社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－DD－46　(1)
2004／9／24

表層表現抽出と文書構造解析に基づく XML 文書変換システム

布目光生[†]　石谷康人[†]　住田一男[†]

本論文では，法令集，官報，約款集，規定集，論文，名刺などの既存文書を応用規格に基づいた XML 文書に自動変換する新しい文書変換システムを提案する．本システムは，表層表現抽出処理，構造詳細化処理，整合性獲得処理の 3 つの機能で構成されている．本システムでは，まず，表層表現抽出により，入力文書から見出し語やキーワードなどの表層表現を自動抽出すると共に，表層表現を手がかりとして文書要素に対する柔軟なタグ付けを行う．次に，文書要素へのタグ付け結果に対して構造詳細化処理を適用することにより，応用規格にしたがった文書構造の複雑化をボトムアップに実施する．そして，整合性獲得処理により，部分構造の並べ替えや不要な文書要素の削除などを行うことにより応用規格に基づいた高品位な XML 文書を自動生成する．実験では，実際の業務で利用されている文書を特定の応用規格に基づいた XML 文書に変換すると共に，変換精度ならびに変換作業時間を計測して提案システムの有効性を評価した．

XML Document Transformation System Based on Information Extraction and Document Structure Analysis

Kosei FUME,[†] Yasuto ISHITANI[†] and Kazuo SUMITA[†]

A new method for document transformation is proposed in this paper as the basis for a document processing system which can convert various existing documents into XML documents. The proposed method consists of information extraction, document structure analysis, and document structure modification. Firstly, keywords or specific portions are detected from an input document by the information extraction process. Secondly, document elements such as words, phrases, sentences, or paragraphs are extracted and tagged according to the information extraction results. Thirdly, the hierarchical structure of document elements is constructed by the document structure analysis process. Finally, this document structure is modified and converted into an XML document in accordance with a specified DTD (Document Type Definition) by the document structure modification process. Experimental results show the method is effective in transforming existing documents to various XML documents.

1. はじめに

近年，XML(Extensible Markup Language) テクノロジーは電子商取引，サプライチェーンマネジメント，電子政府，電子書籍/出版，ナレッジマネジメントなどの分野で重要な基盤技術となっている．その結果，文書の有効活用と文書処理の自動化を目的として，DTD(Document Type Definition:文書型定義) や XML Schema で定義された応用規格が様々な業界や分野で策定されるようになった．しかし，紙文書を含む既存の文書を XML 応用規格に基づいて XML 化する場合には膨大なコストが必要とされている．

† 株式会社東芝 研究開発センター 知識メディアラボラトリー
〒 212-8582 川崎市幸区小向東芝町 1
Knowledge Media Laboratory, Corporate Research & Development Center, Toshiba Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582

最近では，文書内容へのアクセスが頻繁に生じる論文，官報，約款集，規定集，法令集，マニュアルなどを特定の応用規格に基づいた XML 文書（以後，ターゲット XML 文書と呼ぶ）に変換することに対するニーズが高い．このような文書は紙文書，プレーンテキスト，ワープロ文書，PDF 文書など様々なフォーマットで記述されている．このような文書を対象とした従来の XML 文書変換作業は次の工程で構成されている．

ステップ 1： OCR(Optical Character Reader) やテキスト抽出ツールを用いた文書のプレーンテキスト化

ステップ 2： ワープロやエディタを用いたプレーンテキストに対する手動タグ付け

ステップ 3： オートタガーによる文書構造化（構造タグの付与）

ステップ 4： XML エディタによる構造化文書の整形

ステップ 1 では，紙文書を対象とした場合には，OCR

処理もしくはキーボードを介した手入力により紙文書の内容をプレーンテキスト化する必要がある．ワープロ文書や PDF 文書のようなアプリケーションに依存するフォーマットを対象とした場合には，テキスト抽出ツールを用いて文書内容をプレーンテキスト化する必要がある．紙文書のプレーンテキスト化では，OCR 誤りの修正やキーボード入力が煩雑であるという問題点がある．

ステップ 2 では，ワープロやエディタを用いて，プレーンテキスト化された文書における語，フレーズ，センテンス，パラグラフなどの文書要素に対して手作業でタグ付けを行う．この場合，ワープロのスタイル付け機能を利用する場合には，あらかじめタグセットを登録しておく必要がある．また，タグ付け作業のためのガイドラインを定義しておく必要がある．

ステップ 3 では，ステップ 2 で生成したタグ付け結果に対して市販のオートタガーや XSLT(XSL Transformations) を適用することにより文書構造化を行い，応用規格に基づいた XML 文書に変換する．この場合，文書種別と応用規格の組み合わせごとにスクリプト言語を用いたプログム作成作業が必要となる．また，ターゲット XML 文書の文書構造が入力文書の文書構造より大幅に複雑な場合には，オートタガーでは対応できないことがある．

オートタガーによるタグ付け処理により正しいターゲット XML 文書が得られない場合には，ステップ 4 において，XML エディタを用いたタグ付け結果の編集作業が必要となる．この場合，オペレータには，XML 文法に関する知識，文書内容に関する知識，応用規格に関する知識などの様々な専門的知識が必要とされる．

以上のように，従来の XML 文書変換作業は複数の繁雑な作業工程で構成されており，膨大なコストを必要とするものであった．また，作業にあたるオペレータは専門的な知識が必要とされていた．そこで本論文では，これらの問題点の解消を可能とする，紙文書，プレーンテキスト，XHTML 文書，XML 文書を応用規格に基づいた XML 文書に自動変換する新しい方式を提案する．

上述した XML 文書変換工程を外観すると，XML 文書変換を次の 3 つの機能に分解することができる．

(1)　文書中の文書要素に対してタグを付与する機能
(2)　タグが付与された文書要素を構造化する機能
(3)　応用規格に基づいて構造化文書を整形する機能

提案方式では，これら (1)〜(3) の機能をそれぞれ自動化することにより，上述した XML 変換作業のコストを大幅に削減する技術を実現することを目指す．提案方式では，上記 (1) の機能に対して，パターンマッチング技術に基づく表層表現抽出と，表層表現に対する柔軟なタグ付け処理を組み合わせることにより，文書要素に対する柔軟なタグ付け機能を実現する．上記 (2) の機能に対しては，文書種別と応用規格の組み合わせごとに，タグ名の変更，構造の詳細化，事前に定義した部分文書構造に基づいた構造変換などの多様な構造化処理を定義可能とすると共に，それらを組み合わせることによって簡単な文書構造を複雑な文書構造に変換することを可能とする．上記 (3) では，応用規格で定義されている文書構造に基づいて上記 (2) の構造化結果を自動整形することを可能とする．

以下では，2 章で提案方式の基本原理とシステム構成について述べた後，3 章で XML 文書変換アルゴリズムについて説明する．そして，4 章で多様な文書を用いた XML 文書変換の実験結果を示し，提案方式の有効性について評価する．

2. 基 本 原 理

2.1 XML 文書変換システムの構成

本論文で提案する XML 文書変換システムは，入力文書解析，表層表現抽出，構造詳細化，整合性獲得，文書出力で構成されており，既存の紙文書，プレーンテキスト，XHTML 文書，XML 文書を文書型定義に基づいた XML 文書（以後ターゲット XML 文書と呼ぶ）に自動変換する（図 1）．紙文書が入力される場合には，文献[1] の OCR 技術および文書構造解析技術により，章構造，箇条書き構造，表構造，図構造などで構成される XHTML 文書にあらかじめ変換するものとする．

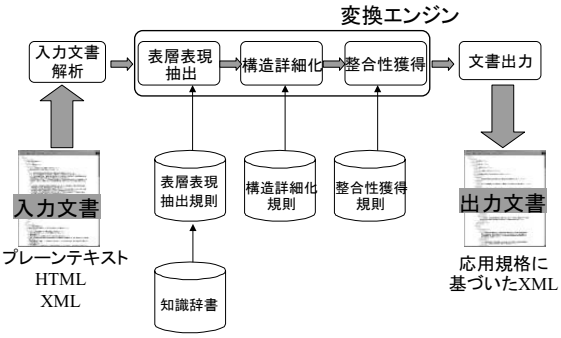


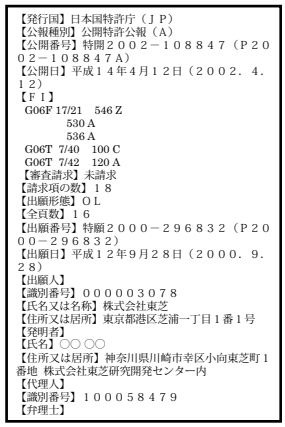
図 1　XML 文書変換の概要

以下では，図 2 に示す，プレーンテキストの特許公報を応用規格である特実広報 XML に基づいて XML 化する例を用いて提案システムの概要について説明する．

入力文書解析では，本システムに与えられた XML や XHTML 文書を XML パーサにより DOM ツリーに変換する．プレーンテキストが入力された場合には，XML 宣言を文頭に挿入すると共に文書全体に <root> タグを付与し，改行単位のテキストに対して <p> タグを付与するものとする．図 2 (a) に入力文書の一例を示す．

表層表現抽出では，入力文書の DOM ツリーからテキストノードを収集すると共に，知識辞書を用いてテ

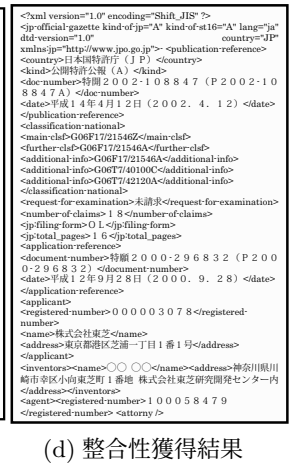
キストノードから文書構造化の手掛かりとなる表層表現を抽出する．知識辞書には，文書中から抽出すべき見出し語やキーワードなどがあらかじめ定義されているものとする．さらに，本システムでは，表層表現抽出規則を用いて表層表現抽出結果をもとに文書要素に初期タグと呼ばれる便宜的なタグを付与する．知識辞書と表層表現抽出規則の構成については次章で詳細に説明する．特許公報中の表層表現に対して初期タグを付与した例を図 2 (b) に示す．



(a) 入力文書

(b) 表層表現抽出結果

(c) 構造詳細化結果



(d) 整合性獲得結果

図 2 XML 文書変換処理過程の例

構造詳細化では，表層表現および文書要素へのタグ付け結果に対して構造詳細化規則を順次適用することにより，応用規格に基づいた文書構造の複雑化を行う（図 2 (c)）．提案方式では，構造詳細化規則を適用して得た中間的なタグ付け結果に対してさらに構造詳細化規則をくり返し適用することを可能としている．このように構造複雑化処理を多段に実施することにより，簡単な文書構造を複雑な文書構造に変換することを実現している．構造詳細化規則については次章で詳細に説明する．

入力文書とターゲット XML 文書において文書内容の出現順序が異なる場合には，構造詳細化処理では局所的な範囲においてのみ文書型定義に基づいたタグ付けが実施されていると見なすことができる．このため整合性獲得処理では，構造詳細化処理で得られたタグ付け結果に対して整合性獲得規則を適用して部分文書構造の並べ替えを行う．さらに，表層表現抽出処理で付与した初期タグや構造詳細化処理で付与した中間的なタグを削除する整形処理を実施する．この結果，文書型定義に基づいた厳密なタグ付け結果を得ることが可能となる．整合性獲得規則については次章で詳細に説明する．整合性獲得処理で得られたタグ付け結果の例を図 2 (d) に示す．

文書出力処理では XML パーサを用いることにより，上述した処理で得られたタグ付け結果が反映されている DOM ツリーをターゲット XML 文書に変換する．

2.2 変換ルールの定義

提案方式で用いる表層表現抽出規則，構造詳細化規則，整合性獲得規則は複数個の変換ルールで構成されている．変換ルールは，初期タグの付与，タグ名の変更，事前に定義した部分的な文書構造に基づく変換（テンプレート変換）など様々な変換タスクの定義を可能としている．提案方式では，対象文書と文書型定義の組み合わせごとに，変換ルールを宣言的に組み合わせて表層表現抽出規則，構造詳細化規則，整合性獲得規則を構成するものとする．

提案方式では，以下に示すように，変換ルールを XML で記述している．

```
<rule>
    <type>コマンド名</type>
    <key>条件記述</key>
    <tag>結果記述</tag>
    <begin>option(始点指定)</begin>
    <end>option(終点指定)</end>
</type>
```

変換ルールを構成する各要素の役割を以下に定義する．

type 要素： 特定の変換タスクをコマンドとして指定する．表層表現抽出規則では 3 種類，構造詳細化規則では 14 種類，整合性獲得規則では 5 種類のコマンドを指定することができる．それぞれのコマンドについては次章で説明する．

key 要素： ルールの起動条件となる表層表現，タグ名，部分文書構造を指定する．

tag 要素： 変換後のタグ名や部分文書構造を指定する．

begin/end 要素： 変換ルールの適用範囲やコマンドオプションを指定する．

以下に，構造詳細化規則のコマンドの一つである

「連続する複数の兄弟関係のタグをまとめ上げると共に親タグを新たに付与する変換タスク」に関して，変換ルールの具体的な記述例とそれを用いた場合の変換例を示す (図 3).

```
<rule>
    <type>add</type>
    <key>a</key>
    <tag>GROUP</tag>
    <begin>1</begin>
    <end>2</end>
</type>
```

ここでは，type 要素において兄弟関係のタグをまとめ上げるコマンド “add”を指定し，key 要素において変換対象となるタグ <a>(図 3 入力例) の要素名を指定し，tag 要素において変換後に付与される新しい親タグ <GROUP>(図 3 出力例) の要素名を指定し，begin 要素においてまとめ上げの範囲の始点となる指定タグ <a> の上位階層のレベルを指定し，end 要素においてまとめ上げの対象となる兄弟タグの範囲の終点を指定している．

図 3 の入力例に対してこのルールを適用した場合，タグ名 <sample1> と <sample2> に対して親タグ <GROUP> が付与されることになる．

```
<document>
  <sample1>
    <a>項目1</a>
    <b>内容1</b>
  </sample1>
  <sample2>
    <c>項目2</c>
    <d>内容2</d>
  </sample2>
</document>
```

入力例

```
<document>
 <GROUP>
   <sample1>
     <a>項目1</a>
     <b>内容1</b>
   </sample1>
   <sample2>
     <c>項目2</c>
     <d>内容2</d>
   </sample2>
 </GROUP>
</document>
```

出力例

図 3 「まとめ上げ」変換ルールの適用例

3. XML 文書変換アルゴリズム

以下では，図 2 に示すプレーンテキストの特許公報を特実広報 XML に変換する例を用いて，文書変換エンジンを構成する表層表現抽出，構造詳細化，整合性獲得の詳細について説明する．

3.1 表層表現抽出

ここでは説明の便宜上，図 2 (a) に示す特許公報のプレーンテキストの一部（図 4 (a)）を用いて表層表現抽出処理の動作を説明する．表層表現抽出ではまず，図 4 (a) の入力文書解析結果に対して図 5 に示す知識辞書を参照しながらパターンマッチングを行なう．図 4 (b) にパターンマッチングによる表層表現抽出結果を示す．次に，このパターンマッチング結果に対して，図 6 と 7 に示すような表層表現抽出規則を適用して図 4 (c) の表層表現および文書要素に対するタグ付け結果を得る．

```
<p>【ＦＩ】</p>
<p>  G06F 17/21   546 Z</p>
<p>               530 A</p>
<p>               536 A</p>
<p>  G06T  7/40   100 C</p>
<p>  G06T  7/42   120 A</p>
<p>【審査請求】未請求</p>
<p>【請求項の数】１８</p>
<p>【出願形態】ＯＬ</p>
<p>【発明者】</p>
<p>【氏名】山田  太郎</p>
<p>【住所又は居所】神奈川県川崎市幸区小向東芝町１番地  株式会社
東芝研究開発センター内</p>
<p>【代理人】</p>
<p>【識別番号】１０００５８４７９</p>
<p>【弁理士】</p>
```

(a) 入力文書解析結果

```
<p> 【ＦＩ】 </p>
<p>  G06F 17/21   546 Z</p>
<p>               530 A</p>
<p>               536 A</p>
<p>  G06T  7/40   100 C</p>
<p>  G06T  7/42   120 A</p>
<p>【審査請求】 未請求</p>
<p>【請求項の数】 １８</p>
<p>【出願形態】 ＯＬ</p>
<p>【発明者】 </p>
<p>【氏名】 山田  太郎</p>
<p>【住所又は居所】 神奈川県川崎市幸区小向東芝町１番地
株式会社東芝研究開発センター内</p>
<p>【代理人】 </p>
<p>【識別番号】 １０００５８４７９</p>
<p>【弁理士】 </p>
```

(b) パターンマッチング結果

```
<p><_FI>【ＦＩ】</_FI> </p>
<p>G06F 17/21 546 Z</p>
<p>530 A</p>
<p>536 A</p>
<p>G06T 7/40 100 C</p>
<p>G06T 7/42 120 A</p>
<p><_req_for_exam>【審査請求】</_req_for_exam>  未請求  </p>
<p><_number-of-claims>【請求項の数】</_number-of-claims> １８ </p>
<p><_filing-form>【出願形態】</_filing-form>  ＯＬ  </p>
<p><_inventor>【発明者】</_inventor> </p>
<p><name> <_name>【氏名】</_name> 山田  太郎 </name></p>
<p><address>  <_address_>【住所又は居所】</_address_>神奈川県川崎
市幸区小向東芝町１番地 株式会社東芝研究開発センター内</address></p>
<p><_agent>【代理人】</_agent></p>
<p><_attorny>【弁理士】</_attorny></p>
```

(c) 表層表現および文書要素のタグ付け結果

図 4 表層表現抽出処理

パターンマッチングの際に用いる知識辞書は，図 5 に示すように文書中から抽出すべき表層表現と ID の組で記述された表層表現リストの集合で構成されている．知識辞書で定義されている表層表現を入力文書と順次照合することによって，ID つき表層表現抽出結果を得る．なお知識辞書中で用いるパターン表現では，図 5 中の “^【請求項 [0-9]+】$”のように，正規表現を含んだ記述を可能としている．このような正規表現を用いることにより，日付，電話番号，製品名，型番，ヘディング付き章見出しなどの数値表現や記号表現を抽出することが可能となる．

提案方式で採用している表層表現抽出規則の変換ルールでは，パターンマッチング結果に対して直接タグを付与する変換コマンド：*direct_tagging*，パターンマッチング結果から行頭や行末までタグを付与する

変換コマンド：*head* または *tail* を選択することができる．パターンマッチング箇所に対して直接タグを付与する場合には，図 6 に示す変換ルールを適用する．この結果，入力文書中の表層表現 “【住所又は居所】” に対し “<_address>” タグが付与される．一方，マッチングされた表現から行末までタグ付けする場合には，図 7 に示す変換ルールを適用する．この結果，入力文書中の表層表現 “【住所又は居所】” と文書要素 “神奈川県川崎市幸区小向東芝町 1 番地　株式会社東芝研究開発センター内” を含む範囲に対して “<address>” タグが付与されることになる．

```
pat;ja01;^【FI】$
pat;ja01;^FI$
pat;ja02;^【弁理士】$
pat;ja03;^【請求項[0-9]+】$
pat;ja04;【発明者】;
pat;ja05;【代理人】
pat;ja06;【住所又は居所】
pat;ja07;【審査請求】
pat;ja07;^審査請求.*$;
pat;ja08;【全頁数】
pat;ja09;【出願形態】
pat;ja10;【請求項の数】
・・・
```

図 5　知識辞書例

```
<rule>
  <type>direct_tagging</type>
  <key>pat.ja06</key>
  <tag>_address</tag>
  <begin />
  <end />
</rule>
```

図 6　表層表現抽出規則 (直接タグ付与) の例

```
<rule>
  <type>tail</type>
  <key>pat.ja06</key>
  <tag>address</tag>
  <begin />
  <end />
</rule>
```

図 7　表層表現抽出規則 (行末までのタグ付与) の例

3.2　構造詳細化

構造詳細化処理では，表層表現や文書要素のタグ付け結果に対して構造化詳細規則を適用することにより特定の応用規格に基づいた文書構造化処理を実施する．構造詳細化規則を構成する変換ルールでは，以下に示す 14 種のコマンドの指定を可能としている．

rename： タグ名の変更

rename_t： 属性や属性値の指定を可能とするタグ名の変更

pas： 指定個数の兄弟要素に対する親タグの付与

group_tagging/group_tagging_simple/ group_tagging_simple_nk： 構造の類似性を有する連続した兄弟要素に対する親タグの付与

addp： 指定タグへの親タグの付与

add： 指定タグへの子タグの挿入

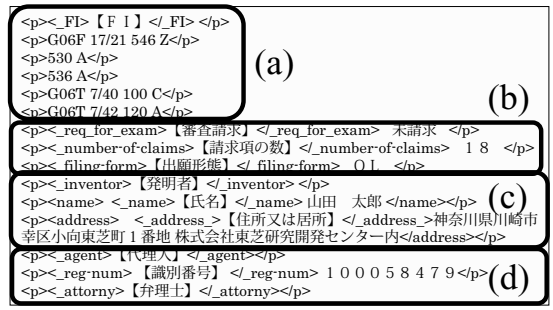
transform： 変換後の XML 構造を見本として与えることによる部分構造の変換 (テンプレート変換)

transform_refrain： 入力文書中のテキストを部分的に再利用するとともにルールの適用回数に応じてカウンタ値を挿入可能とするテンプレート変換

rename_p/rename_p2/rename_p3： 指定のタグから任意階層をたどった先のタグ名の変更

table_trans： 表の空セルへの空タグの付与や表要素間の対応付けを可能とする表構造変換

以下では，一例として，図 4 (c) の特許公報のタグ付け結果を対象とした場合の構造詳細化処理について説明する．ここで，説明の便宜上，図 8 に示すように図 4 (c) のタグ付け結果を (A)–(D) の 4 つの部分領域に分割する．そして，以下に，部分領域 (A)，(B)，(C)，(D) に対する構造詳細化処理をそれぞれ説明する．



(a) 表層表現および文書要素のタグ付け結果

```
<classification-national>
<main-clsf>G06F17/21546</main-clsf>
<further-clsf>G06F17/21530</further-clsf>
<additional-info>G06F17/21536</additional-info>
<additional-info>G06T7/40100</additional-info>
<additional-info>G06T7/42120</additional-info>
</classification-national>
<request-for-examination> <_req_for_exam>【審査請求】</_req_for_exam>未請求</request-for-examination>
<number-of-claims> <_number-of-claims>【請求項の数】</_number-of-claims>１８</number-of-claims>
<jp:filing-form> <_filing-form>【出願形態】</_filing-form>ＯＬ</jp:filing-form>
<inventors><_inventor>【発明者】</_inventor><name> 山田　　太郎 </name><address> <_address_>【住所又は居所】</_address_>神奈川県川崎市幸区小向東芝町１番地 株式会社東芝研究開発センター内</address></inventors>
<agent> <p><_agent>【代理人】</_agent></p>
<registered-number> <_reg-num>【識別番号】 </_reg-num> １０００５８４７９</registered-number>
<attorny><_attorny>【弁理士】</_attorny></attorny> </agent>
```

(b) 構造詳細化結果

図 8　構造詳細化処理の例

(1) 部分領域 (A) の構造詳細化

図 8 の部分領域 (A) を対象とした場合の構造詳細化処理では，構造詳細化規則として以下の変換ルールを適用するものとする．

```
<rule>
    <type>table_trans</type>
    <key>_FI</key>
    <tag>classification-national
            <main-clsf>
            <further-clsf>
            <additional-info>
    </tag>
    <begin>2</begin>
    <end>pt</end>
</type>
```

この変換ルールにより実現される変換タスクでは，まず，図 8 (A) におけるタグ付け結果の <_FI> を始

点として，内容が英数字である要素をまとめ上げると共に，それらの要素に対して親タグ <classification-national> を付与する．次に_FI 要素以降に連続する 3 つの兄弟要素（図 8 (A) の G06F 17/21 546 Z ， 530 A ， 536 A ）に対して，上記変換ルールで指定されている 3 つタグ <main-clsf>，<further-clsf>，<additional-info> をそれぞれ付与する．さらに，それ以降に連続する兄弟要素（図 8 (A) の G06T 7/40 100 C と G06T 7/42 120 A ）が存在する場合には，それらに対して <additional-info> タグを継続して付与する．このような繰り返しによるタグ付け処理は，変換ルールの begin 要素で定義されている内容にしたがっている．begin 要素の内容が “1”であれば，ルールで指定したタグを繰り返し付与するものとする．begin 要素の内容が “2”であれば，tag 要素の内容の最後に定義されているタグを繰り返し付与するものとする．また，変換ルールの end 要素の内容 “pt”は，内容の一部が省略されている要素（図 8 (A) の 530 A と 536 A ）に対して省略されている文字列 (“G06F”と “17/21”) を挿入することを指定するものである．

(2) 部分領域 (B) の構造詳細化

図 8 の部分領域 (B) を対象とした構造詳細化処理では，_req_for_exam 要素，_number-of-claims 要素，_filing-form 要素のそれぞれに対して親タグの変更を行う．以下に，一例として _req_for_exam 要素を対象とした場合の変換ルールを示す．

```
<rule>
    <type>rename_p</type>
    <key>_req_for_exam</key>
    <tag>request-for-examination</tag>
    <begin />
    <end />
</type>
```

この変換ルールにより，_req_for_exam 要素の親タグ が <request-for-examination> に変更される．

(3) 部分領域 (C) の構造詳細化

図 8 の部分領域 (C) を対象とした構造詳細化処理では，以下に示す変換ルール適用することによって部分領域 (C) を囲む親タグを設定する．

```
<rule>
    <type>pas</type>
    <key>_inventor</key>
    <tag>inventors</tag>
    <begin>1</begin>
    <end>3</end>
</type>
```

この変換ルールに基づく変換タスクでは，表層表現タグ付け結果の “<_inventor>”要素を手がかりとして部分領域 (C) を囲む親タグを付与することになる．このとき，変換ルールの begin 要素の内容 “1”にしたがって “<_inventor>”要素の親要素 “ ”を始点とすると共に，変換ルールの end 要素の内容 “3”に基づいて始点を含む 3 つの兄弟要素までの範囲に対して親タグ “<inventors>”を付与する．

(4) 部分領域 (D) の構造詳細化

図 8 の部分領域 (D) を対象とした構造詳細化処理では，まず以下に示す変換ルールを適用することにより，_reg-num 要素の親タグ を <registered-number> に変更する．

```
<rule>
    <type>rename_p</type>
    <key>_reg-num</key>
    <tag>registered-number</tag>
    <begin />
    <end />
</type>
```

次に，部分領域 (C) の構造詳細化処理と同様の手順により，部分領域 (D) を囲む親タグ <agent> を設定する．そして，部分領域 (B) の構造詳細化処理と同様の手順により，_attorny 要素の親タグ を <attorny> タグに変更する．

以上のように，部分領域に対する構造詳細化処理を順次実施することにより，表層表現や文書要素のタグ付け結果（図 8 (a)）に対する構造詳細化結果（図 8 (b)）を得ることが可能になる．なお，他のコマンドを用いた変換ルールとその変換例についてはページの都合上省略する．

3.3 整合性獲得

整合性獲得では，構造詳細化処理の出力結果に対して整合性獲得規則を適用することにより，応用規格で定義されている文書構造の制約を充足するターゲット XML 文書を生成する．整合性獲得規則では，応用規格で定義されている文書構造を tag 要素において再定義することになっている．この作業は，多少煩雑ではあるが，できあがった変換ルールは可読性の高いものになっている．

整合性獲得規則を構成する変換ルールでは，以下に示す 5 種のコマンドを指定することができる．

del： 指定タグの削除

delall： 指定タグとその内容およびすべての子要素の削除

restriction： 指定タグの子要素に対して出現順序制約を適用した並べ替え

neg： 特定の親タグを持たない指定タグとその内容の削除

pos： 特定の親タグを持つ指定タグのタグ名の変更

以下では，一例として，図 9 (a) の構造詳細化結果

を対象とした場合の整合性獲得処理について説明する．この整合性獲得処理では，整合性獲得規則として以下の変換ルールを図 9 (a) の構造詳細化結果に対して適用する．

```
<rule>
    <type>delall</type>
    <key>_reg-num</key>
    <tag />
    <begin />
    <end />
</type>
```

この変換ルールでは，構造詳細化結果から表層表現抽出の際に付与した初期タグとその内容およびその子要素すべてを削除する変換タスクを定義している．図 9 の例では，この変換ルールに基づいて _reg-num 要素が削除されることになる．図 9 ではさらに，_req_for_exam 要素，_number-of-claims 要素，_filing-form 要素，_inventor 要素，_address 要素，_agent 要素，_reg-num 要素，_attorny 要素なども同様に削除される．

以上の整合性獲得処理を実施することにより，図 9 (b) に示す応用規格に基づいた XML 文書を生成することが可能となる．

```
<classification-ipc>
<main-clsf>G06F17/21546</main-clsf>
<further-clsf>G06F17/21530</further-clsf>
<additional-info>G06F17/21536</additional-info>
<additional-info>G06T7/40100</additional-info>
<additional-info>G06T7/42120</additional-info>
</classification-ipc>
<request-for-examination>  <_req_for_exam>【審査請求】</_req_for_exam>未請求</request-for-examination>
<number-of-claims>  <_number-of-claims>【請求項の数】</_number-of-claims>  １８</number-of-claims>
<jp-filing-form> <_filing-form>【出願形態】</_filing-form>ＯＬ</jp-filing-form>
<inventors><_inventor>【発明者】</_inventor><name>山田　太郎</name><address> <_address_>【住所又は居所】</_address_>神奈川県川崎市幸区小向東芝町１番地 株式会社東芝研究開発センター内</address></inventors>
<agent> <p><_agent>【代理人】</_agent></p>
<registered-number> <_reg-num>【識別番号】 </_reg-num> １０００５８４７９</registered-number>
 <attorny><_attorny>【弁理士】</_attorny></attorny> </agent>
```

(a) 構造詳細化結果

```
<classification-ipc>
<main-clsf>G06F17/21546</main-clsf>
<further-clsf>G06F17/21530</further-clsf>
<additional-info>G06F17/21536</additional-info>
<additional-info>G06T7/40100</additional-info>
<additional-info>G06T7/42120</additional-info>
</classification-ipc>
<request-for-examination>未請求</request-for-examination>
<number-of-claims>１８</number-of-claims>
<jp-filing-form>ＯＬ</jp-filing-form>
<inventors><inventor><name>山田　太郎</name><address> 神奈川県川崎市幸区小向東芝町１番地 株式会社東芝研究開発センター内</address> </inventor> </inventors>
<agent><registered-number>１０００５８４７９</registered-number> <attorny />
```

(b) 整合性獲得結果

図 9　整合性獲得処理

図 10 のように，構造詳細化結果における文書要素の出現順序とターゲット XML 文書の文書要素の出現順序が異なっている場合には，構造詳細化結果に対して並べ替えを行う変換ルールを適用して整形処理を実施する．図 10 の例では，以下の変換ルールを適用することにより，storage 要素を approvalbrandname 要素の子要素として配置すると共に，変換ルールの tag 要素の定義内容に基づいて要素の並べ替えを行う．

```
<rule>
  <type>restriction</type>
  <key>approvalbrandname</key>
  <tag>
      <approvalandlicenseno />
      <dateofnhidrugpricelisting />
      <startingdateofmarketing />
      <storage />
      <composition_or_property/>
  </tag>
  <begin />
  <end />
</rule>
```

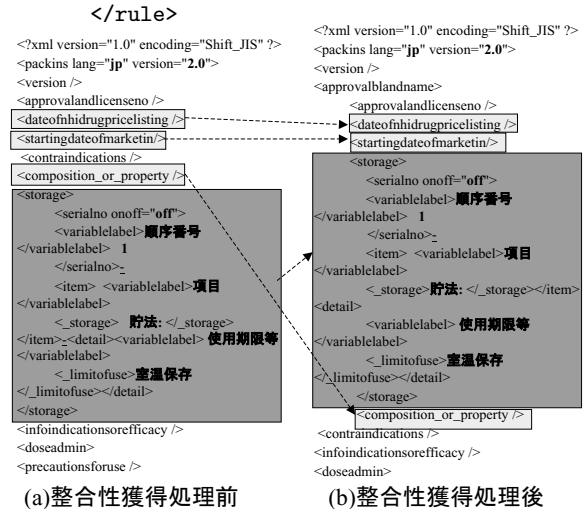


図 10　並べ替えによる整形処理

4. XML 文書変換実験

本研究では，実際の業務で用いられている紙文書やプレーンテキストを対象として，それらを特定の応用規格に基づいた XML 文書に変換する実験を行った．実験では，以下の計測を行って提案方式の有効性を評価した．

- XML 文書変換の精度
- XML 文書変換に要した時間および手作業による XML 変換との比較

以下に，実験方法と実験結果および考察について述べる．

4.1 実験方法

本論文で提案した XML 文書変換システムを Java プログラムとして実装し，PC 上で実験を行なった．実験環境の構成を，OS：Windows2000 Professional, CPU：PentiumIII 1GHz，主記憶：512MB，JDK のバージョン：1.3.1_08 とした．

実験では，A4 サイズの紙文書 8 ページ相当の事務規定文書 1 セット，A4 サイズの医薬品添付文書 5 セット（1 セットあたり 3〜5 ページ相当），プレーンテキス

トで記述された特許公報文書 5 セットを対象として文書変換精度を計測した．事務規定文書を対象とした場合には，弊社で策定した DTD に基づく XML 文書変換を実施した．医薬品添付文書と特許広報を対象とした場合には，省庁が策定した DTD に基づく文書変換を実施した．紙文書を対象とした場合には，あらかじめ文献[1)] の方法を適用することにより，章構造，箇条書き構造，表構造，図構造などで構成される XHTML 文書に変換した．このとき，専用の GUI を用いて画像解析誤り，文字認識誤り，構造化誤りなどの修正を行ってクリーンデータを作成した．

また，医薬品添付文書 1 セットを用いて，提案方式による XML 文書変換時間と手作業による XML 文書変換時間をそれぞれ測定したあと比較検討を行った．

4.2 実験結果

上述した文書サンプルに対する XML 文書変換結果を表 4.2 に示す．表 4.2 における変換ルール総数とは，表層表現抽出規則，構造詳細化規則，整合性獲得規則の合計を表している．タグ種別数とは，応用規格において定義されているタグの総数である．正解文書タグ数とは，入力文書に対して手作業でタグを付与した正解文書に含まれるタグの総数である．タグ付与率とは，正解文書タグ数対する本システムによるタグ付与結果の割合を示したものである．対象タグへのタグ付与率とは，正解例文書に付与されているタグの総数から，本システムで原理的に付与できないタグの数を除いたものに対するタグ付け結果の割合である．

実験結果では，表層表現の未抽出や抽出誤り，変換ルールの未適用や適用誤りなどに起因する文書変換誤りが生じた．表層表現抽出誤りについては，本研究で採用した知識辞書では数量表現，バリエーションを伴う性質の表現，医薬品名や物質名などの固有表現を適切に扱うことができないため抽出誤りが生じたと考えることができる．変換誤りについては，提案方式では変換ルールをボトムアップに適用して一連の変換処理を実施するようになっているため，同一の表層表現に対して複数の構造化処理が存在する場合には対応できないことが原因であると考えることができる．このような場合には，複数の表層表現の局所的分布から実施すべき構造化処理を推定したあとで，適切なルールセットを適用するアプローチを新たに導入する必要がある．

医薬品添付文書 3 ページ相当 1 セットを対象とした場合の手作業による XML 文書変換では，紙文書の内容をキーボード入力してプレーンテキスト化する作業に 57 分要し，XML エディタを用いてプレーンテキストを応用規格に基づいた XML 文書に変換する作業に 185 分要したため，合計 242 分かかった．一方，本システムによる XML 文書変換を実施した場合には，文献[1)] で実現した OCR システムを用いて XHTML 文書のクリーンデータを作成する作業に 7 分を要し，提案方式により XHTML 文書を構造化したあと誤り箇所を修正してターゲット XML 文書を作成する作業に 16 分を要したため，合計 23 分かかった．この結果，提案方式を用いることにより，手作業の 1/10 の時間で既存文書をターゲット XML 文書に正しく変換できることが分かった．

表 4.2 実験結果

文書種別	事務規定文書	医薬品添付文書	特許公報
変換ルール総数 (個)	49	131	107
タグ種別数 (種類)	36	369	116
対象タグ数 (個)	23	215	58
正解文書タグ数 (個)	165	612.7	445.3
タグ付与率 (%)	87.8	75.6	83.5
対象タグへのタグ付与率 (%)	94.1	87.4	91.3

5. まとめ

本論文では，紙文書やプレーンテキストの法令集，官報，約款集，規定集，論文，名刺などを応用規格に基づいた XML 文書に自動変換する新しい文書変換システムを提案した．本システムでは，入力文書に対して表層表現抽出処理を適用することにより見出し語やキーワードを自動抽出するとともに表層表現を手がかりとして文書要素に対して柔軟なタグ付け処理を行うことを可能とした．さらに，文書要素のタグ付け結果に対して構造詳細化処理と整合性獲得処理を順次適用することにより，入力文書から応用規格に基づいた XML 文書を自動生成することを可能とした．本研究では，対象文書と応用規格の組み合わせに対して知識辞書と文書変換規則をあらかじめ準備しておけば，上述した XML 文書変換処理を全自動で実施することが可能となる．その結果，手作業による XML 文書変換工程と比較して 1/4〜1/10 の作業時間で既存文書を応用規格に基づいた XML 文書に変換することが可能となった．また，文書内容，XML 技術，XML 応用規格などに関する専門的知識を持たないオペレータでも XML 文書を作成することが可能となった．

参考文献

1) 石谷 康人, 住田 一男. 紙文書を対象としたピボット XML 文書に基づく XML 文書変換システム. 電子情報通信学会 信学技報, TL2003-30, pp7-12, 2004.