# XML スタイルデータ変換ソフトウェアの開発と印刷業務への適用

奥田　篤，堀　武司，波　通隆

# Development of XML Style Data Conversion Software, and application to Printing Process.

Atsushi OKUDA, Takeshi HORI, Michitaka NAMI

**抄　録**

近年，印刷物と共に電子データの納入を求められることが増えているため，印刷業界は，紙媒体と電子媒体に同時に対応できる工程の整備を必要としている。汎用性・可搬性・可読性に優れた XML が注目されているが，スタイルデータを既存の組版システムで処理できる形式に変換することが難しく，既存印刷工程と XML を利用する文書処理プロセス（XML フレームワーク）の融合が進んでいない。

そこで，XML 化されたデータにスタイルを与える XSL-FO スタイル指示データを，既存組版システムが処理できるデータ形式へ変換するソフトウェアを開発した。さらに，変換ソフトウェアを既存印刷工程へ適用して，ワンソースマルチユースを実現した。

**キーワード**：XML，XSL-FO，組版システム，変換ソフトウェア，ワンソースマルチユース，印刷工程

**Abstruct**

Since it is increasing that printing companies receive orders of electronic data together with printed matter from their customers, the printing industry just needs to construct processes which can produce products as paper medium and as electronic media simultaneously. XML excellent in pliability, portability, and readability is expected as key technology for building such processes. But because it is difficult to change XML style data into forms which can be processed by existing typesetting systems, it is not progressing to fuse document processing processes (XML frameworks) which use XML technology and existing printing processes.

Then, a software that converts XSL-FO data into data forms which typesetting systems can operate was developed. And an one-source-multi-use printing process was built using the conversion software.

**KEYWORDS**：XML, XSL-FO, Typesetting System, Conversion Software, One-Source-Multi-Use, Printing Process

## 1．はじめに

近年，コンピュータネットワークの普及や情報技術の著しい発達に伴い，これまでのように文書を紙に印刷して利用するだけでなく，電子化・データベース化してネットワーク上で利用することが，広く求められるようになってきた。

このような状況により，印刷物作製の実務を担っている印

課題名：印刷業務に於ける文書記述言語の応用に関する研究
（一般研究：14〜15 年度）

刷業界では，顧客から，印刷物に加えて電子化したデータの納入を求められる事例が増えている。そのため，印刷業界は，従来から作製してきた印刷物の他に，様々な電子データを作製する必要に迫られており，既存の印刷工程とは別に，電子データを作製する工程を，データ形式などに応じて幾つも構築して，顧客ニーズに対応している。

しかし，工程を複数構築して維持することは負担が大きい上に，求められる作製物の多様化に伴って工数が増えてしまう。従って，これらの工程を統合して簡素化するために，同一のソース（素材）から複数の作製物を自動かつ多重に生成できる工程（ワンソースマルチユースの工程）を構築することが業界全体の課題になっている[1−3]。

異なる媒体を対象とした工程を統合して，ワンソースマルチユースを実現するには，工程に依らずに利用でき，コンピュータ処理に適したソースデータと，その処理技術・システムが必要になる。ソースデータの記述言語として，汎用性・可搬性・可読性に優れた構造化文書記述言語である Extensible Makeup Language（XML）[4] が期待されており，印刷業界では XML の利用に関する技術の開発や工程の整備を進めている[1−3]。道内中小印刷業者も早急にソフトウェア技術の蓄積を進め，そのような技術開発に取り組む必要がある。

このような状況を踏まえて，当場は印刷業務における XML の利用に関する技術開発を進めてきた。その一環として，XML データから生成したスタイル指示データを，組版システムが処理できるデータ形式へ変換する，XML スタイルデータ変換ソフトウェアを開発した。さらに，変換ソフトウェアを既存印刷工程へ適用して，XML を基盤とするワンソースマルチユースの実現を試みた。本報告では，開発した変換ソフトウェアと印刷工程への適用事例について紹介する。

## 2．変換ソフトウェアの開発

### 2.1　印刷工程の概略および XML の適用とその問題点

#### 2.1.1　基本的な印刷工程

広義の印刷工程は，基本的に，テキスト・図版・画像などの要素データの作製・加工から，これらをレイアウトして得られる紙面のイメージを保持する刷版を作製するまでの印刷前工程，刷版上の紙面イメージを紙に刷る狭義の印刷工程，刷り上がった紙を裁断・製本する印刷後工程からなっている。

印刷前工程は，要素データを組み合わせて紙面をレイアウトして組版データ（版下）を作製する組版，組版データを紙面イメージにレンダリング（可視化）して製版フィルムに転写するフィルム出力，製版フィルム上の紙面イメージを刷版へ転写する製版の，各処理から構成されており，組版は組版システムで，フィルム出力はイメージセッターで，製版は製版機で，それぞれ行われる（図 1）。印刷工程は，印刷前工程で作製された刷版を用いて印刷機で行われる。

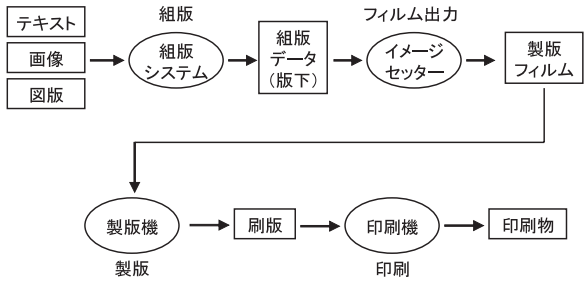


**図 1　一般的な印刷工程**

#### 2.1.2　印刷工程への XML の適用と問題点

このような印刷工程では，一般的に，HTML（Hyper Text Markup Language）ファイルや XML ファイルの作製やデータベースの構築のような，電子データの作製は行えない。そのため，電子データを作製する必要がある場合は，別途工程を構築することになるが，複数の工程の構築・維持に要する負担も大きく，工程の統合・簡素化が望まれている。

そこで，大手印刷企業では，XML 関連技術を利用する文書処理プロセス（XML フレームワーク）の導入により，多重化した工程を統合してワンソースマルチユース化することが試みられている。

XML データは，文書の論理構造をデータ型（スキーマ）として表現するために予め定義されたタグセット（スキーマ言語）を使用して，データ型を満足するようにタグ付け（マークアップ）されており，個々の構成要素や要素間の関係について機械可読である。XML フレームワークは，このような XML データの性質に着目して，要素の位置の変更による構造変換や要素自体の変更などをソフトウェアで行うことで，XML データをソースとするワンソースマルチユースを図る。即ち，特定のスキーマの XML データを所定のデータ形式に変換するソフトウェアを，所望のデータ形式に応じて必要数用意することでフレームワークを構成する。一度フレームワークが構成されると，同一のスキーマを持つ XML データを自動的に各々のデータ形式に変換することが可能になる。

XML データを処理するソフトウェアは，機能・変換出力などの要求仕様を満足するように，フレームワークごとに個々開発されることが多いが，その汎用化を目的に変換指示言語 XSLT（XSL (Extensible Stylesheet Language) Transformations）[5] の標準化と処理ソフトウェア（XSLT プロセッサ）の実装が進んでおり，これを利用することも可能である。XSLT プロセッサを用いる，一般的な，XML フレームワークを図 2 に示す。

このフレームワークでは，ソースとなる XML データを，所望の出力が得られるように XSLT で記述された変換指示データ（スタイルシート）に従って，XSLT プロセッサで様々な形式に変換処理する。HTML データや XML データなどの電子データは，XSLT プロセッサによる電子的な処

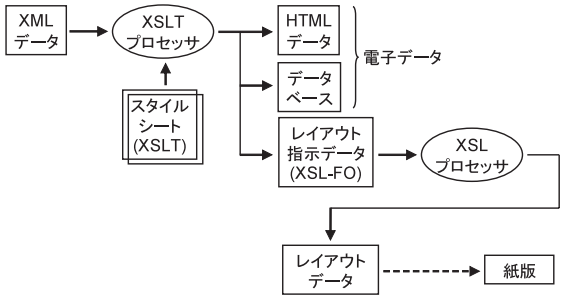


図2　XML 関連技術を利用する文書処理プロセス

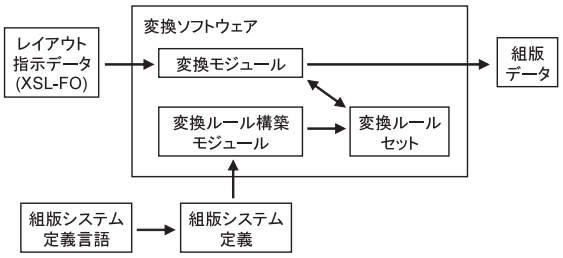


図3　変換ソフトウェアの構成

理で得られる。印刷用版下を得るためには，レイアウトした紙面イメージをレンダリングする必要があるので，XML データにレイアウトを与えることを目的に標準化されているスタイル指示言語 XSL-FO（XSL Formatting Objects）[6] を処理するソフトウェア（XSL プロセッサ）が利用される。XSL プロセッサは，XSLT プロセッサで FO 変換されたスタイル指示データ（FO データ）を紙面イメージにレンダリングする。得られた紙面イメージを紙にプリントアウトし，これを紙版として利用することで印刷物が作製できる。

このような XML フレームワークを構築することで，工程のワンソースマルチユース化が実現できるが，実際には，紙版は多面付けできないため，既存の印刷工程では利用できない場合が少なくない。そこで，多面付けした版下を得るために，スタイル指示データをさらにソフトウェア的に処理して既存組版システムに適用することが，試みられている。そのためには，スタイル指示データを組版システムに固有のデータ形式に変換するソフトウェアの開発が必要になるが，ソフトウェア開発技術の蓄積が少ない道内中小印刷企業では，取り組むことが難しいのが現状である。そこで，組版システムに依らずに利用できるスタイル指示データ変換ソフトウェアを開発した。その詳細について以下に述べる。

## 2.2　変換ソフトウェアの概要

開発した変換ソフトウェアは，XSL-FO で記述されたレイアウト指示データを，変換対象に指定された組版システムに固有のデータ形式へ変換する。

変換ソフトウェアは，Java 言語で実装されており，ユーザインタフェースにはコマンドラインインタフェースが与えられている。

変換ソフトウェアは，変換処理に先だって，対象とする組版システムに応じて，変換ルールセットを構築する変換ルール構築モジュールと，構築されたルールセットに従って，変換を行う変換モジュールから構成されている（図 3）。また，変換ルール構築モジュールの動作を動的に制御する組版システム定義と，その組版システム定義を記述するために用いられる組版システム定義記述言語も，このソフトウェアの重要な構成要素である。

## 2.3　変換ルール構築モジュール

組版システム定義記述言語は，XML 仕様に従って派生語彙として設計されたスキーマ言語であり，DTD（Document Type Definition）（文書型定義）として実装されており（図 4），組版システム定義を記述するために用いられる。

組版システム定義は，変換の対象とする組版システムについて，システムの組版コマンドセットの各コマンドと XSL で定義されている FO の対応関係を記述する。記述例を図 5 に示す。記述は組版システム定義言語を用いて行う。

変換ルール構築モジュールは，組版システム定義を読み込み，変換モジュールが利用する変換ルールセットを構築する。

変換ルール構築モジュールで組版システムに応じて変換ルールを構築することにより，変換モジュールでは対象組版システムが抽象化されるので，このソフトウェアは対象組版システムに依らず汎用的に利用できる。

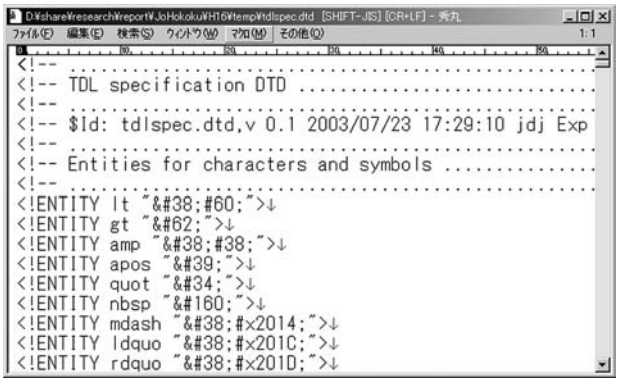


図4　組版システム定義言語の文書型定義

## 2.4　変換モジュール

変換モジュールは，変換ルールセットに従って，XSL で記述されたスタイル指示データを，対象とする組版システムが処理できる組版データへ変換して出力する。

変換モジュールは，XSL 基本仕様で定められた以下のカテゴリーに含まれる FO を適切に処理できるように実装されている。

- 行内要素
- 段落要素

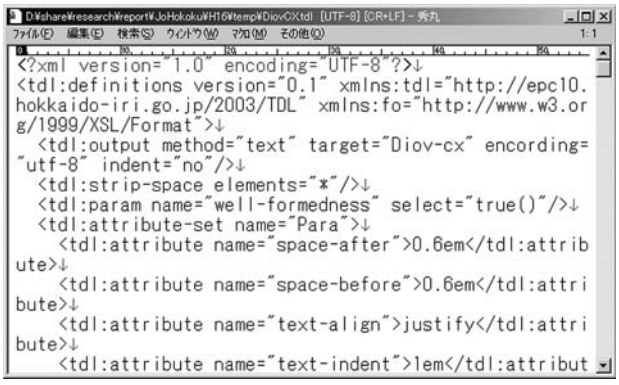


図5　組版システム定義の例（Diov-CX 用）

・ページ指定要素

変換モジュールの動作フローを図 6 に，変換挙動の模式図を図 7 に示す。変換モジュールが起動されると，指定されたレイアウト指示データを読み込み，構文解析を行って FO やテキストを要素として分離し，これらを木構造に配置したソース側ドキュメントオブジェクトモデルを生成してメモリ上に展開する。次いで，ドキュメントオブジェクトモデルの各要素に探索順位を付けて，ルート要素から始まる探索経路を設定する。探索順位の探索経路に従って要素をたどりながら，各要素ごとに変換ルールセットから対応する変換ルールを検索して，変換操作を決定する。変換操作を正しく決定するために，必要な場合は，限度無く前後の要素を先読みないし後戻りすることを許している。決定された変換操作に従って，対応する組版コマンドやテキストをメモリ上に生成し，これを要素とする結果側ドキュメントオブジェクトモデルを木構造に組み立てて行く。この際，必ずしも木構造の中に配置できない要素も有り得るが，ダミー要素を親要素として生成して，木構造中に位置づける。探索経路を終端までたどりきって，ソース側ドキュメントオブジェクトモデル中の全ての要素に対して変換操作が実行された後，組み立てられた結果側ドキュメントオブジェクトモデルの要素を順次組版データとしてファイルに出力して行く。全ての要素を出力し終わると，変換モジュールは実行を停止する。

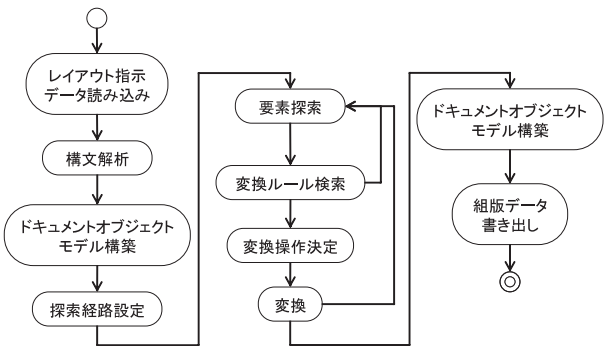


図6　変換モジュールの動作フロー

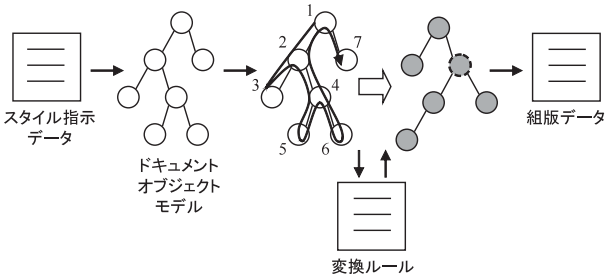


図7　変換操作の模式図

## 2.5　変換ソフトウェアの評価

Diov-CX（スパイシーソフト社）を対象組版システムとして，変換ソフトウェアの評価を行った。

Diov-CX は，電算写植の流れを組むコマンド編集と WYSIWYG 編集の二通りの編集方法の共存を実現した組版システムである。コマンドベースの情報処理機能に優れていることから，頁物印刷を主体としている中小印刷企業で広く採用されている組版システムである。

Diov-CX で作製された既存組版データからレンダリングされる紙面イメージを元に，相当する紙面イメージをレイアウトできるスタイル指示データを XML フレームワークで作製する。このスタイル指示データを変換ソフトウェアで変換して得られた組版データと，既存組版データとを比較した。

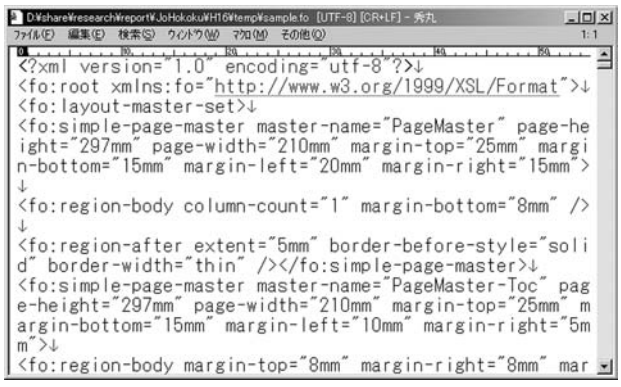


図8　試験に用いたスタイル指示データの例

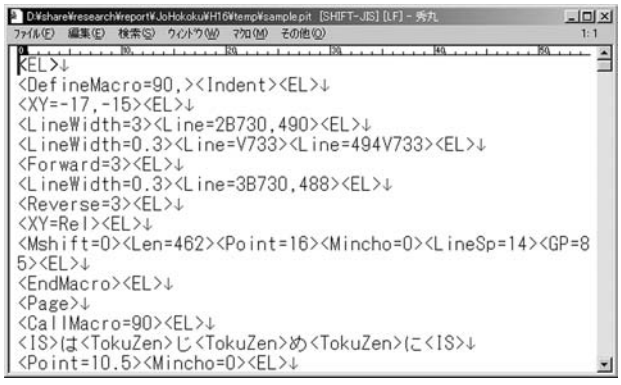


図9　変換出力された組版データの例