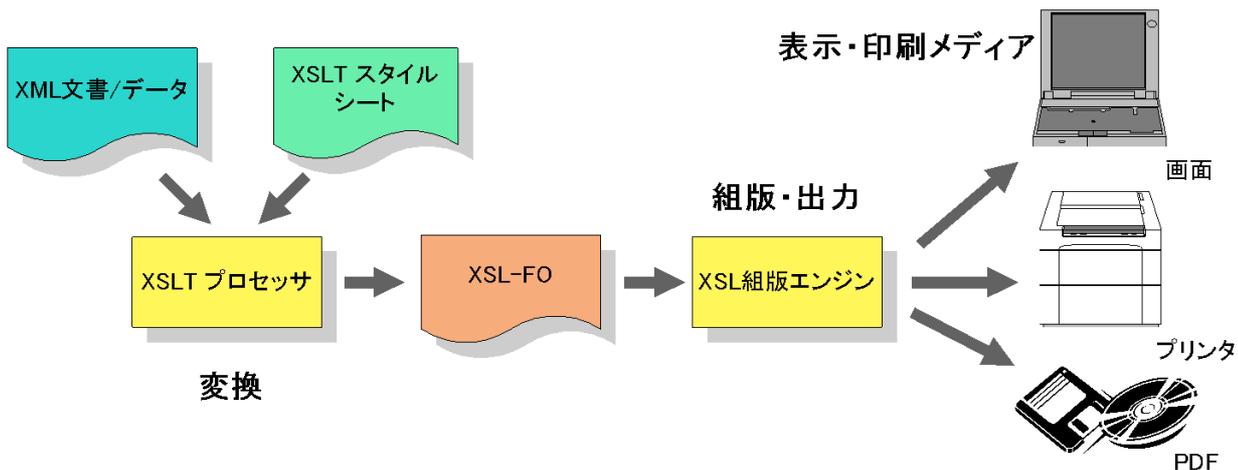
- CSS スタイルシートを使う。
- XML から HTML に変換して、ブラウザで確認する。
- XSL (XSL-FO) 技術を使う。XML を XSL-FO に変換して XSL-FO 組版エンジンで表示、印刷する。

HTML 仕様と XSL-FO の仕様の関係。HTML が画面の上で情報を閲覧する目的で設計されているのに対し、XSL-FO はページ概念をもつ媒体 (紙、PDF) に対して情報を綺麗にレイアウトして出力するものです。

HTML を閲覧するツールが Web ブラウザで、XSL-FO を印刷するツールが XSL-FO 組版エンジンです。この 2 つは、競合関係にもありますし、画面出力と印刷出力という補完関係にもあるといえます。

XSL を使った表示/印刷のステップ

XSL を使った XML の表示/印刷は次のステップで行われます。



XSL 組版エンジン

XSL-FO インスタンスは、コンテンツにフォーマット・オブジェクトとプロパティでレイアウト指定を施した文書です。同じ XSL-FO インスタンスを異なる XSL-FO 組版エンジンで処理した結果は、大体、同じ組版結果になります。⁽¹⁾

アンテナハウスでは、XSL V1.0 仕様に準拠した組版エンジン「XSL Formatter V2」を 2002 年 1 月から全世界で販売しています。

Antenna House XSL Formatter V2.3

- ・プラットフォームは Windows95/98/Me/NT4.0/2000/XP。近く、Solaris、Linux 版も出荷予定。
- ・Windows 版は ActiveX コントロールの組版エンジンと、それを制御するユーザーインタフェース・プログラムから構成されています。組版エンジンはコマンドラインや VBScript から COM 経由で使用できるインタフェースを備えています。サーバーでも使用できます。
- ・Windows 版では、組版結果は Windows 画面かプリンタに出力します。Acrobat Distiller をプリンタとして指定して印刷し、PDF を作成することもできます。
- ・独自開発の PDF 出力エンジンを使用すれば自力で PDF ファイルの生成も可能です。PDF オプションは、接続数無制限のサーバ上でお使い頂けます。フォントの埋め込みもできますので、世界中に流通する PDF を作成できます。
- ・Windows 版は、TIFF、JPEG、BMP を初めとする主要なラスター・イメージ、EPS の表示、印刷、PDF への埋め込みができます。SGV、MathML をプラグインを使ってレンダリングして埋め込むこともできます。

世界の XSL-FO 組版エンジンの比較

世界には、XSL-FO 組版エンジンが幾つかあります。主要なものは、アンテナハウスの XSL Formatter V2.3 と RenderX の XEP の 2 つです。次に、2002 年 11 月 16 日に Yahoo の XSL-FO グループに投稿された「XSL-FO の実装比較」と題するメールをご紹介します。

Message: 1

Date: Sat, 16 Nov 2002 11:14:48 -0600

From: "W. Eliot Kimber" <eliot@isogen.com>

Subject: Evaluating XSL-FO Implementations

[I have cross-posted this to the xsl-list and to xsl-fo@yahoogroups.com. I have started using XSL-FO@yahoogroups.com exclusively for my posts on XSL FO, but I thought this message needed a wider audience.]

At XML 2002 I will be delivering a paper on the suitability of XSL FO for production-quality composition. As part of that effort, as well as an ongoing aspect of my job as an XML systems integrator and as a developer of an XSL FO course, I have been doing deep evaluations of all the XSL FO implementations that I know of. As a side effect of that, I often post here with responses of the form "Implementation X does Y".

I thought it would be useful to the community to list the implementations I'm working with and their current status with respect to FO features implemented and extensions provided. In particular, I don't want anyone to think that I am privileging any implementation in any way--my job as an integrator requires that I understand all the available options so that I can make the best recommendations to my customers. ISOGEN is (or soon will be) partners with all the vendors who produce XSL FO implementations--we have no vested interest in any particular implementation and, as a matter of company policy, fiercely guard our product neutrality.

In addition, my focus is on production-quality systems, so I am focused on value, as opposed to cost. That causes me to prefer depth of features over license cost--that is more features implemented at a higher cost may reflect higher value than fewer features at a lower cost. Also, when implementing a production system, the license cost for software is often the smallest single cost, far outweighed by the cost of integration, configuration, and maintenance. This means, in particular, that the

(1) 一般に 2 つの組版エンジンの組版結果を完全に一致させるためには、和文では禁則処理、欧文であれば Hyphenation のアルゴリズム等を含む行組版の方法を完全に一致させる必要があります。XSL-FO 仕様では、行組版のアルゴリズムまで規定しているわけではなく、組版エンジンの実装者に任されています。例えば、ページ数の多い文書を組版する場合、禁則処理の違いで文字の改行位置が異なり、そのために、行数が変わって、結果的に総ページ数まで異なってしまいます。

XSL Formatter V2.3 のご紹介

implementation that implements the most features sets the bar for all other implementations, regardless of license cost. It also means that once a set of implementations implement all the useful features of FO, that they will be competing almost entirely on the quality of their extensions, performance, maintainability, platform support, and customer service.

Implementations I know of, listed in order of feature completeness:

- XSL Formatter, Antenna House (www.antennahouse.com). Currently Windows only. Free evaluation version available.
- XEP, RenderX (www.renderx.com). Pure Java. Free evaluation version available.

With the 3.12 release of XEP, XSL Formatter and XEP are very closely matched for features and extensions. With a few small exceptions, it is possible to write style sheets that will give nearly identical results from both implementations. And with just a little bit of conditional processing, one can write a style sheet that will generate extended FO for either product given an input parameter that specifies the target implementation. The main difference here is how support for PDF bookmark generation is implemented.

All of my testing to date has been with these two products simply because they implement the most features and allow me to create working examples of almost every feature for use our FO course.

- Epic 4.3, Arbortext (www.arbortext.com). Multi-platform (Windows, various Unix. No Linux). Fairly complete implementation that is constantly being improved. Suffers from constraints imposed by the underlying FOSI-based composition engine. Arbortext has announced the intent to provide a revamped composition engine in their 5.x version, due middle of 2003. Possible to create style sheets that will work with Epic, XEP, and XSL Formatter.

Have started testing Epic using the samples and examples developed for the FO course.

The remaining implementations, listed below, have yet to implement a number of FO important features. Of these, only FOP is really usable for any kind of production work--that is, it implements enough FO features to be able to satisfy many layout requirements and what it does it does well enough.

--中略--

In the works:

- Adobe Document Server. I'm told by Adobe that they are in the process of developing an FO implementation on top of Framemaker+SGML that will be part of the Adobe Document Server product sometime around the end of 2002. Don't know anything more about it. Framemaker+SGML certainly has all the composition features one would need to implement FO rendering semantics--the trick will be interpreting the FO spec and providing good performance. Could be an interesting solution, especially for existing Adobe customers who would like to start using FO.

Cheers,

Eliot



XSL 仕様の意義

既存の方法は

XML を組版する方法は大きく分けると次のようになります。

- ・ DSSSL などの SGML 印刷ツールを使用する。
- ・ 組版ソフトの XML インポート機能を使って XML を組版ソフトに取り込んで組版する。大日本スクリーンの AVANAS PageStudio、モリサワの MC-B2、アドベント社 3B2 など。
- ・ FrameMaker などの構造化文書エディタに XML をインポートして XML を組版する。

既存ツールの問題点

主要ツールの問題

DSSSL は、ISO で標準化された SGML の変換とレイアウトの処理仕様です。従って世界標準と言えますが、仕様が複雑なため、商用プロセサは実質的に日本のネクスト・ソリューション社製しか存在しません。その他、フリーのソフトもありますが、商用利用には無理と思われる。従って、実質的にベンダ依存です。

AVANAS BookStudio、MC-B2 は優れた日本語組版機能を持っています。日本語の XML 文書を組版するには適切と思いますが、一番大きな問題は価格が高く、日本語専用であるということでしょう。後で述べるように XML は、全世界をひとつのマーケットとみなすグローバル・スタンダードです。日本語専用で、価格が高いシステムというものは XML の世界では早晩通用しなくなると考えられます。XSL-FO 組版エンジンは世界の市場を対象に開発、販売しているものですので、価格も世界標準です。

3B2 は、英国のアドベント社の製品です。アドベント社は 1980 年頃から、Unix 上で文書組版ソフトを開発、販売しており、欧米のマークアップ、多言語の組版のニーズをもつ出版社等に多くの実績があります。しかし、日本でのビジネス展開はうまくいっていないといわれています。3B2 では XML を読んでレイアウトを付加して組版できますが、純粋な XML アプリケーションではないためデータのマルチユースの際に、後加工が必要になると言われるなど XML との親和性が完全ではないようです。なお、アドベント社は、2003 年 8 月のバージョンで XSL-FO に対応すると予告しています。

FrameMaker は 7.0 が出荷開始になりました。7.0 から FrameMaker サーバが商品に加わり、サーバ上での自動組版が可能になりました。しかし、残念ながら FrameMaker サーバも、スタイルシートは FrameMaker 独自の技術である EDD を使っています。また、FrameMaker 7.0 でも、欧米系の 1 バイト系の言語と日本語しか保証しておらず、多言語組版機能では遅れています。私見ですが、FrameMaker 7.0 は時代に遅れていると考えています。

スタイルシートの相互運用性の問題

XML にはレイアウト情報がないため、どのような組版ソフトであっても、XML の要素に対してレイアウト指定するためのなんらかのスタイルシート（あるいは、スクリプト）を作成することは必須です。

スタイルシートの作成は、レイアウト指定が高度・複雑になれば、それなりに時間がかかります。また XML の構造をそのままレイアウトするのではなく、組版の前に XML の構造変換することも必要になります。その例が、目次や索引の作成です。ドキュメントが XML になっていれば、目次や索引は自動的に作成できますが、これは XML の構造変換にあたります。

DTP と XML 組版の根本的な違いは、XML 組版ではスタイルシートを予め開発し、バッチ処理で大量のデータを処理することにあります。しかし、このスタイルシートの作成は、だれでもできるものではありません。DSSSL のスタイルシートは、ほとんどネクストソリューションしかできないと思われますし、同社はスタイルシートを高額で請け負っているようです。

また、3B2 は本体も高価格で、そのうえ、スタイルシートの開発費も高くつくとも聞いています。結局は、ベンダ依存のためスタイルシートを開発する技術者が少なく、それが高価格に繋がることとなります。

スタイルシートには大きなコストが掛かります。DSSSL、3B2、FrameMaker という 3 種類の組版ソフトが混在してしまえば、ばらばらに学習し、開発し、ばらばらにメンテナンスしなければなりません。スタイルシートに相互運用性のない組版エンジンを採用すれば、スタイルシートの開発とメンテナンスが非常に大きな投資となります。

XSL のメリット

標準仕様である

XSL-FO 仕様は、W3C という Web の標準化に大きな影響力をもつ団体が定めた標準仕様です。特定のベンダのみの独自仕様ではなく、だれでも実装することができます。既に述べたように、既存の商用製品は厳しい競争を行っています。さらに、アドビも 2003 年 4 月には日本で XSL-FO プロセッサ機能をもつドキュメント・サーバを発売する予定です。アドベント社も 2003 年 8 月の次期バージョン・アップで、XSL-FO をサポートする製品を出す予定です。今後、多数のベンダからサポートする製品が出てくることが期待されます。

これは、当然の事ながら、市場において優勝劣敗の厳しい淘汰がなされ、優れた製品のみが生き残るだろうということです。そして、仮に敗れたベンダの製品を使っているとしても、別のベンダの製品に乗り換えることが簡単にできます。

XML から XSL-FO への変換は XSLT スタイルシートを使って行いますが、XSLT スタイルシートもまた W3C の標準仕様です。Eliot 氏が既に確認した通り、XSL-FO プロセッサが異なっても、相互運用の可能なスタイルシートを記述することができます。

欧米のユーザ、特に多くの企業にコンサルティングを提供している人達は、このことに気が付いていて、ベンダ独自仕様からの脱却を強く志向しています。これが、欧米の専門家が XSL-FO に注目と期待を寄せている最大の理由です。

コンテンツとスタイルの分離

さて、XML と XSL のメリットの中で一番大きなものは、**コンテンツとスタイルの分離にある**と考えています。コンテンツとスタイルを分離することで、たとえば、次のようなことができます。

- ・ ひとつのソースから、一覧表と一葉毎の帳票のようにまったく異なる見栄えの出力を得ることができます。
- ・ 相手により大きな文字で印刷物を出力したり、レイアウトを変更できます。
- ・ データベースから必要なデータを XML で取り出して、XSL でレイアウトを与えることでダイナミックな出力に簡単に綺麗なレイアウトを与えることができます。
- ・ コンテンツをデータベースから取り出したデータで自動的に作成し、それを結合、分離、ツリー変換などの加工を施して、最終成果物の内容を組み立てた上で、レイアウト情報を外側から与えて印刷することができます。

コンテンツとレイアウトの分離により情報を自在にコンピュータで加工することができるようになるわけです。

TeX

バッチ処理の組み版エンジンとして有名なものに TeX があります。TeX と比べたとき、XML+XSL の大きな違いは、TeX ではコンテンツとスタイルが完全に分離されていないのに対し、XML+XSL ではコンテンツとスタイルが完全に分離されていることでしょう。

自動組版に最適な仕様

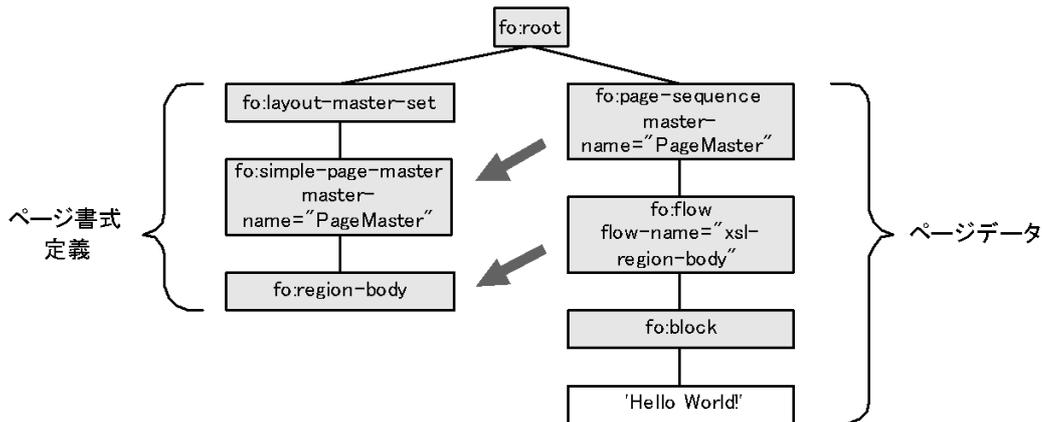
もうひとつ見逃してはならないのは、XSL-FO は自動組版に最適な仕様ということです。これは後述します。



XSL-FO とは

XSL 仕様は、「フォーマッティング・オブジェクト」(FO) と呼ぶ、ページ書式の表現、文書の段落、表、リストなど、組版のための様々な種類のオブジェクトを定義します。ページサイズやフォント・ファミリー名、フォント・サイズ、行の高さ、などは FO のプロパティとして定義しています。

XSL 仕様ではひとつの組版対象を XSL-FO 文書として表します。XSL-FO 文書は、XML ファイルで次のような構造をもちます。



XSL-FO 文書の全体構造

組版の対象となる FO には上述のように様々なものがありますが、もっとも頻度が高く使われるものが段落オブジェクト fo:block です。次の図は、fo:block の例です。

fo:blockのプロパティ

```
<fo:block
  font-size="14pt"
  space-before="0.5em"
  space-before.precedence="1"
  space-after="0.5em"
  keep-with-next.within-page="always"
  keep-with-next.within-column="always"
  start-indent="10mm"end-indent="10mm"
  id="IDwfmY1">
  見出しのブロック
</fo:block>
```

fo:block フォーマッティングオブジェクト

段落オブジェクトの設定例

XSL-FO の機能説明

XSL 仕様を使うことで、次のような専門的な印刷技術に基づいて XML をレイアウトすることが可能です。また、

- ページの書式は fo:simple-page-master で定義する。ページはヘッダ領域、フッタ領域、左右のサイドバー領域、本文領域 (fo:region-before) に分かれ、各領域の大きさを指定できる。

- `fo:region-body` には段組を指定できる。段組には全段抜きの指定ができる。
- 標準の欧文横書きのみでなく、日本語の縦書きやアラビア語、ヘブライ語のような右から左へ書き進める言語の文字進行方向や表のセル進行方向を簡単に指定できる。この指定は、本文領域全体のみでなく、表に指定すれば表のカラムや行の進行方向を変えることもできるし、表のセルの中だけで指定することもできる。
- `fo:layout-master-set` を使って、さまざまなページ書式 (`fo:simple-page-master`) の出現順を規定して、切替ができる。左右ページの書式切り替え、先頭や最後のページの書式切り替えなど。
- 文字配置 (インデント、ジャスティフィケーションなど)、行の高さの決定方法の変更、段落と段落の間隔を自由に制御できる。
- `keep-together` による段落間の改頁や改段の抑制、または `break-before/after` による改頁や改段の強制指定ができるなど自動組版向きの機能が豊富。
- `fo:marker`、`fo:retrieve-marker` を使って爪、柱の機能を使える。
- `fo:footnote` で脚注を作成したり、`before-float` でページの先頭にオブジェクトを配置できる。
- `side-float` 機能を使えば、グラフィックスを本文テキストの両脇に配置し、テキストをグラフィックスの縁に回り込ませることができる。
- ページ番号を自動生成する `fo:page-number`、オブジェクトの置かれたページ番号を参照する `fo:page-number-citation` 機能があり、目次、索引の作成が簡単。
- `fo:basic-link` により文書内、文書外へのリンクを定義できる。ビューアでこれを実装して、文書内外へのリンクを実現したり、PDF にリンクを埋め込むことができる。
- 目次や索引のための `fo:leader` 機能があり、目次の見出しとページ番号の間のリーダ等を自動的に引ける。
- Unicode との整合性が高い。特に `fo:bidirectional-override` という Unicode-BIDI 処理のための機能を定義しており、国際化時代の仕様となっている。



XSL Formatter の応用例

次に、XSL Formatter の応用例をいくつか挙げます。

月刊情報誌のページ組版

概要

項目	データ
ユーザ名	Druck & Media GmbH (http://www.druck-media.de)
ライセンスタイプ	XSL FormatterV2 (Client 版) Corporate License
稼動開始時期	2002 年 10 月
開発会社	MID/Information Logistics Group GmbH (リセラー)

ドイツ Cologne 所在の法律関係の中堅出版社が、月刊誌「GesR GesundheitsTecht (健康と法律)」誌を創刊するにあたり、組版と印刷を担当する Druck & Media GmbH 社は、広告を除く全ページのページ組版を行うツールとして XSL FormatterV2.2 を採用しました。

運用

この雑誌は、画像をまったく使っていない文章のみですが、XSL で自動組版したものとしては、複雑なレイアウトに属します。フォントは、出版社の独自フォントです。XSLT スタイルシートは、MID が開発しました。ドイツ語の TeX ハイフネーション辞書をつかって、ハイフネーションしていますが、V2.2 はハイフネーションの頻度が不足していたため、ソフトハイフンを手で追加入力しました⁽²⁾。創刊号は 2002 年 10 月 21 日に発行されましたが、XSL Formatter で組版したページを PDF 出力オプションで PDF に出力して印刷・製本しました。出版社の要求するレイアウトは、3B2 の対話式組版機能に基づくものであったため、出版社の要求するレイアウトを自動組版で実現するのに苦労したようです。出版社の方は、完全に自動化した組版とページ・レイアウト・プロセスの必要性をまだ認識していないが、印刷会社の方は満足しているとのこと。特に、2 段組で脚注を内側に集めていますが、XSL V1.0 仕様の脚注の機能と異なるためこの部分に手作業が残っているようです。

XSL Formatter 選定の理由

印刷会社から、新雑誌を新しい技術をつかって、完全に自動化した組版とレイアウト・プロセスで行いたいと考えて出版社に提案しました。この印刷会社はドイツで一番最初に 50 種類のルーズリーフ出版物を SGML と Datalogics 社の組版ソフト (Pager) で印刷した会社として有名な会社です。MID と印刷会社は共同で出版社の承認を得られるレイアウトとスタイルシートを開発しました。

⁽²⁾XSL FormatterV2.3 メンテナンス・リリース (11 月 1 日) にて、ハイフネーション頻度を増やす改良をしました。

GesundheitsRecht



1. Jahrgang - Heft 1/2002 - S. 1

VERTRAGSARZTRECHT

Auswirkungen des Weiterbildungsrechts von Ärzten und Psychotherapeuten auf das Vertragsarztrecht

Richter am Bundessozialgericht Dr. Ulrich Wemmer, Kassel

Zwischen ärztlichem Weiterbildungs- und Berufsrecht einerseits und dem Vertragsarztrecht andererseits besteht seit langem ein Spannungsverhältnis. Die Gesetzgebungskompetenzen sind verschieden, und an der in diesem Bereich besonders bedeutsamen untergesetzlichen Rechtssetzung wirken im vertragsärztlichen Bereich die Krankenkassen als Kostenträger entscheidend mit, während die Weiterbildungs- und Berufsordnungen allein von den Ärztekammern erlassen werden. Die rechtlich und bisweilen auch berufspolitisch begründeten Thesen vom Vorrang des ärztlichen Berufsrechts und einer Bindung der Normgeber im Vertragsarztrecht an dieses Recht sehen sich dem Verdacht ausgesetzt, zumindest auch auf die Zurückdrängung von Wirtschaftlichkeits- und Kostensparargumenten abzielen. Die zum 1.1.1999 in die vertragsärztlichen Versorgungsstrukturen integrierten psychologischen Psychotherapeuten bzw. Kinder- und Jugendlichenpsychotherapeuten kennen noch kein dem ärztlichen Bereich vergleichbares Weiterbildungsrecht. Ob dieser Zustand geändert werden soll, kann nicht ohne Blick auf die Auswirkungen beruflicher Spezialisierungen auf die Leistungserbringung im Rahmen der vertragspsychotherapeutischen Versorgung entschieden werden. Auf diesem Hintergrund informiert der Beitrag über die Rechtsprechung des BSG zu den Auswirkungen des ärztlichen Weiterbildungsrechts auf das Vertragsarztrecht sowie über die eigenständigen Abgrenzungsregelungen der ärztlichen Disziplinen im Vertragsarztrecht. Am Ende steht ein Ausblick auf die Rechtslage bei den Psychotherapeuten, zu der naturgemäß noch keine Rechtsprechung des BSG ergangen ist.

bewerben, der seine Eintragung in ein Arztnachweisregister, das von der Kassenzentralen Vereinigung geführt wird (§ 93 Abs. 2 SGB V). Die Eintragung in das Arztnachweisregister setzt die Approbation als Arzt sowie den erfolgreichen Abschluss entweder einer allgemeinmedizinischen Weiterbildung oder einer Weiterbildung in einem anderen Fachgebiet mit der Befähigung zum Führen der entsprechenden Gebietsbezeichnung voraus (§ 95a Abs. 1 SGB V). Das Vertragsarztrecht verweist damit zum einen auf Bundesrecht, nämlich soweit es um die Approbation als Arzt geht. Deren Erteilung richtet sich nach § 3 der Bundesärztereordnung, die als Bundesgesetz auf die Kompetenzregelung in Art. 74 Nr. 19 des Grundgesetzes (GG) gestützt ist. Danach hat der Bund die Kompetenz für die Zulassung zu ärztlichen und anderen Heilberufen. Mit dem Erfordernis der Weiterbildung verweist § 95a Abs. 1 SGB V dagegen auf Landesrecht, denn nach der Kompetenzordnung des Grundgesetzes sind die Länder für die ärztliche Weiterbildung und die Regelung des Facharztwesens zuständig.¹ Dabei ist eine Übereinstimmung der Weiterbildungsregelungen in den einzelnen Bundesländern mindestens in den Grundzügen unverzichtbar.² Die Freizügigkeit der Berufsausübung im Bundesgebiet wäre nachhaltig gefährdet, wenn ein in einem Bundesland weitergebildeter Arzt, der die Berechtigung zum Führen einer Facharztbezeichnung erworben hat, in anderen Bundesländern auf Grund eines völlig abweichenden Inhalts der Weiterbildungsordnung auf seinem Fachgebiet nicht tätig werden könnte. Die Funktion der Vereinheitlichung übernimmt in diesem Bereich die vom Deutschen Ärztetag beschlossene Musterweiterbildungsordnung, an die sich die Weiterbildungsordnungen der Ärztekammern regelmäßig anlehnen. Insoweit findet der Gesetzgeber im Krankenversicherungsrecht ein bundeseinheitlich zumindest stark vorgeprägtes landesrechtliches Weiterbildungs- und Facharztrecht vor, an das er auch mit bundeseinheitlichen Regelungen im Zulassungsrecht anknüpfen kann.

I. Vertragsarztrechtliche Ausgangslage

1. Weiterbildung als Zulassungsvoraussetzung

Um die Zulassung als Vertragsarzt kann sich jeder Arzt

2. Auswirkungen des Vertragsarztrechts

Das Vertragsarztrecht, das sich als Bundesrecht auf die Bundeskompetenz zur Regelung der Sozialversicherung

© Dr. Ulrich Wemmer, Richter am BSG, Mitglied des 6. Senats (Vertragsarztrecht und Vertragskassenarztrecht).
 1 BVerfGE v. 9.3.1972 - 1 BvR 318/62, BVerfGE 33, 125.
 2 Haas in: Kötzler/Konemann, § 95a SGB V Rz. 3.

GesR GesundheitsRecht 創刊号 1 頁

After Sales Information System

概要

項目	データ
ユーザ名	Sorman Information AB (スウェーデン Vaxijo) の After Sales Information System に組み込んで再販売。
ライセンスタイプ	XSL FormatterV2 OEM ライセンス
稼働開始時期	2002 年 8 月
開発会社	Sorman Information AB (OEM パートナー)

Sorman After Sales Information System (SAIS) について

SAIS は Sorman 社が欧州の自動車メーカ、電機メーカ向けに受注製作しているシステムです。製品を販売した顧客の満足度を高め、ブランド力の強化、顧客との長期に渡る関係の構築のためのシステムです。SAIS は、次のシステムからなります。

- ・ 情報・ドキュメントを蓄積するサーバ
- ・ PDM、ERP、CRM などから必要な情報を取り出す API
- ・ アフター・セールス情報を制作するためのクライアント
- ・ プロジェクト全体を管理する管理用端末
- ・ コールセンター用のクライアント・システム
- ・ 顧客が Web や Mobile 端末で情報を取り出して見ることができるようにする配信システム

SAIS では、XML の利用により、対象グループが必要とする情報を取り出して、紙、CD-ROM、インターネットによる情報配布等、多チャンネルの情報配信が可能です。

XSL Formatter の選定理由

欧州では紙へ印刷して保存することが法律で定められているものが多いため SAIS にとって印刷と PDF 化は必須です。ドキュメントは、断片的に XML ツリーでできていて、それを組み合わせてレポートを作成して紙に印刷したり、PDF 化します。

従来、SAIS では FrameMaker+SGML、または、Epic/E3 を使って PDF 化をしていました。FrameMaker+SGML では EDD を開発しなければならず、Epic/E3 の場合は、FOSI によるフォーマット指定を行います。これらはベンダ独自の異なるスタイル付け技術のため、Sorman の開発者は標準技術である XSL によるフォーマット指定を待ち望んでいました。XSL Formatter の OEM となることで、同社の販売する SAIS に XSL Formatter と PDF 出力オプションを組み込んで納品することが可能になり、標準技術を使って安価なシステム構築ができるようになりました。

MSDS (Material Safty Data Sheet) 発行管理システム

概要

項目	データ
ユーザ名	JSR 株式会社
ライセンスタイプ	XSL FormatterV2 サーバライセンス
稼動開始時期	2002 年 9 月
開発会社	JNT システム株式会社 (リセラー)

本システムは化学製品を購入した顧客に過不足なく MSDS を提出することを目的とするシステムです。

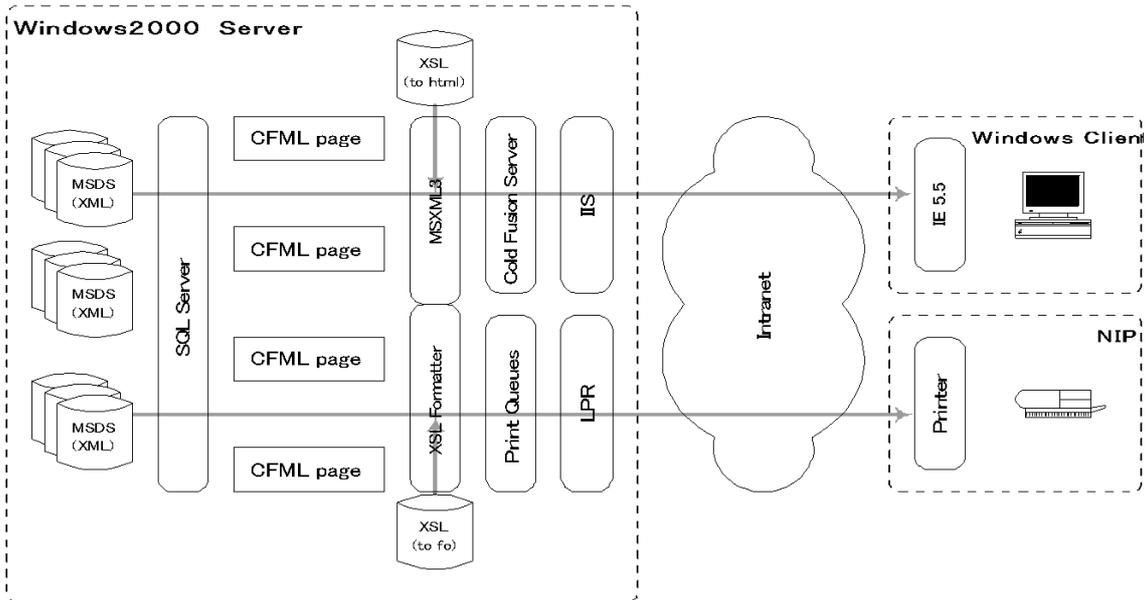
MSDS は、化学製品の取扱説明書のようなもので、以下に挙げるような情報が記述されており、製品を購入先への提出が一部義務付けられています。

- ・ 製造者 (社) に関する情報
- ・ 製品に何が含まれているか
- ・ 製品/成分の、物理/化学的性質
- ・ 危険有害性に関する情報
- ・ 飲んでしまったら/目に入ったら/皮膚に付いたら、どう対処するのか
- ・ 火がついたら/こぼれたら、どう処置するのか
- ・ 取り扱うときに/保管するときに/棄てるときに/運ぶときに、留意する点
- ・ 関連法規制や参考文献等

ひとつの MSDS は大項目が 16 あり、各大項目は中項目、小項目と分かれていて記述すべき内容が決まっています。MSDS は、どのような成分がどれだけ含まれているかによって記述ボリュームが変わりますが、本システムではひとつあたり 4 ページから 8 ページの文書となります。各項目は文章情報や表からなり、可変の長さになります。

運用

製品別の MSDS 内容データは、別途、用意されてデータベースに格納されています。営業担当者が顧客別に提供すべき MSDS をシステムに問い合わせ、印刷を要求することによって、製品別の MSDS 内容データが顧客情報、システム情報とマージされて Web ブラウザでプレビューできます。営業担当者の確認後、印刷指示により、XSL Formatter がサーバ上で組版を行い、結果をプリンタに出力します。フォーマットの基本は 1 種類ですが、若干のバリエーションがあり、これは XSLT で変換するときにコントロールされます。



システム構成図

XSL Formatter 選定の理由

- ・ MSDS が、伝票/帳票と云った定型文書ではない為、ダイナミックに印刷物を得られるツールを探していた。
- ・ 画面でのプレビューは必須の機能であるために、ワンソースマルチユースの仕組みを実現したかった。
- ・ 電子文書化の要請もあり、同じ仕組みを使つての PDF 生成が可能とみられた為（採用時、「PDF 出力オプション」は未発売。）
- ・ XSL 自体は、W3C で制定された「由緒正しい」規格である為、将来的にも安心できる。
- ・ 機能差の無い試用版が入手できたため、十分な評価が可能であった。
- ・ 製品版の値段も安く、導入に際してのハードルにならなかった。



XSL Formatter による自動組版

自動組版自体は、特に新しいものではありません。QuarkXpress のような代表的 DTP ソフトでも CSV、XML を読み込んで自動組版することができます。この場合、AppleScript 等のスクリプト言語を使って、アプリケーションがデータを読み込む動作を制御します。日本語組版ソフトもスクリプトを使って、XML を初めとする外部ファイルをインポートして自動的にページ組版する機能をアピールするものも珍しくありません。

従来の自動組版ソフトと比較して、XSL による自動組版には特に次の特徴があると考えます。

- XSL-FO の仕様上、テキストのボリュームが可変で、出来上がるページ数も可変になるような書類、文書を組むための機能が充実しています。XSL-FO の仕様の設計者は、自動組版を強く意識しています。
- XML を組版して出力するまでのプロセスがシームレスであること。すなわち、XSLT によるツリー変換、XSL-FO プロセッサによるページの生成、PDF 出力という一連の流れがシームレスに操作できます。これは、アプリケーションを作成する面からは簡単にシステムを作ることができることを意味しています。
- コンテンツとレイアウトが完全分離されています。そしてコンテンツである XML の加工を XSLT によるツリー変換や DOM プログラミングという標準手法で行うことができます。このことから、コンテンツの生成、更新、変更をダイナミックに行い、それに応じて組版出力を直ちに得ることができます。
- XSL-FO プロセッサはサーバ上で動作すること。XSLFormatterV2.3 は、現時点で Windows サーバ専用ですが、現在、マルチプラットフォーム・フォーム化しています。アンテナハウスは Solaris 版と Linux 版を 2003 年第一四半期に出荷開始する予定です。
- インターネット、イントラネット・サーバ上で PDF の出力ができます。PDF 出力は、Acrobat Distiller を使うこともできますが、Distiller のサーバ・ライセンスは高価です。XSL Formatter は自力で PDF を生成するエンジンとして「PDF 出力オプション」を持っていますので、無制限接続のサーバ上で、XML 組版を行い PDF を返信するシステムを簡単に、かつ、安価に構築できます。



多言語組版について

XML と多言語組版

多言語の製品説明書を、同時進行で作成するのに XML と XSL 技術が有効です。特に、XSL は国際化を念頭において設計された仕様になっています。アンテナハウスの XSL-FO 組版エンジン XSL Formatter は、V2.3 で世界で初めて、日本語や英語、西欧の言語のほか、中国語、韓国語、アラビア語、ヘブライ語、タイ語のような言語までを含む XML を組版して PDF を作成できる機能を備えました。

多言語組版の予備知識

文字と言語

言語は文字によって記述されます。言語をコンピュータで扱うためには、まずその前提として、言語を表記する文字を扱うことができなければなりません。

1980 年代から 1990 年代までのパーソナル・コンピュータの OS は、各地域別の文字コード表に基づいていたため、OS の上で動作するアプリケーション・プログラムも地域別でした。例えば、日本語 WindowsMe までは、OS 内部ではシフト JIS (JISX0201 と JIS X0208) でテキストと文字処理を行っています。このため、WindowsMe で動くアプリケーションでは、A with diacresis : Ä、O with diacresis : Ö、U with diacresis : Ü などのラテンの特殊文字を正しく処理するのが容易ではありません。これらの文字は日本語の半角カタカナと文字コードがぶつかってしまうためです。

Unicode

現在では、Unicode が単一のアプリケーションで全世界のあらゆる言語の文字とテキストを取り扱う基盤を与えています。Unicode の 1.0.0 版は 1991 年 10 月に発行されましたが、その後、10 年以上に渡り改良されてきました。また、Microsoft を初めとする主要な OS やアプリケーション・ソフトウェアのメーカーが Unicode を採用することで、その勢力を広げてきました。現在、Unicode の最新版は 3.2 版で約 300 の言語を扱えるという。

Unicode は単に符号化文字集合を定義するだけではありません。Unicode 文字データベースによって文字毎のを進行方向、Unicode Line Breaking Properties で禁則文字など文字の改行特性、UnicodeBIDI で双方向性処理、など、テキスト処理の各種の基準を定義し、アプリケーション・プログラムを作成時に参照となる実装を提供しています。

フォント

言語をコンピュータで処理する時の次の基盤はフォント技術です。実際に使えるフォントがなければ、組版をすることもできませんし、その結果を画面に表示したり、印刷できません。次の表は Microsoft Windows2000/XP に標準で添付されているか、あるいは、インターネットで無償ダウンロードできる Windows2000/XP 用のフォントのリストです。

この中で、Arial Unicode MS は、Unicode の全域をカバーする唯一のフォントですが、まだ Unicode2 のレベルであって、Unicode3.2 の全文字は揃っていませんし、また、必ずしも品質は良くないという欠点があります。しかし、言語別にみれば欧米の言語、スラブ系言語、日本語、中国語 (繁体字、簡体字)、韓国語、アラビア語、ヘブライ語、タイ語等については TrueType または OpenType (TrueType フォーマット) のフォントを無償で揃えることができます。もちろん、デザインを重視した場合、このようなフォントのみでは不足ですが、技術説明書、製品のサービス説明書などの用途では十分実用的でしょう。

フォント・ファミリー	カバーする主要な文字	入手方法	分類
Arial Unicode MS	Unicode V2 全文字	Office2000/XP etc.	Sans-serif

フォント・ファミリー	カバーする主要な文字	入手方法	分類
Arial	Latin,Greek,Cyrillic,Arabic, Hebrew	2000/XP	Sans-serif
Courier New	Latin,Greek,Cyrillic,Arabic, Hebrew	2000/XP	Monospace
Lucida Console	Latin,Greek,Cyrillic	2000/XP	Monospace
Lucida Sans Unicode	Latin,Greek,Cyrillic, Hebrew, 記号類	2000/XP	Sans-serif
Microsoft Sans Serif	Latin,Greek,Cyrillic,Arabic, Hebrew, Thai	2000/XP	Sans-serif
Tahoma	Latin,Greek,Cyrillic,Arabic, Hebrew, Thai	2000/XP	Sans-serif
Times New Roman	Latin,Greek,Cyrillic	2000/XP	Serif
Vernada	Latin,Greek,Cyrillic	2000/XP	Sans-serif
Arabic Transparent	Arabic	2000/XP	Sans-serif(Latin), Cursive (Arabic)
Traditional Arabic	Arabic	2000/XP	Sans-serif(Latin), Cursive (Arabic)
Sylfaen	Latin, Greek, Cyrillic, Armenian, Georgian	XP	Serif
MS Hei	Simplified Chinese	IE5, Global IME5	Monospace(Latin), Sans-serif(Chinese)
MS Song	Simplified Chinese	IE5, Global IME5	Monospace(Latin), Serif (Chinese)
SimSun	Simplified Chinese	XP	Monospace(Latin), Serif (Chinese)
MingLiU	Traditional Chinese	2000/XP	Monospace(Latin), Serif (Chinese)
PMingLiU	Traditional Chinese	Office2000	Serif
Mangal	Devanagari	2000/XP	
Palatino Linotype	Greek Polytonic	2000/XP	Serif
Shruti	Gujarati	XP	
Raavi	Gurmukhi	XP	
David	Hebrew	2000/XP	Serif
David Transparent	Hebrew	2000/XP	Serif
Fixed Miriam Transparent	Hebrew	2000/XP	Monospace
Miriam	Hebrew	2000/XP	Sans-serif
Miriam Fixed	Hebrew	2000/XP	Monospace
Miriam Transparent	Hebrew	2000/XP	Sans-serif
Rod	Hebrew	2000/XP	Monospace
MS Gothic (MS ゴシック)	Japanese	2000/XP	Monospace(Latin), Sans-serif(Japanese)
MS Mincho (MS 明朝)	Japanese	2000/XP	Monospace(Latin), Sans-serif(Japanese)
Tunga	Kannada	XP	
Batang	Korean	2000/XP	Serif

フォント・ファミリー	カバーする主要な文字	入手方法	分類
Gulim Che	Korean	IE5, Global IME5	Monospace(Latin), Sans-serif(Korean)
Estrangelo Edessa	Syriac	XP	
Latha	Tamil	2000/XP	
Gautami	Telugu	XP	
MV Boli	Thaana	XP	
Angsana New	Thai	2000/XP	Serif
Cordia New	Thai	2000/XP	Sans-serif
IrisUPC	Thai	2000/XP	Sans-serif

PDF 技術

多言語組版のもう一つの推進要素は PDF 技術でしょう。多言語の説明書を組版することができても、それを紙に印刷して持ち歩かなければならないということでは、全地球をカバーする電子化時代には間に合いません。

紙を電子化した媒体である PDF ができたことによって、多言語の組版結果をインターネット経由でどこにでも送信したり、CD-ROM などの電子媒体で流通させることができるようになりました。

重要なことは、PDF にフォントのアウトラインの埋め込みができるようになったことです。国境を越えて PDF を流通させたり、アラビア語やタイ語を組版して PDF 化した場合は、アウトラインを埋め込んでいない PDF には実質的に意味がありません。PDF にフォントのアウトラインを埋め込んで流通させることができることで、多言語組版が名実ともに実用になりました。

XML と XSL 技術

XML と XSL は、Windows、Unicode、フォント、PDF という多言語組版の基盤技術の上に構築されます。

XML

XML は多言語の文書のコンテンツを表現するのに最適です。その理由は、

- XML は、UTF-8、UTF-16 という Unicode ベースの符号化方式を標準として採用しています。従って文字コード変換をすることなく Unicode テキストを扱えます。
- XML では文書ファイルを分割して作成したり、あるいは、画像ファイルは外部ファイルとして置くことができます。この仕組みを使って、文書作成時にいろいろな言語の文書で異なる部分は別ファイルとして作成、共通の部分は一つとし、画像はすべての言語で共通にするというような構成として、組版時にパーツ文書を組み立てて最終形にするような方法が取れます。
- ひとつで世界のあらゆる言語を扱える WYSIWYG 編集ソフトは、Microsoft Word しかありません。これに対して、多言語を扱うことのできるテキスト・エディタは、Windows の NotePad や、UniPad などがあります。XML はテキストファイルなので、必要に応じてテキスト・エディタを使って記述することができます。

XSL

XSL は、次の点で国際化を考慮した仕様です。

Writing-mode

文字の進行方向と行の進行方向を指定できます。ページレベルで日本語の縦書き、あるいは、アラビア語のように文字を右から左に書く言語用に使う他、表に指定すれば表のセルの列と行の進行方向を指定できます。

UnicodeBIDI と fo:bidirectional-override

ラテン・アルファベットや日本語のような左から右へ記述する文字と、アラビア文字やヘブライ文字のように右から左へ記述する文字が混在すると、文字の進め方が曖昧になることがあります。多言語混交の問題解決のために、Unicode では UnicodeBIDI という処理方法を定めています。XSL では fo:bidi-override を使って UnicodeBIDI を制御できます。

言語別の組版ルールについて

XSL は、既述のように組版の対象となるオブジェクトを、文字レベル、インライン・レベル、ブロック・レベル、リスト（箇条書き）、表、リンクなどのダイナミック・オブジェクト、ページの領域（本文、ヘッダ、フッタ、サイドバー、あるいは、脚注領域、前方フロート領域、サイド・フロート領域）などで定め、それぞれについてのレイアウトや修飾の指定をプロパティで与えます。各言語独自の組版ルールの大部分は、XSL のオブジェクトとプロパティの指定方法で実現できます。一方、次の項で挙げるように XSL-FO プロセサで実現しなければならない部分もあります。

XSL Formatter と多言語組版

XSL Formatter が独自に解決している多言語組版の課題は、次のようなことです。

グリフ置換の問題

日本語や中国語の縦書きの時には、同じ文字コードを横書きとは異なるグリフで表示したり印刷する必要があります。Windows の画面や、Acrobat による PDF 作成は Windows の機能で自動的に行われますが、PDF 出力オプションでは、TrueType/OpenType の縦書き処理用の Feature をプログラムでセットして縦書きのグリフが表示されるようにします。アラビア語では、同じ文字コードが単語の開始位置、中間、終了位置でグリフが変化します。タイ語では、Unicode の複数文字をまとめて一つの文字の上下に数階建てで表示します。この処理は Windows 画面と Acrobat での PDF 化では Windows がグリフを変化させますが、PDF 出力オプションでは自分でグリフの置き換え処理を行います。

禁則処理

禁則処理の規則は XSL では規定されていません。しかし、Unicode に文字別の Line Breaking Properties が定義されています。XSL Formatter は、これに準拠して禁則処理を行っています。なお、行頭、行末禁則文字を追加したり、削除するカスタマイズは V2.3 では変更できません。将来、機能を追加します。

ハイフネーション

XSL では、ハイフネーションのオン/オフ、ハイフネーション処理に関する幾つかのプロパティを定義しています。XSL Formatter は、TeX のハイフネーション・アルゴリズムを実装しています。言語別の TeX のハイフネーション・パターン辞書（を XML 形式にしたもの）を使ってハイフネーションを行うことができます。ハイフネーションの頻度は、V2.3 メンテナンス・リリース 1 から Chicago Manual の頻度にほぼ同じになるように調整しました。

両端揃え（ジャスティフィケーション）とワード・スペーシング

行のレイアウトで両端揃えを指定することは、様々な言語で一般的です。しかし、両端揃えを指定した時の改行位置の決定方法、文字間の調整、単語間の調整のルールは言語によって異なります。日本語は、文字と文字の間隔で調整します。また、英語ではワード・スペーシングが多少変化しても良いが、アラビア語では、ワード・スペーシングが変化してはなりません。このためアラビア語の両端揃えでは、文字間にカシダというグリフを挿入して単語の長さを伸ばすことで両端揃えを実現しています。

綺麗な組版

なお、英語でも、ワード・スペーシングが大幅に変化することは綺麗な組版という面では望ましくありません。両端揃えでは、改行位置の調整、フォント・ストレッチやレター・スペーシングを調整することで綺麗な組版を実現するのが望ましい。V2.3 ではそこまでは実装できていません。高品位な出版物の組版を実現するためには、急いで解決すべき問題と考えています。

多言語の混在文書の組版

日本語や英語のような左から右に記述する文章の中にアラビア語やヘブライ語のような右から左に記述する文字が混在することがあります。これを BIDI（双方向性）といいます。

進行方向の異なる文字が入れ子になっていると、文字の進行方向に関して曖昧さが生まれることがあります。これを処理するため、Unicode は BIDI アルゴリズムを定義しています。Unicode の BIDI は、文字の特性に基づく暗黙の部分と、埋め込み、オーバライドを明示的に制御する部分があります。

XSL では、Unicode の方向整形コード、または、fo:bidirectional-override を使って処理を指定します。XSL Formatter はこれを正しく処理できます。次に簡単な例を上げます。

FO の中でアラビア語を括弧で括った文字列の場合

```
<fo:block>ضمض(ضمض) ENGLISH</fo:block>
```

括弧は、文字の方向性を持たない中立(ニュートラル)の文字です。一般的に、中立の文字は周辺の文字の方向性に影響を受け、Left-to-Right と Left-to-Right にはさまれた文字は Left-to-Right になり、Right-to-Left と Right-to-Left にはさまれた文字は Right-to-Left になります。この周辺の方向性が対立した場合は、上位の方向性に従うので、fo:block の writing-mode と同じになります。

したがって、上の fo:block は次のように表示されます。

ضمض(ضمض) ENGLISH

これを回避する方法の 1 つは、Unicode の方向整形コード (RLM、RLE) を使うことです。⁽³⁾

RLM を使った例

```
<fo:block>ضمض (ضمض) &#x200F;ENGLISH</fo:block>
```

RLE を使った例

```
<fo:block>&#x202B;ضمض (ضمض) &#x202C;ENGLISH</fo:block>
```

この二つはいずれも次のように表示されます。

ضمض(ضمض)ENGLISH

あるいは、fo:bidirectional-override を使って同じことを指定できます。

多言語組版の将来

これまでの組版ソフトで、ただひとつのバージョンで世界の主要な言語をすべて扱うことのできるものはなかったと思います。XSL Formatter V2.3 によって、説明書のレベルであれば、全世界の言語を組版することが可能になりました。しかし、V2.3 では製品の説明書を漸く多言語で組版できるようになったということであって、初級レベルをクリアしたに過ぎません。今後は、全世界の言語を高品位に組版できる、という目標に向けて改良を続けていきたい。専門家の皆様の教をを請いたいと考えています。

⁽³⁾FO のデータの例では、Unicode の LRO (U+202D : 続く文字列は左から右方向として扱う。) を使ってアラビア語の文字を強制的に (入力順に) 左から右に表示されるようにしています。組版後の表示順とは逆になっていますので注意。