

電子書籍制作における XML の活用

大日本印刷(株) 前川 真二*

1. はじめに

インターネット書店、プリントオンデマンドサービス、電子書籍ダウンロードサービスの登場に見られるように、インターネットの普及に伴い、出版物の流通形態が著しく変化している。特に、いわゆる「活字離れ」による出版物の売り上げ低迷、返品率増加によるコスト上昇は、出版社にとって深刻な問題となっており、在庫を抱えないプリントオンデマンドサービス、電子書籍ダウンロードサービス等新しい出版流通に対する取り組みが盛んに行なわれるようになってきている。

こうした状況の中、DNP でもこの新しい出版流通に対して、企画・製造から販売に至るまで一貫したアプローチを行なっている。(図1参照)

販売面では、書籍そのものを取り扱っている「専門書の社」(<http://www.semmonsho.ne.jp>) や、電子書籍及びオンデマンドブックを取り扱った「ウェブの書齋」(<http://www.shosai.ne.jp>) といった自社の販売サイトを構築・運営している。また、最近では、書籍だけでなく、写

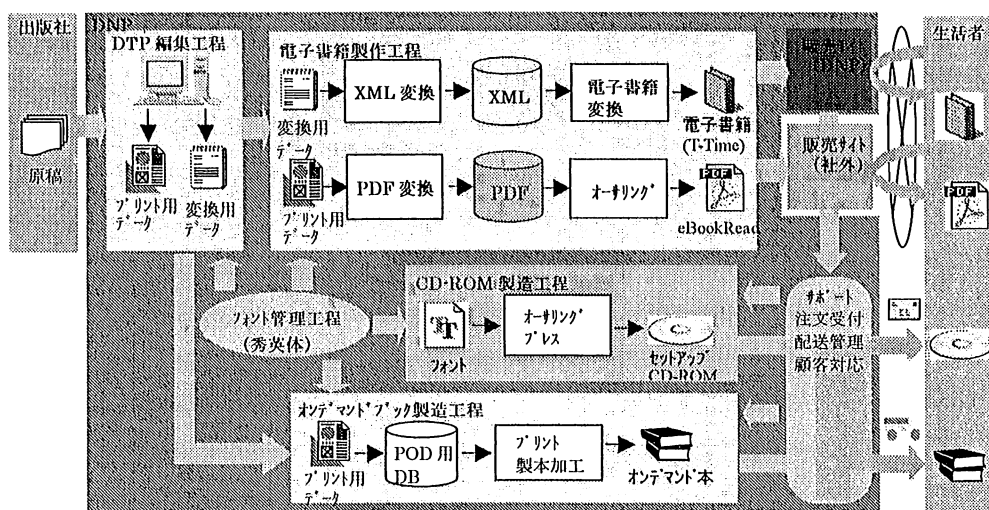
真や映像まで幅広い電子コンテンツを扱うモールサービス「AnyStyle」(<http://www.anystyle.jp>) を立ち上げている。コンテンツの制作面では、既存の資産を有効に活用しながら、オンデマンドブックや電子書籍に対応できることが理想的である。例えば文庫本のコンテンツ制作において、一般書籍と同様に、既存の DTP 工程にて編集を行い、そこからのデータ流用が可能となれば、印刷会社だけでなく校正を行う出版社にとっても大きなメリットとなる。また、電子書籍フォーマットが多様化した場合、単に DTP のデータからダイレクトに電子書籍に変換するような工程では対応ができない。こうした課題を解決する手段として、XML の活用に着目している。

しかしながら、データ管理の効率化、生産性の向上、データの再利用などの実現が期待されているものの、具体的にどう取り組んでいくかについては、まだまだ模索の段階といえる。

現状、想定される課題として、XML を核とした様々なデータ交換を可能とする工程構築や、得意先の保有する多様なデータをニーズに応じて加工・配信する等、よりフレキシ

図1.DNP の出版コンテンツ流通ビジネス全体概要

fig1. The outline of the publication contents circulation business in DNP



ブルなコンテンツ流通を実現する必要がある。今回の取り組みは、DTP から電子書籍へ変換する工程において、XML を中間フォーマットとして採用することにより、最初の実用

* C&I 事業部 IT 開発本部
〒162-0066 東京都新宿区市谷台町 6-3 市谷大東ビル
C&I Operations, IT Development Headquarters.,
6-3 Daimachi Ichigaya Shinjuku, Tokyo 162-0066

的な取り組みとして重要な意味を持つものとする。

当社では、実際にこの仕組みを構築・運用することにより、文庫系コンテンツにおいて、オンデマンドブックと電子書籍の同時進行を実現している。

本資料は、電子書籍を制作する過程でのXML採用にあたり、DTP データを活用する為の編集ルールや DTP データから XML を経由し電子書籍データへ変換するプロセス及び開発したツール類の機能等を説明し、当社の XML を活用した制作工程構築への取り組みについて、具体的に説明を行ったものである。

2. XML と印刷工程

(1) XML 化の状況

実際のコンテンツ制作部分の説明に入る前に、印刷工程に XML を活用することに対する、一般的な状況について触れておくことにする。

XML の登場以来、その話題性から急速に普及したように思われている面もあるが、実際の印刷現場において XML を活用した事例が目立つような状況には至っていない。しかしながら、ADSL 等の最新技術による低価格な高速常時接続環境の急速な普及等により、各企業のインターネットを活用した様々な情報提供ビジネスがより盛んになり、ユーザの必要とする情報を望まれる形式で提供できるようになると考えられる。そうなれば、豊富な情報を一元管理し、そこから必要な部分を取り出して、様々なスタイルに自動変換することが必須条件となってくる。このような機能を実現するにあたっては、XML の採用が有力視される。また、電子政府構想においても、文書の構造化が推進されており、世の中の様々な分野の情報が XML で管理されるようになりつつある。

このような情報の構造化の動きは、出版業界にとっても例外ではない。出版物の流通の活性化を目的に、コンテンツを紙だけでなく、様々な形態に展開することに取り組んでいる出版社にとって、XML の採用は重要な要素であり、業界の中で様々な検討が開始されている。例えば、電子出版市場の普及に備え、様々なフォーマットへ容易に交換を実現できるよう、XML をベースとした JapaX を策定し提供している日本電子出版協会の取り組みがあげられる。

こうした出版業界の動きに対して、印刷会社としても XML を中心とした制作、製造工程への取り組みの必要性が高まっている。

(2) XML 採用における課題

電子書籍制作工程に XML を採用するにあたっては、いくつかの課題が存在する。

レイアウト情報主体の既存制作工程に、文書構造情報主体の XML を持ち込むことは、オペレータの作業手

順が現状と大きく変わってしまう為、それに伴った作業負荷の軽減、人材の育成、新たな工程の構築など、様々な問題をクリアする必要がある。

また、XML に対応した編集ソフトや、エディタなど、XML を扱う環境もまだまだ不十分な状況である。

こうした状況において、全ての書籍を網羅するような取り組みも必要であるが、そのことを視野に入れつつ、まずは最適な部分に対して XML を活用した事例に取り組み、しっかりとした効果を検証するべきである。その上で拡張展開するというアプローチが重要であると考えられる。

3. 電子書籍制作フローの設計

前章で示した課題を解決した、新たな電子書籍制作フローを設計するにあたり、いきなり全ての書籍と全ての工程を XML 対応にすることは、コストや実現時間の面から考えても現実的でないと判断し、最適な部分に XML を採用し、今後拡張展開していくこととした。また、既存の DTP、印刷工程をなるべく変更することなく、低コストかつ現状の現場オペレータのスキルレベルにて電子書籍制作ができることを目的とした。

従って、書籍制作のための既存の DTP 工程をそのまま活用し、DTP からのデータを変換して電子書籍データを生成するフロー（図1参照）を設計した。その過程において、XML を中間フォーマットとして採用し、自動処理を行うところ、今回の取り組みのポイントである。

中間データに XML を採用したことにより、今後電子書籍市場が成長し、様々なビューワ（フォーマット）が登場したとしても、そのフォーマットに対応したスタイルシート（XSL）に切り替えることで多フォーマットへの対応が可能となり、その場合のコストや対応期間を低く抑えることにつながっている。

この電子書籍制作フローにより以下のようなメリットが挙げられる。

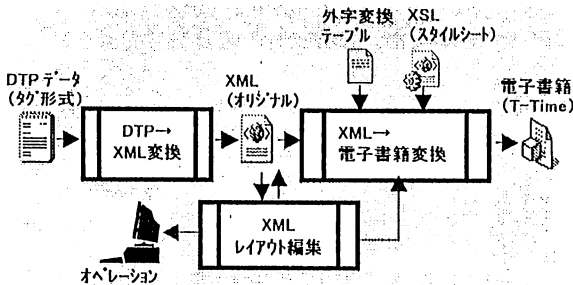
- 1) 印刷物と同じ環境（ゲラ）で得意先校正が行える
- 2) XML を自動的に生成される中間データという位置付けにすることで、オペレータは XML を意識することなく作業が行える
- 3) DTP の完全データは、そのままオンデマンドブック製造工程で活用できる
- 4) データ変換を行うプログラム以外は、特別なシステム等の導入は必要ない

4. 電子書籍変換処理

電子書籍変換処理では、DTP データから XML、電子書籍データへと変換させていくが、それぞれの持つ特性が異なるため、工程設計においては、その利点を生かすような工夫と、外字情報の取り扱いには、十分な検討を行った。

紙への印刷を目的として優れた日本語組版機能を持つDTPと、パーソナルコンピュータ上での表示と操作性を重視した電子書籍とでは、表現力と機能性に大きな格差があり、DTPで編集した全てのレイアウト情報を電子書籍に反映させること、或いは、電子書籍が要求する機能をDTP編集にて指示することは不可能である。また、XMLは文書構造を定義するものであり、レイアウトに関する情報を内包することはないが、逆にDTPや電子書籍はレイアウト主体であって構造情報を持つことはない。こうした格差を工程のどの部分でどのように埋め、効率良く制作するかが大きな課題であった。また、書籍を扱う上で外字の活用は必要不可欠であり、異なるフォーマットを経由させる過程において、いかにして外字情報を正確に伝えていくかも重要な要素である。これらを十分に検討したうえで、今回の工程設計を行っている。こうした検討をもとに、図2に示す通り、DTPデータからXMLを介して電子書籍へ変換するにあたり、2種類の変換プログラムと1種類の編集ツールを開発している。変換プログラムは入力と出力のファイル名を指定することで、自動的にフォーマット変換を行うものである。編集ツールは、自動変換では対応できない細かな修正箇所に対応する目的で開発している。

図2.DTP→電子書籍変換フロー
fig2. The conversion flow from DNP



読み込み対象としているDTPデータは、DTPが持つ自動組版機能にて使用されるトリガーコマンドと呼ばれるタグ付テキストデータである。また、中間生成するXMLは、日本電子出版協会が策定したJepaXをベースに、当社が必要な情報を付加している。JepaXは出版業界において電子出版コンテンツの蓄積や交換を行う為の交換フォーマットとして利用されることが想定されており、将来的に業界標準的なフォーマットになる可能性が考えられることや変換ツール等の充実が予想されることから、採用している。

変換対象としている電子書籍フォーマットは、現段階ではボイジャー社のT-Timeとしているが、XML→電子書籍変換プログラムにおけるXSLを切り替えることにより、様々なフォーマットへの展開を可能としている。

5. 電子書籍制作工程詳細

(1) DTPでの編集

通常の印刷物制作におけるDTPの編集は、レイアウト指示に基づき作業を進行すれば良いが、構造情報主体のXMLを生成するにあたっては、入稿原稿から構造化を検討したうえで、その結果を何らかの形でDTPデータに反映させる必要がある。

原稿から構造化を検討するにあたっては、見出し部分のスタイルと階層の深さで判断し振り分けるというルールを設定し、オペレータにリリースしている。

また、DTPデータに構造情報を反映させるには、DTPの持つ「浮動枠」という機能を活用している。これは、任意の範囲をボックス単位で区切り、そのボックスに対してレイアウトには反映されないような属性情報を指定できる機能である。前述したルールに基づき構造化検討した結果から、それぞれの構造を判別できるような属性を付加する。このような編集を行うことにより、DTPデータに構造情報を反映させることを実現している。

表1に示したのは、スタイルと浮動枠属性の関連ルールである。

また、実際のレイアウトに対する属性値の設定のイメージを図3に示す。

表1 浮動枠属性と原稿スタイルの関連

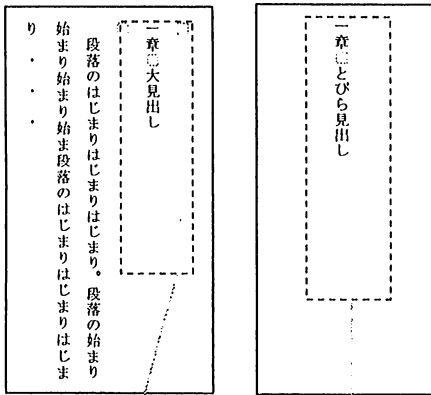
Table 1. Relation between floating frame and a layout

	DTP浮動枠属性			原稿スタイル	XMLタグ名
	属性値	表題	レベル		
1	大	見出し	1	改見見出し	改見見出し1
2	中	見出し	1	改行見出し	改行見出し1
3	小	見出し	1	改行見出し	改行見出し2
4	独立	見出し	2	改行見出し	改行見出し3
5	独立	見出し	3	改行見出し	改行見出し4
6	独立	見出し	4	改行見出し	改行見出し5
7	独立	見出し	5	改見見出し	改見見出し2
8	独立	見出し	6	改見見出し	改見見出し3
9	独立	見出し	7	改見見出し	改見見出し4
10	独立	見出し	8	見出し予備	null
11	独立	見出し	9	見出し予備	null
12	独立	その他	1	扉見出し	扉見出し1
13	独立	その他	5	扉見出し	扉見出し2
14	独立	その他	6	扉見出し	扉見出し3
15	独立	その他	7	前書き等	独立文章枠
16	独立	その他	8	通常文章	文章枠
17	独立	その他	9	余白	余白枠

※レベルが文書階層の深さを示す

図3.浮動枠属性設定例

fig3. The example of an attribute setting of a floating frame



本文中ははじめに出現する「大見出し」なので、変換テーブルNo.1, No.7 ~ No.9の「改頁見出し」を使用する
とびら見出しなので、変換テーブルNo.12 ~ No.14の「とびら見出し」を使用する

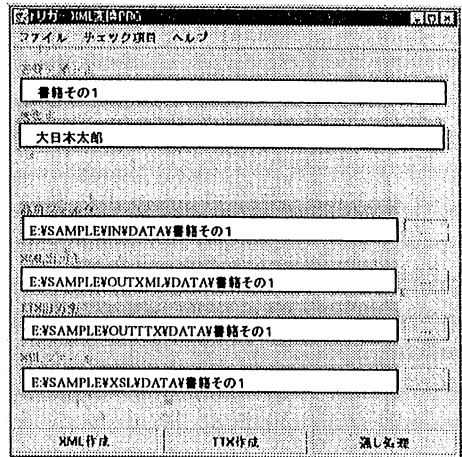
DTP 編集における入力ルールは、構造化（浮動枠）だけでなく、欧文、ルビ、連数字などの各種組版に対しても設定している。例えばルビは、親文字との文字列長との関係、親文字の位置（行頭、行末など）など様々な条件によって組版規則（JIS X 4051）が定められている。更に出版社によっても、より細かな規則が定められている場合もある。こういった細かな規則は DTP では表現できるが電子書籍では表現できない。また DTP では、見た目に基づき、スペース等を使用しながら微妙な調整を行っているが、出力されたデータをプログラム処理する際には、それらスペース等の意味を解釈することは不可能であり、結果的には崩れたデータを生成してしまうことにつながってしまう。こうしたことを回避するために、DTP での編集に対して、ルビであれグループルビは使用せずモノルビとし、スペース等による調整は行わないなど、ある程度制限を設定し運用を行っている。こうしたルールを DTP 編集現場にリリースすることで、その後のプログラムによる自動処理を可能としている。

(2) プログラムによる自動変換

DTP→XML→電子書籍への変換は、図4に示すようなメニューに対して、入力する DTP と出力する電子書籍のファイル名を指定するだけで自動に行えるようなプログラムを開発している。基本的には、中間に生成される XML を意識することなく、DTP から直接電子書籍に変換されるイメージであるが、内部の処理においては、必ず XML を生成したうえで電子書籍に変換を行っている。

図4. 自動変換メニュー画面

fig4. The menu of an a conversion program

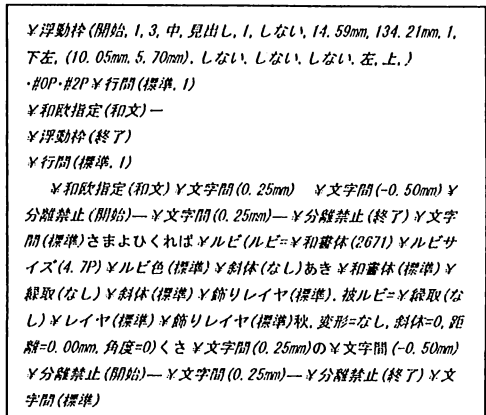


(2) - 1. DTP から XML への変換

DTP より出力されるトリガーとは、図5に示すように、様々な組版や制御に関するコマンドタグを含んだテキストファイルである。1つのコマンドタグの中に、そのコマンドの条件を示す要素が設定されており、コマンドの機能によっては、複数の要素が設定されているものもある。

図5. トリガー例

fig5. An example of the trigger command outputted from DTP



プログラムでは、このファイルを読み込み、組版や制御に関する情報部分のテキスト処理にて XML のタグ付けを行っている。おおまかな処理のロジックは、以下のとおりである。

- ①単一のコマンドタグ毎へ分解する
- ②単一のコマンドタグの要素を取り出す
- ③①~②の結果を解析シタグ形式変換を実行する
- ④①~③までの処理を繰り返し、全体のタグ変換を実行して一時的なファイルを作成する（構造情報は反映されておらずフラットな状態）
- ⑤構造情報を示すタグを解析し、全体の構造マップを作成する
- ⑥構造マップに従い一時ファイルのタグ変換を実行し、構造情報を反映させて最終的な XML を出力する

このような処理の結果、図6に示すような XML ファイルを作成している。

本来 XML は構造情報を定義するものであるが、この処理の過程においては、DTP で編集されたレイアウト情報を電子書籍に反映させる必要がある。そこで、構造情報以外にレイアウトに関する情報を定義するタグを設定し、DTP でのレイアウト情報を格納するようにしている。この XML の内容については、後の項で説明する。

図 6. XML 例
fig6. The example of XML

```
<?xml version="1.0" encoding="Shift_JIS" ><jepax>
<bookinfo>
<book-title>海燕2</book-title><book-author>小島政二郎
</book-author></bookinfo>
<front></front>
<body>
<div class="改頁見出し1" id="body_00001">
<head>
<title>
<style br_before="2" h_level="H2" br_after="2" index="false"
pb_before="true"></style>
その<ruby><rt>ひ</rt></rt></ruby>
</title>
</head>
<div class="改行見出し1" id="body_00002">
<head>
<title>
<style br_before="1" h_level="H3" index="false"
pb_before="false"></style>
—
</title>
</head>
<p> —さまよひくれば<ruby><rt>あき</rt></rb><rb>秋
</rb></ruby><く</rb></ruby></p>
```

(2) - 2. 外字への対応

処理の過程において、外字についても考慮が必要なポイントである。トリガーファイル内の外字は、DTP 独自の書体及び文字コードで定義されている。電子書籍側の外字は、T-Time であれば PC 上の書体名とシフト JIS コードにて定義されているが、今後登場が予想されるビューワについては必ずしもシフト JIS とは限らない。また外字は、品目毎に新たな文字が必要となり、それに伴い追加作成されており、追加外字についても変換処理に確実に反映させるよう、管理・運用を行わなければならない。当社では、文字管理部門にて

外字の作成と一元管理（独自のユニークなコード体系）が行われ、DTP 編集工程、オンデマンドブック製造工程、電子書籍閲覧ユーザに提供されており、外字を扱っていくうえでは文字管理部門との連携が必要不可欠である。こうした点から、XML 内部での外字は CTS コード体系にて表現し、各ビューワの文字体系に併せた変換テーブルを用意することで、随時対応していく体制としている。

(2) - 3. XML から電子書籍への変換

図7に例を示すように、電子書籍 T-Time のデータは HTML に近いタグ付きテキスト形式の仕様となっている。

XML から電子書籍への変換は、XSLT に基づいたプログラムを開発して行っている。おおまかな処理のロジックは以下の通りである。

- ①XSLT 変換にて XML から T-Time タグへ変換する
- ②特定文字列の置換処理を行う
- ③外字コード変換処理を行う（CTS→シフト JIS）
- ④T-Time ソース整形処理（改行挿入）
- ⑤特定置換処理（欧文アクセント対応）

XSLT に関しては、XML 技術を利用した Apache 関連ソフトウェアの開発プロジェクトである Apache XML Project から提供されている Xalan を活用し、開発負荷の軽減をはかっている。

図 7. 電子書籍データ例 (T-Time)
fig7. The example of digital book data(T-Time)

```
<HTML>
<HEAD>
<TITLE>海燕2</TITLE>
<T-TIME charspace=0 homnonsize=16 linespace=12 indent=0
tatoyoko=2 midashisize=18 stroke=vertical start=0L
hashiravisible=true></T-TIME></HEAD>
<!------->
<BODY valign=top>
<P></P>
<!------->
<T-PB valign=middle>
<H1>海燕2</H1>
<T-PB valign=top>
<!------->
<H2>H次</H2>
<T-PB valign=top>
<T-PB valign=top hashiravisible=true>
<BR>
<H2>その<T-R>H (ひ) </H2>
<BR>
<H3>— </H3>
<BR>
—さまよひくれば<T-R>秋 (あき) くさの—<BR>
フラ<T-CODE src="glfgaiji/22227222.gif"><T-CODE
src="glfgaiji/22227223.gif">と、<T-R>三 (み) <T-R>干 (ち) <T-R>
代 (よ) は<T-R>海 (うん) <T-R>野 (の) の<T-R>家 (うち) の<T-R>
前 (まへ) に<T-R>来 (き) てゐた。<BR>
```

(3) XMLでのスタイル編集

自動変換で生成された電子書籍のスタイルに関して編集や修正を加える場合、専用のエディタ等がない為に、テキストエディタ等により、電子書籍のフォーマットを理解したうえで行う必要があり、オペレータにそのスキルが要求される。また、章や節単位に同じレイアウトがあっても、電子書籍上では文字列単位でしか扱えない為、作業負荷や修正ミスの可能性が高くなる。こうしたことに対応する目的で、変換の途中に生成されるXMLに対して、XMLタグを意識することなくスタイル編集が行えるツールを開発した。

このツールの特徴は以下の通りである。

- ①XMLの構造情報をツリー表示する (図8参照)
- ②ツリー表示された構造タグを指示することで、その文章内容が確認できる (図8参照)
- ③各階層の意味にあった編集メニューが自動表示される (図9参照)
- ④同階層に対して、共通・個別のどちらの単位でもスタイル指示が行える (図9参照)
- ⑤スタイルシートを追加することにより、様々なフォーマットへの変換が可能となる

このようなツールを開発しリリースすることで、XML及び電子書籍に関しての知識が少ないDTPオペレータでもXMLを介した電子書籍編集作業を可能としている。

図8. 文書構造ツリー表示

fig8. The tree display of document structure

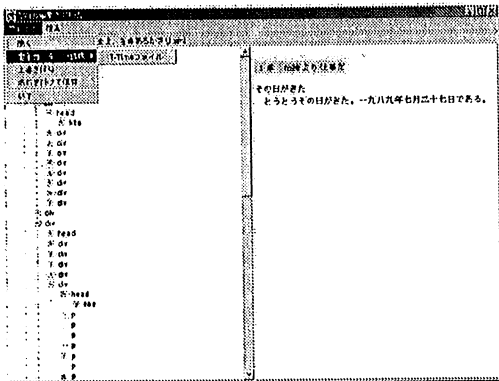


図9. 各編集メニュー例

fig9. The difference in the menu by structure

■ JepaX 階層

- 目次
 - 目次表示に関する内容等
- ページスタイル
 - 組み方向など全体スタイルに関する内容等

■ title 階層

- 改行
 - 見出しの直前直後の改行や改頁の設定等
- 目次・柱
 - タイトルを目次に表示した場合の設定

- 同じパスに同じスタイルを設定
 - ・ チェックボックスをチェックすることで同一階層に同じスタイルが設定される
 - ・ チェックしなければ個別の設定が行われる

(4) XMLフォーマットについて

これまでの中でも説明してきたが、DTPから電子書籍に変換する過程において生成しているXMLは、JepaXをベースに、スタイルに関する独自タグを加えた仕様としており、スタイル設定が必要な構造タグの直後に、

```
<style 要素1 要素2 要素3・・・></style>
```

といったように、<style>というタグを挿入することで、その構造に対して、「要素」の部分に指定した値のレイアウトが設定できるようになっている。

内部にスタイルに関する情報を設定することは、本来構造情報(XML)とスタイル情報(XSL)を分けて管理するといったXMLの定義からは外れてしまうものであるが、電子書籍への変換の効率化を最優先に考え、最低限必要なスタイル情報を把握可能な状態で残す方法としている。仮に、出版社等に純粋なJepaXの提供を望まれた場合、XSLT変換によりスタイル情報を取り外すことにより提供することが可能である。

このように、XMLの交換フォーマットとしての利点を十分に活かし、従来工程のフローをできるだけ変えることなく効率的に生産が行えることを優先し、スタイルを残したXMLを採用するに至っている。

なお、表2は<style>に関する要素の一覧、図10は<style>の指定例、である。

表2. style 要素一覧
Table2. A list of the element in a styletag

設定項目	属性名		
	jepax	Title	p
フォントサイズ(本文)	honmonsize		
フォントサイズ(見出し)		Xsize	xsize
行間	linespace		
字間	charspace		
本文フォント			fontname
見出しフォント	fontname	fontname	
インデント	indent		
先頭文字		top_string	top_string
見出しインデント		br_flag1	br_flag1
行揃え		align	align
前の改行数		br_before	
後の改行数		br_after	
前改頁		pb_before	
前改頁の位置		pb_position	pb_position
後改頁		pb_after	
柱レベル		h_level	
段落前制御			p_control
目次行揃え	valign		
目次出力(全体)	index		
目次出力(見出し)		index	
目次のインデント		index_indent	
目次のフォントサイズ		index_xsize	
目次の後の改行数		br_after_index	
柱表示	hashiravisible		
組み方向	stroke		
縦中横有効文字数	tateyoko		

図10. XMLにおける style タグ使用例
fig10. The example of style tag use in XML

```

<jepax>
<body>
<div id="PBtitle1_1">
  <head>
  <title>
    <style title_indent="2" index_inde="3" hyouji="true">
    </style>
    XML技術
  </title>
  </head>
</div id="PBtitle1_2">
  <head>
  <title>
    <style title_indent="3" index_inde="1" hyouji="false">
    </style>
    XMLとは?
  </title>
  </head>
</div id="PBtitle1_3">
  <head>
  <title>
    <style br_before="2" h_level="H2" br_after="2"
    index="false" pb_before="true">
    </style>
    XMLって? </title>
  </head>

```

6. 今後の課題

これまで述べてきた内容は、電子書籍制作の一部において、従来の印刷物製造工程からXMLを介してデータ変換を実施するといった取り組み事例である。

今後は、初めにXMLを作成・管理してから、印刷物も含め様々なデータ展開を行うフローの実現が重要な課題であると考えている。こうしたことが実現すれば、大量の情報の効率活用(串刺し検索等)や2次の展開が実現し、付加価値の高いサービス提供を可能とすると共に、製造工程コストの大幅な低減につながるものと予想される。

しかしながら、全ての書籍や印刷物を対象としてXMLありきの工程を実現するにあたっては、全体を確実に網羅できる標準的な構造化フォーマットの設定やそれを制作・管理・編集するためのツール(XMLエディタ、XMLデータベース、DTP等)の開発が必要となり、実現へのハードルは高いものがある。また、書籍や印刷物の全てのデータが構造化されることは理想的ではあるが、書籍の種類によっては、構造化の効果が望めるものばかりとは言いきれない。

よって、現段階においては、目的やコストに応じて、例えばJepaXのように、特定の目的に対して開発されているフォーマットを活用し、それに対して最小限の手を加えることで実現しておき、必要に応じてプレーンな状態或いは標準的なフォーマットに移行することも可能である、という道筋をつけておくことが次善の策であると考えている。

但し、こうしたアプローチはあくまでも「標準」というグローバルな拠り所があつての取り組み方であり、当社としても、標準的なXMLフォーマットの開発を前提とした印刷工程の構築に取り組んでいく必要があると思われる。

文 献

- (1) 日本印刷出版協会出版データフォーマット標準化研究委員会:「JepaX - JEPA 電子出版交換フォーマット -」, Ver0.9, 全44頁
- (2) 網モリサワ:「印刷とXML2002」, Ver1.20, 全14頁
- (3) 日本工業標準調査会 審議、日本企画協会 発行:「日本語文書の組版方法 JIS X 4051」, 全69頁