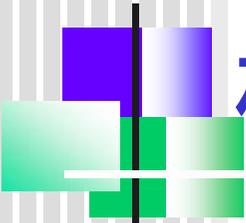


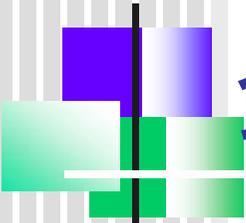
XSL-FOを用いてJepaXを冊子誌面にレイアウトする仕組み

クロスメディアパブリッシング部会
技術系サブグループ

A decorative graphic consisting of overlapping colored squares (purple, green, white) and a black crosshair.

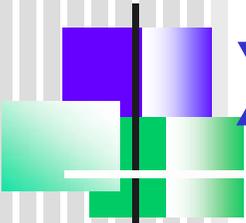
概要

- 技術系サブグループでは、XML文書を冊子誌面にレイアウトする規格であるXSL-FOに変換し、誌面を自動生成する仕組みを作成した。
- JepaXの例を紹介する。

A decorative graphic consisting of overlapping squares in purple, green, and white, with a black crosshair.

発表内容

- XSL-FOについて
- XSLを作成した理由
- JepaXを選んだ理由とJepaXの概要
- Apache FOPを選んだ理由
- 進め方
- 苦労した点

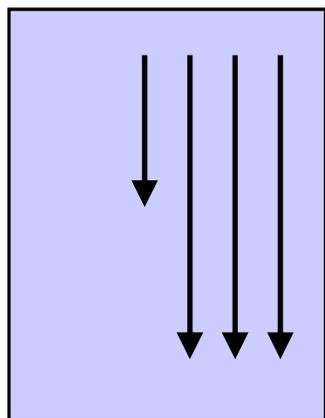


XSL-FOとは

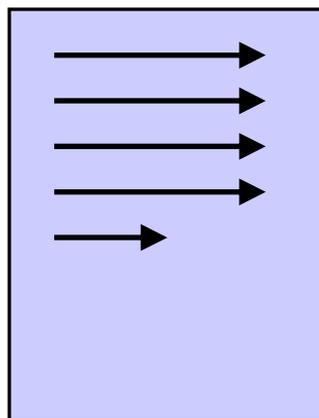
- 組版（誌面レイアウト）のための規格
 - W3Cが開発、勧告
 - XMLで記述
 - XSL=XSL-FO + XSLT (FOはXSLの一部)
 - 2001.10 Ver.1.0 勧告
 - 2006.12 Ver.1.1 勧告
 - 2008.03 Ver. 2.0 Working Draft
 - 複数ページで構成される媒体を想定
 - 目次、索引、ヘッダー、フッター、脚注

XSL-FOとは

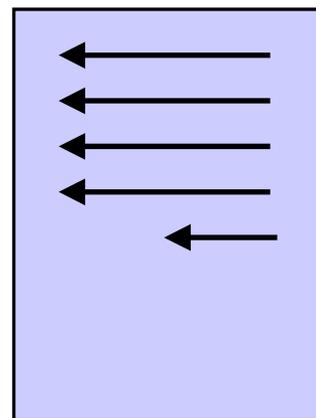
- 種々の言語に対して使われることを想定
 - 文字の流れ方向: 縦、横(→、←)



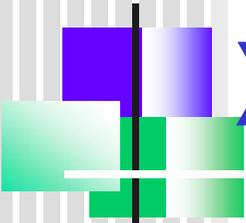
日本語



英語



アラビア語

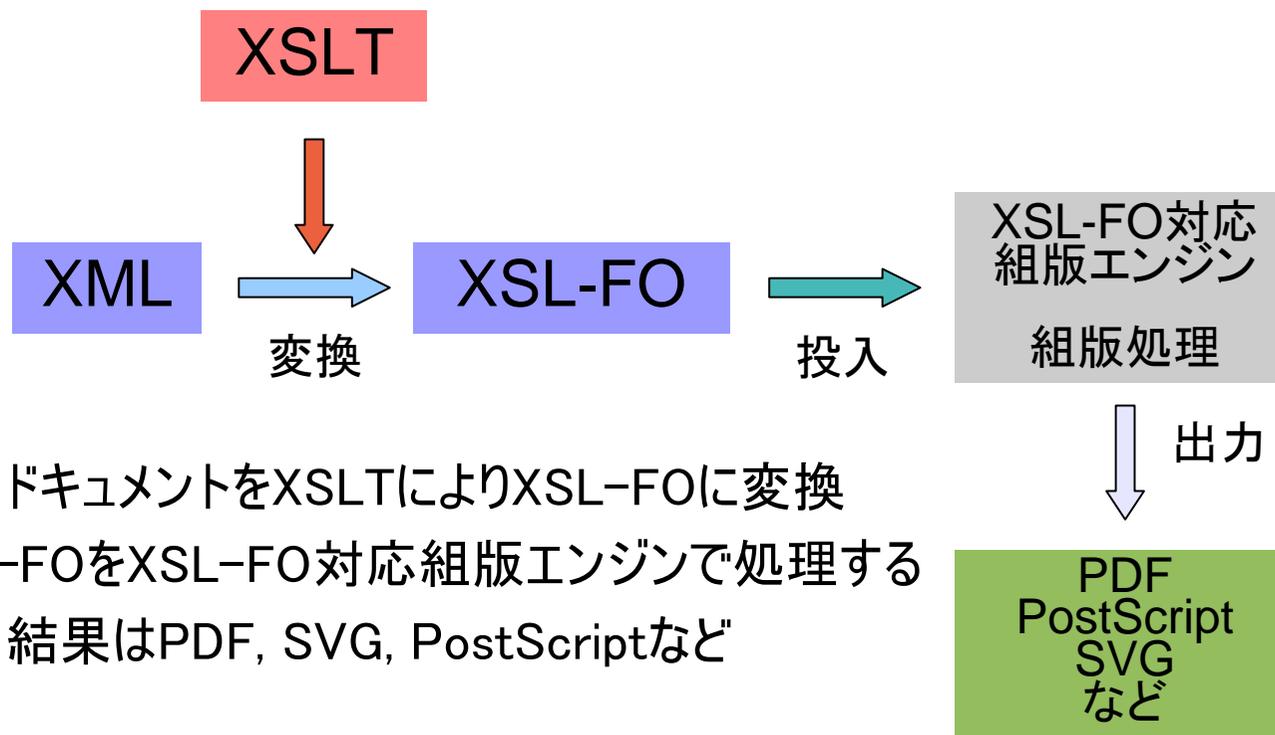


XSL-FOとは

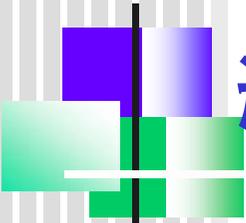
- 詳細なレイアウトの設定が可能
 - 誌面の読みやすさ
 - レイアウトされた誌面全体のバランスに大きく左右される
 - 余白、フォント、フォントサイズ、行間
 - テキスト配置方法(右寄せ、左寄せ、均等配置…)
 - グラフ・図・写真の配置方法
 - Etc…
- 海外では普及しているが、日本国内ではこれから

XSL-FOとは

- XSL-FOを用いた制作フロー

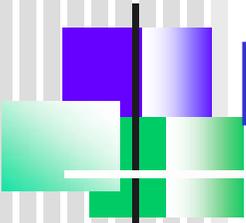


- XMLドキュメントをXSLTによりXSL-FOに変換
- XSL-FOをXSL-FO対応組版エンジンで処理する
- 出力結果はPDF, SVG, PostScriptなど



組版（ページレイアウト）とは

- 文字、イラストや写真などをページ上に配置すること
 - 活版 11世紀？
 - 活字を並べて印刷用原版をつくる
 - 写真植字 1924～
 - 文字のネガを光学的に印画紙に焼き付ける
 - DTP（ページレイアウトソフト） 1986～
 - WYSIWYGでページのレイアウト作業が行える
 - PageMaker, (LaserWriter, PostScript)



DTP vs. FO

• DTP

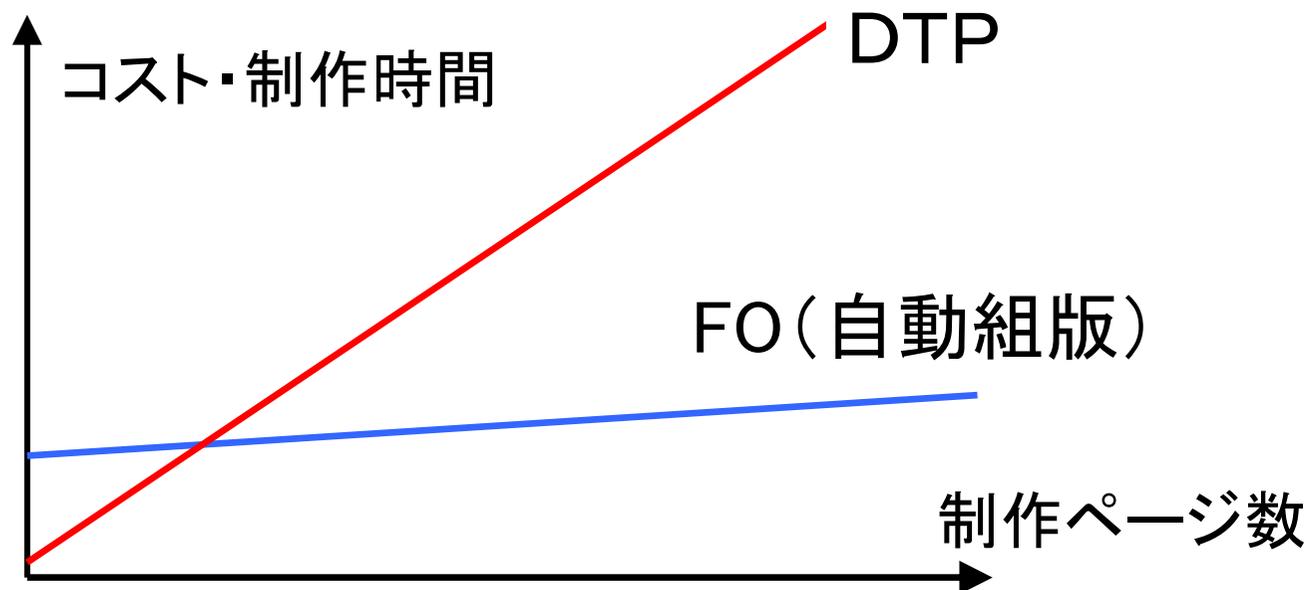
- 基本は手作業
 - ページ数増加 = コスト、時間増加
 - 人件費の高い日本ではコスト増加の主要因
- コンテンツとレイアウト情報が混在
 - コンテンツの再利用が難しい
- 各社独自フォーマット
 - 特定のソフト、ベンダーにロックインされる

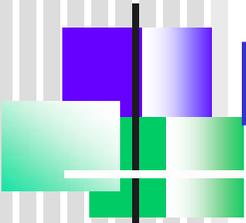
• FO

- 自動組版
 - ページ数が多くなっても、コスト、時間はあまりかわらない
- コンテンツとレイアウト情報を分離
 - コンテンツの再利用が容易
- W3Cの規格
 - オープンな規格

DTP vs. FO

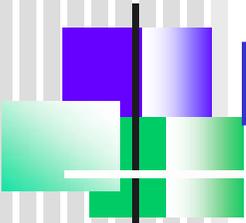
- ページ数が多くなると、FOがコスト・制作時間の点で有利





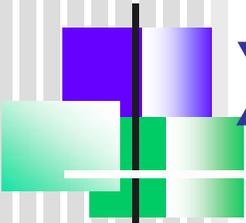
FO の特徴

- 目次、索引、ランニングヘッダ …の生成
 - DTPの場合
 - 多くの場合、特に手間がかかる
 - FOでは、自動生成可能
 - コンテンツに改変があった場合でも、それに応じた変更が自動で行える
- リンク情報の生成
 - PDFのリンク情報を生成する



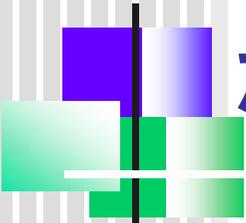
FO の特徴

- 内容駆動 (Content Driven Layout)
 - 任意の長さのコンテンツを、所定の規則に従ってページ上に流し込んでいく
 - 書籍、論文、法務関係文書
 - 現在のFOは、この自動組版に向いている
- 割り付け駆動 (Layout Driven Layout)
 - コンテンツを流し込む領域が決まっている
 - コンテンツは、その領域に入るよう調整が必要
 - 新聞、雑誌
 - 次期FOのバージョン(2.0)に期待



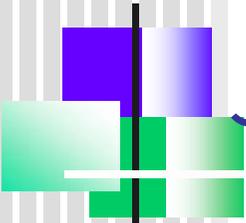
XSLの有効性

- 内容駆動型冊子誌面作成の際、ページ数が多くなってもコスト、時間があまり変わらない
 - コンテンツ内にレイアウト情報を含まないため他の多くのメディアへの再利用が容易
 - 特定ベンダに依存しない
- メリットはあるが国内の普及度は低い



なぜXSLを作成したのか

- XSLの国内での普及度は低い
 - 日本語のドキュメントが少ない
 - スキル・ノウハウを身につける場が無い→ 技術系サブグループとして集まって勉強
- 本年度の集大成
 - 興味を持った方が試せるXSL-FOを用いた自動組版の仕組みとして作成



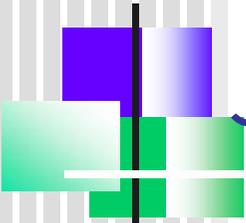
JepaXとは

■概要

- 日本の出版社・印刷会社などで構成される日本電子出版協会において開発された
- 電子出版コンテンツの蓄積・交換を行うためのフォーマット

■JepaXの目的

- 電子書籍は、メーカー独自の技術によって、高度な暗号化が行われる
- そのような配布フォーマットに左右されずコンテンツを長期保存するための、標準フォーマット
- なお、JepaXは、「ジェパエックス」と発音する。



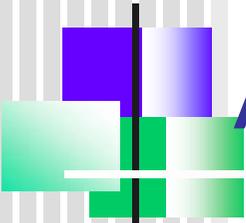
JepaXの特徴

- Free & Open
- 電子出版コンテンツとして必要な情報を持つ
 - 表紙、あとがきなど
- フォントや「行間」といったスタイルを指定する仕様は極力排除
 - 「章」「セクション」「段落」「注釈」といった、文書構造の定義を重視
- ルビ、上付文字、図、外字などにも対応

その他フォーマットとの比較

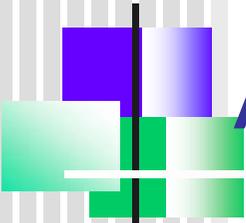
■ 検討の結果、当部会では今回JepaXを採用

	プラス	マイナス
JepaX <small>Jepa</small>	<ul style="list-style-type: none"> ■ DITA、DocBookに比べると仕様が複雑でない ■ 今までの勉強会で扱ってきており、なじみやすい 	<ul style="list-style-type: none"> ■ 1999年からバージョン0.9のまま更新が止まっている ■ 他の2つに比べると、知名度が低い。
DocBook <small>OASIS</small>	<ul style="list-style-type: none"> ■ 知名度が高い ■ OASIS標準 ■ 更新もなされている 	<ul style="list-style-type: none"> ■ DocBook 対応の文書作成ツール、スタイルシートが多く配布されている ■ エディタ(oxygen)にもスタイルシートが組み込まれている
DITA <small>OASIS</small>	<ul style="list-style-type: none"> ■ 知名度が高い ■ OASIS標準 ■ 更新もなされている 	<ul style="list-style-type: none"> ■ 仕様が膨大。 言語リファレンス: 500ページ 構造仕様: 93ページ



Apache FOPを選んだ理由

- 選択基準
 - XSL-FO仕様に準拠
 - 入手、利用がしやすい
- XSL-FO処理系 <http://w3.org/Style/XSL/2006/01/xsl11-implimentation>
 - Altsoft Xml2PDF version3.0
 - Antenna House XSL Formatter
 - Apache FOP
 - Inventive Designers Scriptura
 - PTC/Arbortext
 - RenderX XEP4

A decorative graphic consisting of a vertical black line intersected by a horizontal white line, with a purple square above and a green square below the intersection.

Apache FOPを選んだ理由

- Apache FOPの特徴
 - Apache XML Graphicsプロジェクトが開発しているXSL-FO処理系の実装
 - XSLTでFOに変換されたXML文書をPDFに変換
 - ApacheライセンスVersion2.0で公開されているオープンソース

JepaX用変換プログラム

「FANTaStiKK(ファンタスティック)」作成の進め方

- まず、最初にJepaXの仕様書を精査・解析した。

全体の仕様は下記のような「体裁単位」におおまかに分類される。

- 全体構造
- 論理単位要素
- リスト要素
- ブロック要素
- 表
- 図
- インライン要素
- 共通の属性

JepaX用XSLT作成・GUI作成

- 次に、前述の体裁単位を元に各担当者を決め、XSLT作成を振り分けた。

- 各種XSLT作成担当
- 進捗管理・テスター担当
 - プリセット3タイプ作成
(かんたん設定)
- GUI・パラメータ設定担当

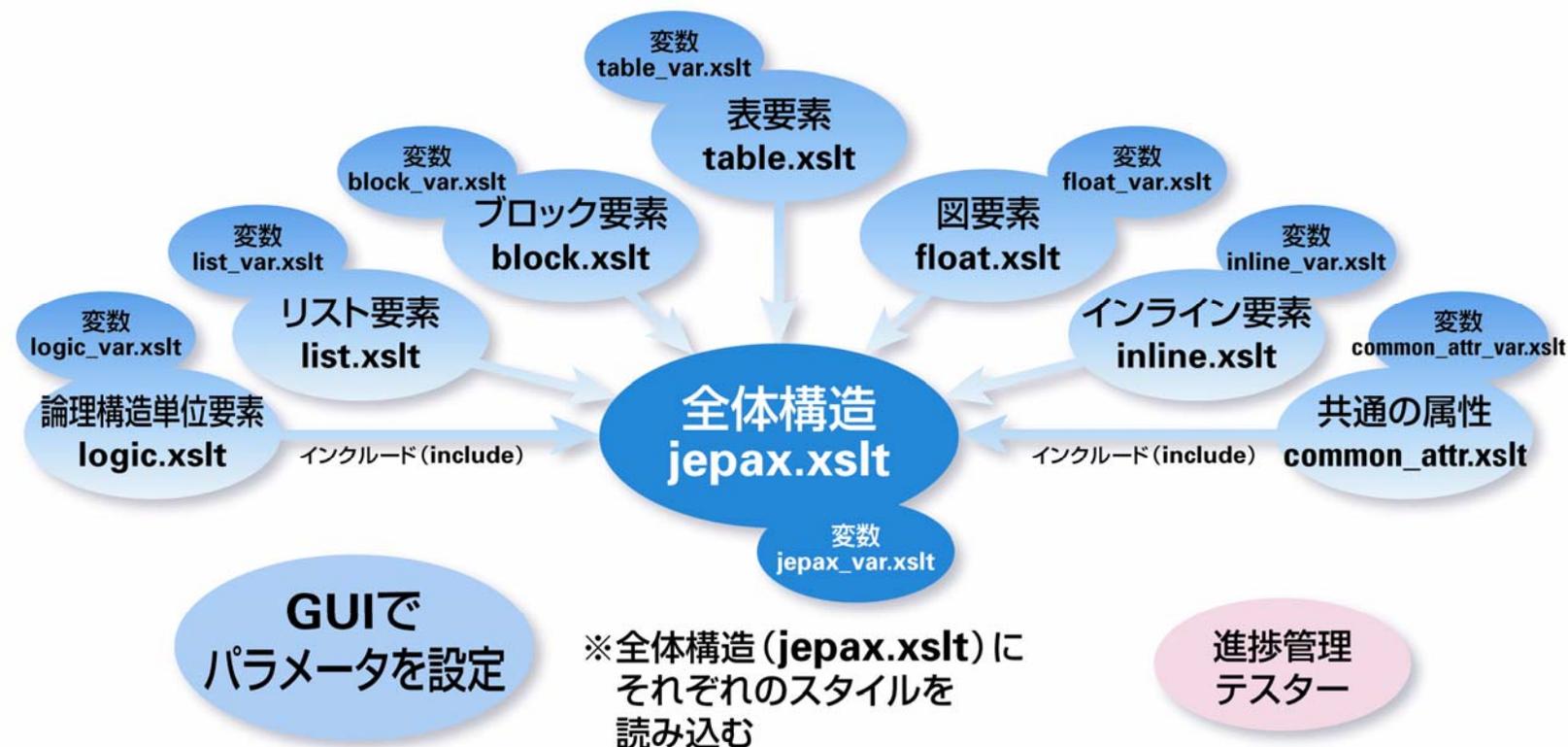
```

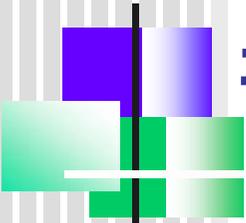
<!--ルート要素 (JepaX) にマッチー>
-<xsl:template match="jepax">
  -<fo:root language="ja" country="JP">
    -<fo:layout-master-set>
      -<fo:simple-page-master master-name="jepax_normal_spm_odd" page-height="{ $j:page_height }"
        <fo:region-body writing-mode="{ $j:writing_model }" margin-top="{ $j:region_before_after }" margin-right="{ $j:region_left_right }" column-count="{ $j:column_count }" column-gap="{ $j:column }
        <!--region-beforeなどのextentは、region-bodyのmarginに揃えた-->
        <fo:region-before extent="{ $j:region_before_after }" />
        <fo:region-after extent="{ $j:region_before_after }" />
        <fo:region-start extent="{ $j:region_left_right }" />
        <fo:region-end extent="{ $j:region_left_right }" />
      </fo:simple-page-master>
      -<fo:simple-page-master master-name="jepax_normal_spm_even" page-height="{ $j:page_height }"
        <fo:region-body writing-mode="{ $j:writing_model }" margin-top="{ $j:region_before_after }" margin-right="{ $j:region_left_right }" column-count="{ $j:column_count }" column-gap="{ $j:column }
        <!--region-beforeなどのextentは、region-bodyのmarginに揃えた-->
        <fo:region-before extent="{ $j:region_before_after }" />
        <fo:region-after extent="{ $j:region_before_after }" />
        <fo:region-start extent="{ $j:region_left_right }" />
        <fo:region-end extent="{ $j:region_left_right }" />
      </fo:simple-page-master>
    -<fo:page-sequence-master master-name="jepax_normal_psm">
      -<fo:repeatable-page-master-alternatives maximum-repeats="no-limit">
        <fo:conditional-page-master-reference master-reference="jepax_normal_spm_odd" odd-

```

XSLTファイルの構成

- 実際のファイル構成は以下のようなになる。



A decorative graphic consisting of several overlapping squares in purple, green, and white, with a black crosshair-like structure.

苦勞した点

- FOPの問題
 - 日本語での組版における不具合
 - FO の仕様の一部が未実装
- class 属性の扱い
- 設定項目数