情報流通分野における XML 変換方式の研究

鳥海 幸輝，春日 史朗，坂田 哲夫，小林 伸幸，芳西 崇
NTT サイバースペース研究所
情報ベースプロジェクト　情報ベース検索方式グループ

Abstract

XML 電子伝票を用いた企業間電子商取引においては， XML 電子伝票のスキーマが統一されている必要がある．そこで，B2B フレームワークによって XML 電子伝票の標準スキーマが規定されている。各企業のデータベースに格納されている XML 電子伝票は，独自スキーマを用いている場合が多いため，標準スキーマと各企業の独自スキーマを相互変換する必要がある．これらの相互変換を行うには変換ルールの作成が必要であるが，XML 電子伝票の種類が増加し，構造が複雑になるにつれて，記述する変換ルール数が増加し，その作成稼動が増加するという問題がある．本稿では，この問題に対し，2 つのスキーマ間で，タグ名称の類似性と木構造の類似性に基づいて変換ルールの候補を自動生成することで，その作成稼動を削減する手法を提案する．更に，本手法の有効性を確認するためにプロトタイプを作成し，同業種間でのスキーマ変換において本手法を用いた場合，従来方式と比較して最大約 60%稼動の削減が行えることを確認した．

1. はじめに

近年，インターネット等，ネットワークのオープン化に伴い，新たな情報流通の共通フォーマットとして，XML が注目されている．XML は，アプリケーションやソフトウェアに依存しない汎用的なデータ記述言語であるため，活発化している企業間電子商取引(B2B)の分野においても積極的に採用され始めている．これに伴い，XML 電子伝票の流通のための規約を定めた B2B フレームワーク(RosettaNet[1], ebXML[2]等)と呼ばれるものが登場している．B2B フレームワークとは，電子商取引の手段を標準化したものであり，その中で XML 電子伝票の標準スキーマが規定されている．

一方，各企業内のシステムも XML により構築されるようになっているが，各企業で用いられている XML 電子伝票は，各企業のシステムに固有であるため，独自のスキーマを用いている場合が多い．そのため，企業が B2B に参加し，各企業の XML 電子伝票を B2B フレームワークを通して他企業に送信する際に，各企業のシステムに格納された独自の XML 電子伝票のスキーマを B2B フレームワークの標準スキーマに変換する必要がある．

現在，異なるスキーマ間の変換ルール(XML の変換に必要な要素と要素の対応関係)を作成する方法としては，GUI ツールを用いて変換ルール作成者が変換ルールを作成する方法が主流となっている[3][4][5]．しかしこれらの GUI ツールは，流通する XML の種類が増加し，構造が大きくなるにつれ，記述する必要のある変換ルール数が膨大となり，変換ルール作成者の稼動が増大するという問題がある．

そこで本稿では，XML 電子伝票の変換において，人手による変換ルール作成の稼動を削減するため，変換ルールの候補を自動的に生成し，変換ルール作成者が変換候補を選択する半自動変換方式を提案する．

以下，2 章では，従来手法を概観し，その問題点を明確化する．3 章では，本稿の提案手法と，そのアルゴリズムについて述べる．4 章では，提案アルゴリズムを実装したプロトタイプシステムを用いた評価実験の結果を示す．第 5 章以降ではこれらを受けて考察し，今後の課題などについて述べる．

2. 従来手法およびその問題点

XML の変換ルールの作成手法としては，GUI ツールを用いて変換ルールを作成する手法と，変換ルール作成の自動化を行う手法がある．本章では，それらの従来手法を概観し，その問題点を明確化する．

(1)GUI ツール

XML の変換を行う変換言語として，W3C により標準化された XSLT[6]がある．しかし，XSLT は記述が複雑であり，直接記述することが困難である．そのため，現在は GUI を用いて XSLT スクリプトを作成するツール(GUI ツール)が主流となっている．

しかし，これらの GUI ツールでは，要素同士の対応関係を，全て変換ルール作成者が判断するしかなく，流通する XML の種類や構造が大きくなるにつれ，変換ルール生成の稼動が増大するという問題がある．

(2)変換ルール生成の自動化

GUI ツールの問題点に対し，変換ルール作成者の稼動削減ために，変換ルールを自動的に導出する手法[7]が提案されている．

しかし，この手法は変換ルールを完全に自動生成するものであり，生成された変換ルールに，誤変換が混入する場合もある．そのため，厳密性が要求される B2B には，そのままでは適用出来ないという問題がある．

3. 提案手法

3.1. アプローチ

前章で述べた問題点に対して本稿では，変換ルールの候補を自動生成した上で，変換ルール作成者がその候補から所望の組み合わせを選ぶことで，最終的な変換ルールを確定する半自動手法を提案する．以下に示す 2 つの手法を組み合わせることで，半自動手法を実現する．

(1)変換候補生成

XML 電子伝票の変換において，通常，ある 1 つの要素の変換先とは，それと同等な内容を有する要素である．よって，要素と要素が似ている度合(以下，類似度)を算出し，類似度の高い要素を選び出すことで，変換先となる可能性を持つ要素(以下，変換候補)を選定する．

(2) 変換ルール作成者による変換ルールの確定

B2B の世界は，取引情報のやり取りが行われることから厳密さが求められる．従って，類似度が最も高いものから順位付け(以下，ランキング)された変換候補に対して，変換ルール作成者が判断を行うことによって最終的な変換ルールを確定する。

3.2. 変換パターンの分類

XML の変換パターンには多くの種類が存在するため，それら分類し，本稿で対象とする変換パターンを個別に検討する必要がある．本稿では，XML の変換パターンを図 1 のように分類する．

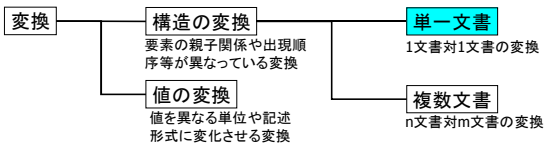


図 1　XML の変換パターン

(1)構造の変換

要素の親子関係や出現順序などが異なっている変換を指す．構造の変換は更に 2 つの種類に分類できる．

(a)単一文書の変換

単一文書の変換とは，1 文書対 1 文書の変換パターンを指す．この変換は，図 2 に示すように，あるスキーマ A から別のスキーマ B に変換する．

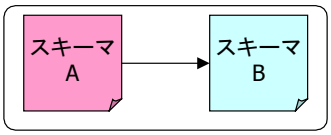


図 2　1 文書 対 1 文書の変換

(b)複数文書の変換

複数文書の変換とは，n 文書対 m 文書の変換パターンを指す．図 3 に示すように，スキーマ A，B を統合し，C，D，E に分割するような変換である．

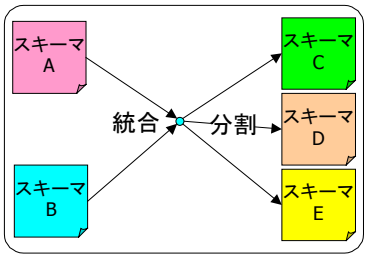


図 3　複数文書の変換

(2)値の変換

図 4 に示すように，値を異なる単位や記述形式に変化させる変換を指す．

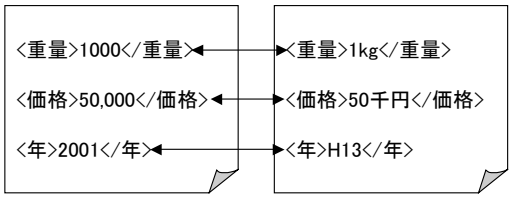
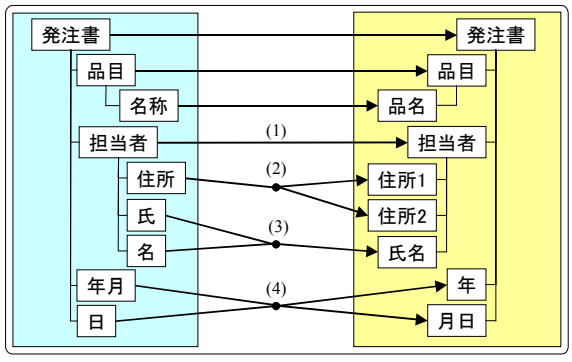


図 4　値の変換

上記のように分類された中で，値の変換は既に既存研究[8][9]で実現されており，複数文書の変換は，単一文書の応用形であると考えられるため，本稿では，値の変換を伴わない単一文書の変換を検討する．

図 5 に示すように，単一文書同士の変換は，全て要素対要素の対応関係(変換ルール)によって定義することができる．本稿では，図中の 4 種類の変換の中で最も基本的な変換ルールである，1 要素対 1 要素変換を検討する．



(1)1 要素対 1 要素変換　　(3)n 要素対 1 要素変換
(2)1 要素対 n 要素変換　　(4)m 要素対 n 要素変換

図 5　単一文書同士の変換

3.3. XML の特性

本稿は, XML の変換の中でも, B2B 分野における XML 電子伝票の変換を扱う. XML 電子伝票の例を図 6 に示す. 我々はこのような電子伝票を幾つか検討した所, B2B 分野における XML 電子伝票は以下の傾向を持つことが分かった.

●傾向 1　変換対象となる要素同士は, タグ名称が類似している. また, タグ名称は複合語を用いている場合が多い
図 6 に示すように, <情報区分コード>, <色柄サイズ番号>等, 複合語が用いられている場合が多い.

●傾向 2　値は末端要素に記述されている
図6に示すように, 値は全て末端要素(<品番>, <品名>等)に記述されており, 中間要素(<製品一覧>, <マルチ明細>等)には記述されていない.

●傾向 3　中間要素は, その特性を表現している子要素を下位に持つ
図6に示すように, 中間要素(<製品>)は, その特性を表現している子要素(<品番>, <品名>等)を下位に持っている.

ここで, 子要素を持たない要素を末端要素, 子要素を持つ要素を中間要素と定義する. これらの特性を利用した効率的な変換方法を次節で検討する.

```
<?xml version="1.0" encoding="Shift_JIS" ?>
                    :
  <製品一覧>
    <製品>
      <品番>SA512</品番>
      <品名>長袖 T シャツ</品名>
      <発売元>アパレル興業(株)</発売元>
      <単価>2000</単価>
      <情報区分コード>8501</情報区分コード>
      <訂正コード>1</訂正コード>
      <ブランドコード>21</ブランドコード>
      <年度>2001</年度>
      <シーズンコード>1</シーズンコード>
      <アパレル製品備考 1>洗濯時色落ち有り</アパレル製品備考 1>
      <アパレル製品備考 2>洗濯時縮み有り</アパレル製品備考 2>
      <マルチ明細>
        <色柄サイズ番号>L</色柄サイズ番号>
        <色柄参照番号>N14</色柄参照番号>
        <色柄サイズ別数量>1</色柄サイズ別数量>
        <色柄別数量の単位>10</色柄別数量の単位>
      </マルチ明細>
    </製品>
  </製品一覧>
                    :
```

図 6　XML 電子伝票の例(抜粋)

3.4. 変換候補生成法

本節では, XML 電子伝票の特性を考慮した変換候補生成法について述べる. XML 電子伝票のスキーマの類似性には以下の 2 つの観点があるため, それらの類似性を組み合わせることによって, 要素と要素の類似性を判断する.

⑴タグ名称に基づく類似性
⑵木構造に基づく類似性

3.4.1. タグ名称に基づく類似性

本方式では, タグ名称間の類似性を判断するため, タグ名称間の意味的な距離(以下, タグ名称類似度)を用いる.

ここで, XML のタグ名称は複合語を用いている場合が多いため(傾向 1), 複合語間の意味的な距離を算出する必要がある.

それに適する方式として, 形態素解析によって複合語を単語に分割し, 単語群間の距離計算を行う方式(辞書 CB 方式)[10]が提案されており, 我々はこれを採用した. 辞書 CB 方式を用いることによって, 同じ概念を指す, 異なる表記の複合語に対してもタグ名称類似度[0～1]の算出が可能である(例えば, 要素<オーダーID>と要素<発注番号>の辞書 CB 方式によるタグ名称類似度の実測値は, 0.44 である).

3.4.2. 木構造に基づく類似性

本稿では, 3.3.節で述べた XML 電子伝票の特性を元に, 以下の観点で木構造に基づく類似性を判断する.

⑴末端要素と末端要素は, それらのタグ名称が似ていれば類似している(傾向 1 による)
末端要素は子要素を持たないため, 考慮されるのは, 結果的にタグ名称の類似性のみである.

⑵末端要素と中間要素の変換は存在しない(傾向 2 による)
下記の様に, 末端要素と中間要素は性質が全く異るため, それらの変換は存在しない.
・末端要素：値を持つが子要素を持たない
・中間要素：値は持たないが子要素を持つ

⑶中間要素と中間要素は, それらのタグ名称が似ているか, それらの直下の子要素同士で似ているものが多ければ類似している(傾向 3 による)

3.4.3. アルゴリズム

前節で示したタグ名称に基づく類似性と木構造に基づく類似性を元に, 本節では, 要素間の類似度を算出する下記のようなアルゴリズムを提案する. なお,

- S：変換元のスキーマ
  T：変換先のスキーマ
- lsim(s, t)：辞書 CB 方式に基づいて算出される, 要素 s と t のタグ名称類似度 (linguistic similarity coefficients)[0～1]
- cssim(s, t)：中間要素 s と t が持つ, 子要素集合同士の類似度(child set similarity coefficients)[0～1]
- wsim(s, t):要素 s と t のタグ名称類似度(lsim(s, t))に, 木構造に基づく類似性を反映させた類似度で

あり，変換候補として提示する際，ランキングに用いる(weighted similarity coefficients)[0～1]とする．

≪アルゴリズム≫

《Step 1》前処理

図 7 にスキーマ S とスキーマ T の例を示す．

S の全要素：{s_1, s_2, ‥‥, s_N}

T の全要素：{t_1, t_2, ‥‥, t_M}

と表せるものとする．

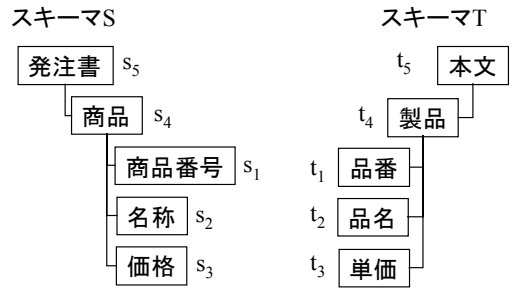


図 7　スキーマ S とスキーマ T の例

《Step 2》初期マトリックスの作成

スキーマ S の全要素数 N と，スキーマ T の全要素数 M としたとき，N×M の 2 次元配列(以下マトリックス)を用意し，全ての配列要素に対して配列の(s, t)の位置へ，lsim(s, t)を代入する．これを初期マトリックスと呼ぶ．初期マトリックスは，N と M において，総当りで lsim を記述した表である．初期マトリックスの例を図 8 に示す．

lsim

		t_1	t_2	t_3	t_4	t_5
		品番	品名	単価	製品	本文
s_1	商品番号	0.357901	0.132006	0.389349	0.230821	0.016713
s_2	名称	0.048502	0.740476	0.043345	0	0.009911
s_3	価格	0.02102	0.037401	0.430992	0.162992	0
s_4	商品	0.008087	0.191293	0.292344	0.324934	0.018272
s_5	発注書	0.050219	0.148417	0	0.060621	0.059899

図 8　初期マトリックスの例

《Step 3》出力マトリックスの作成

《Step 3-1》出力マトリックスの準備

初期マトリックスと同サイズの配列をもうひとつ用意する．これを，出力マトリックスと呼ぶ．

《Step 3-2》wsim の算出と出力マトリックスへの格納

出力マトリックスへ，マトリックスの左上より右下の順で以下の演算結果を格納する．

(ⅰ)要素 s と要素 t が共に末端要素の場合

$wsim(s, t) = lsim(s, t)$

(ⅱ)要素 s と要素 t が一方が末端要素で，一方が中間要素の場合

$wsim(s, t) = 0$

(ⅲ)要素 s と要素 t が共に中間要素の場合

$$wsim(s, t) = k_{const} \cdot lsim(s, t) + (1 - k_{const}) \cdot cssim(s, t)$$

k_{const}：重み定数[0～1]

(ⅲ)より，wsim(s, t)は，s と t が持つ子要素集合同士の類似度が反映される．それにより，スキーマ内の下位の要素間の wsim(s, t)が，上位の要素間の wsim(s, t)へ再帰的に反映されることになる．

【cssim(s, t)の算出方法】

[手順 1]s の子要素 s' = {s'1, s'2, ‥‥, s'n}と t の子要素 t' = {t'1, t'2, ‥‥, t'm}の排他的な組み合わせを求める．

・本稿は 1 要素対 1 要素変換を対象としているため，1 つの要素に対して 2 つ以上の変換ルールを作成することは考慮していない．従って，cssim(s, t)を算出する際，各組み合わせにおける対応関係は，s'および t' について必ず排他的でなくてはならない．

・排他的な組み合わせは，s'の数を n，t'の数を m とした時，

・n≦m の時 n 個

・n＞m の時 m 個

の対応関係を組み合わせたものである．これは，子要素数が多い方を n 少ない方を m とした時，順列 nP m 通り存在する．

図 9 に n = 3，m = 2 における全ての排他的な組み合わせ($_3P_2$ = 6 通り)を示す．

[手順 2](a)で得られた全ての排他的な組み合わせの中で，各対応関係の wsim(s', t')を平均した値が最大なものを cssim(s, t)とする．

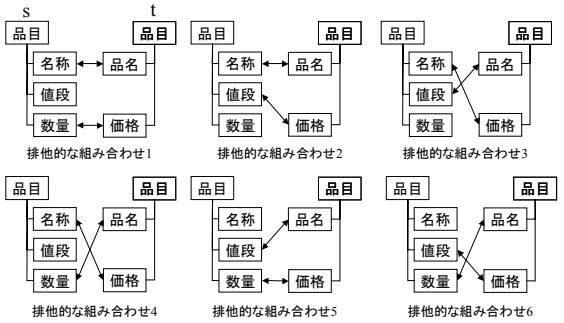


図 9　排他的な組み合わせ(n = 3，m = 2)

Step 3-2 における演算結果を格納した出力マトリックスを図 10 に示す.

wsim

	品番	品名	単価	製品	本文
商品番号	0.357901	0.132006	0.389349	0	0
名称	0.048502	0.740476	0.043345	0	0
価格	0.02102	0.037401	0.430992	0	0
商品	0	0	0	0.417362	0.009136
発注書	0	0	0	0.0303105	0.23863

図 10　演算結果格納後の出力マトリックスの例

3.4.4. 変換ルール作成者の確認

GUI を用いた本手法のイメージを図 11 に示す．変換元のスキーマ S 上で，変換ルール作成者が指定したある要素 s に対する，変換先のスキーマ T 上の変換候補となる要素をランキングし，上位 m 件を提示する (図 5 の場合は m = 3)．変換ルール作成者は候補の中から変換ルール(要素 s と t)を選択して確定する．

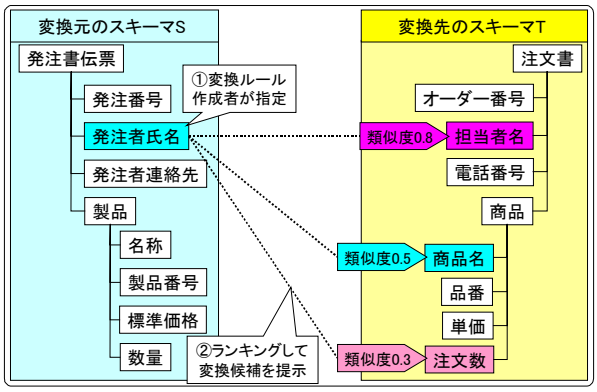


図 11　GUI を用いた本手法のイメージ

4. 評価実験

提案アルゴリズムの有効性を確認するために，プロトタイプを作成し評価実験を行った.

4.1. 評価方法

発注者情報や製品情報等を含む XML の発注書伝票を想定した．変換元として，同等の商品を扱っているオンラインショッピングの web ページより, 10 件のスキーマを作成した．変換先としては，実際に B2B で XML の発注書伝票として用いられているスキーマ 1 件を使用し，重み定数 k_{const} 及び候補提示数 m を変化させ，計 10 通りの変換について以下の観点で評価した.

(1)提示する候補の正当性

変換候補を m 件提示した時，全ての変換ルールについて上位 m 件に正解が含まれる割合(以下，ヒット率)を求める.

(2)変換ルール作成の稼動削減

本方式を適用する場合，しない場合について，変換ルール作成者が変換ルールを作成する稼動を算出し，それらの比率 (以下，稼働比)を求める．稼働比は値が小さい程，稼動が削減されていることを表す.

4.1.1. ヒット率の算出方法

ヒット率を以下の手順で算出する.

《Step 1》変換ルール作成者が変換ルールを判断した結果(以下，正解)を用意する.

《Step 2》ヒット率 P を下記のように算出する

・変換候補の中に正解が含まれている，変換元のスキーマの要素数：V

・変換元のスキーマの全要素数：W

とすると，ヒット率 P は

$$P = V / W$$

である.

4.1.2. 稼働比の算出方法

以下の手順で稼働比を算出する．なお,

・変換元のスキーマの全要素数を N

・変換先のスキーマの全要素数を M

とした時，本稿では，変換ルール作成者が，変換元の 1 つの要素に対して，変換ルールを探す作業を M 回行うことを稼動と仮定する.

《Step 1》稼働の算出

(ⅰ)従来方法(GUI ツール)を用いた場合の稼動 C_A

C_A を次式で算出する.

$$C_A = N \cdot M$$

(ⅱ)変換候補を提示した場合の稼動 C_B

変換候補を提示した場合の稼動 C_B は, 4.1.1.節Step 2 で算出したヒット率 P を用いて, 以下のように算出する.

・変換候補の中に正解が含まれている場合の稼動量 α は，変換候補に正解が含まれていた数の合計 (P・N)に変換候補の提示件数 m を掛ければ算出できるため,

$$\alpha = P \cdot N \cdot m.$$

・変換候補の中に正解が含まれていない場合の稼動 β は, 変換候補に正解が含まれていなかった数の合計((1−P)・N)に，変換先のスキーマの全要素数 M を掛ければ算出できるため,

$$\beta = (1-P) \cdot N \cdot M.$$

従って,

$$C_B = \alpha + \beta$$
$$= \{P \cdot m + M \cdot (1-P)\} \cdot N.$$

《Step 2》稼働比の算出

稼働比 R は上記で得られた稼動 C_A，C_B に対して，次式より算出される．

$$R = C_B / C_A$$
$$= (P \cdot m / M) + 1 - P$$
$$= \frac{P \cdot m + (1-P) \cdot M}{M}$$

4.2. 実験結果

10 通りの変換に対して，ヒット率及び，稼働比を算出した．図 12 に実験に用いた変換元のデータの一例を，図 13 に変換先のデータを示す．

なお，今回の実験使用したデータは，平均で中間要素数が約 10，末端要素数が約 32 であった．

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<発注書>
  <発注書処理番号>00003</発注書処理番号>
  <発注書作成日>2001/3/1</発注書作成日>
  <発注書作成時刻>11:00</発注書作成時刻>
  <発注番号>200</発注番号>
  <発注日>2001/3/5</発注日>
  <指定配送日時>
    <指定配送日>2001/3/9</指定配送日>
    <指定配送時刻>10:00</指定配送時刻>
  </指定配送日時>
  <配送先>
    <名前>電信四郎</名前>
    <郵便番号>345-6789</郵便番号>
    <電話番号>0987-65-4321</電話番号>
  </配送先>
  <担当者情報記入欄>
    <受注担当者コード>512</受注担当者コード>
    <担当者氏名>電信五郎</担当者氏名>
    <担当者氏名半角>ﾃﾞﾝｼﾝｺﾞﾛｳ</担当者氏名半角>
    <担当者所属部署>営業部 1 課</担当者所属部署>
    <所属部署コード>001</所属部署コード>
    </担当者情報記入欄>
  <発注商品リスト>
    <発注商品>
      <商品番号>5790H-120</商品番号>
      <商品名>半袖 T シャツ</商品名>
      <商品販売元>アパレルジャパン(株)</商品販売元>
      <価格>1000</価格>
      <商品区分番号>2263</商品区分番号>
      <ブランド分類コード>10</ブランド分類コード>
      <製造年度>2001</製造年度>
      <シーズン区分>2</シーズン区分>
      <商品備考 1>洗濯不可</商品備考 1>
      <商品備考 2>ドライクリーニング可</商品備考 2>
      <商品詳細情報>
        <サイズ>S</サイズ>
        <色>スカイブルー</色>
        <数>1</数>
      </商品詳細情報>
    </発注商品>
  </発注商品リスト>
  <会計情報>
    <小計額>1000</小計額>
    <消費税額>50</消費税額>
    <合計額>1050</合計額>
  </会計情報>
  <特記事項>
    <発注書作成者>電信四郎</発注書作成者>
    <発注責任者>電信六郎</発注責任者>
  </特記事項>
</発注書>
```

図 12　変換元のデータ

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<本文>
  <データ処理番号>00001</データ処理番号>
  <データ作成日>2001/3/6</データ作成日>
  <データ作成時刻>14:00</データ作成時刻>
  <発注番号>100</発注番号>
  <発注日>2001/3/6</発注日>
  <配達情報>
    <着荷指定日>2001/3/10</着荷指定日>
    <着荷指定時刻>13:00</着荷指定時刻>
  </配達情報>
  <発注者情報>
    <発注者コード>128</発注者コード>
    <発注者名全角>電信一郎</発注者名全角>
    <発注担当者電話番号>
    12-3456-7890
    </発注担当者電話番号>
    <発注者の郵便番号>123-4567</発注者の郵便番号>
  </発注者情報>
  <受注者情報>
    <受注者コード>256</受注者コード>
    <受注者名全角>電信二郎</受注者名全角>
    <受注担当者名半角>ﾃﾞﾝｼﾝｼﾞﾛｳ</受注担当者名半角>
    <受注部課名全角>衣料資材課</受注部課名全角>
    <受注部課コード>001</受注部課コード>
  </受注者情報>
  <製品一覧>
    <製品>
      <品番>SA512</品番>
      <品名>長袖 T シャツ</品名>
      <発売元>アパレル興業(株)</発売元>
      <単価>2000</単価>
      <情報区分コード>8501</情報区分コード>
      <訂正コード>1</訂正コード>
      <ブランドコード>21</ブランドコード>
      <年度>2001</年度>
      <シーズンコード>1</シーズンコード>
      <アパレル製品備考 1>洗濯時色落ち有り</アパレル製品備考 1>
      <アパレル製品備考 2>洗濯時縮み有り</アパレル製品備考 2>
      <マルチ明細>
        <色柄サイズ番号>L</色柄サイズ番号>
        <色柄参照番号>N14</色柄参照番号>
        <色柄サイズ別数量>1</色柄サイズ別数量>
        <色柄別数量の単位>10</色柄別数量の単位>
        </マルチ明細>
    </製品>
  </製品一覧>
  <支払い額>
    <小計>20000</小計>
    <消費税>1000</消費税>
    <総計>21000</総計>
  </支払い額>
  <備考欄>
    <発注書記入者>電信太郎</発注書記入者>
    <発注書確認者>電信三郎</発注書確認者>
  </備考欄>
</本文>
```

図 13　変換先のデータ

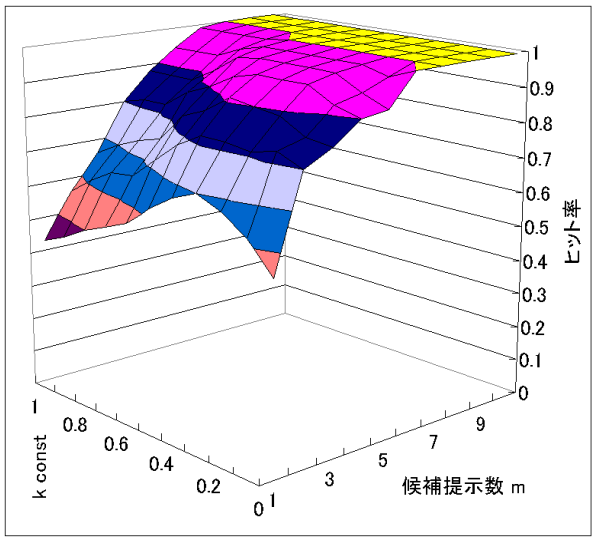


図 14　中間要素同士のヒット率(平均)

図 14 に 10 通りの変換に対する，中間要素同士のヒット率の平均値グラフを示す．中間要素同士のヒット率は候補提示数 m が増す毎に上昇する傾向が得られた．これは，wsim をランキングして提示するため，候補提示数 m を増やせば，正解が変換候補に含まれる可能性が高くなり，それに伴ってヒット率が上昇することを表している．

また，重み定数 k_{const} = 0.2～0.5 においてピークとなる傾向が得られた．これは，wsim を算出する際，子要素集合同士の類似度(cssim)を重視するとヒット率が高くなることを示唆している．

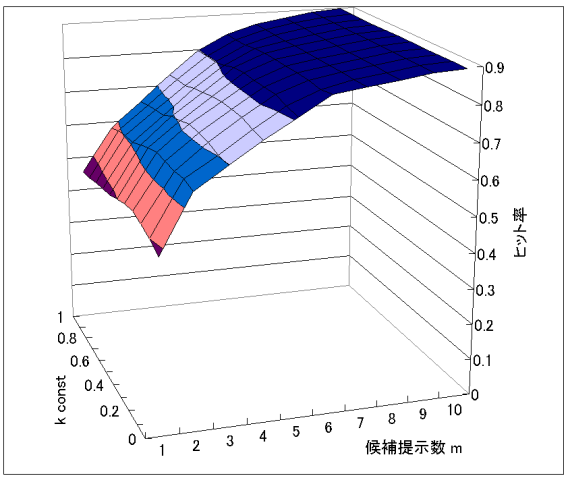


図 15　中間要素と末端要素を合わせたヒット率(平均)

図 15 に 10 通りの変換に対する，中間要素と末端要素を合わせたヒット率(以下，全体のヒット率)の平均値グラフを示す．全体のヒット率も，図 14 と同様に k_{const} = 0.2～0.5 においてピークとなる傾向が得られた．しかし，全体のヒット率には中間要素数の約 3 倍ある末端要素同士のヒット率も含まれているため，重み定数 k_{const} によるヒット率の差が図 14 の様に明確に現れていない．

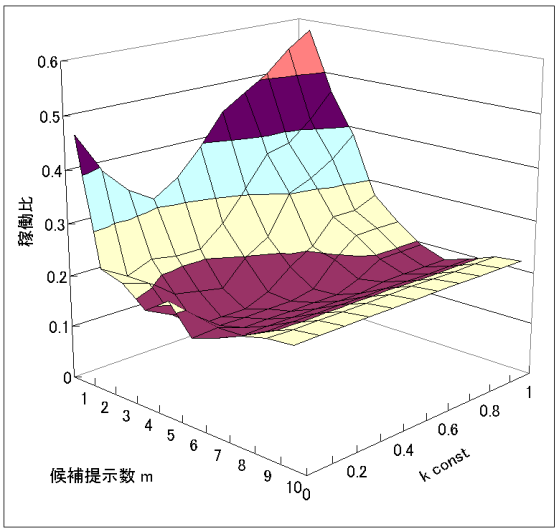


図 16　中間要素同士の稼働比(平均)

図 16 に 10 通りの変換に対する，中間要素同士の稼働比の平均値グラフを示す．中間要素同士の稼働比は候補提示数 m = 4 の時，最適値(0.12～0.27)となった．これは，候補提示数が少なければヒット率も悪くなり，候補以外の要素を判断するため稼動が掛かり，逆に候補提示数が多くなると，正解の有無に関わらず判断すべき要素が多く，稼動が掛かることを表している．また，候補提示数 m = 4 の時，重み定数 k_{const} = 0.2～0.4 において最適値(0.12)が得られた．

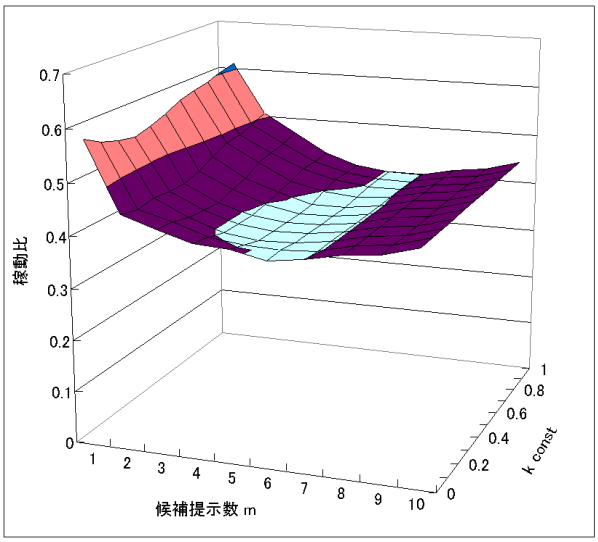


図 17　中間要素と末端要素を合わせた稼働比(平均)

図 17 に 10 通りの変換に対する，中間要素と末端要素を合わせた稼働比(以下，全体の稼働比)の平均値グラフを示す．全体の稼働比は，図 16 と比較して全体的に稼働比が上昇している．これは，中間要素同士よりも稼働比が高い末端要素同士の稼働比も含まれているためである．また，候補提示数 m = 5，重み定数 k_{const} = 0.2～0.5 において最適値(0.38)が得られた．これは，従来方式の稼働と比較して，約 60%の稼働削減が行えることを表している．

5. 考察と今後の課題

5.1. 考察

今回の実験結果から，重み定数 k_{const}，変換候補提示数 m とヒット率，稼働比の関係について考察する．

(1)候補提示数 m とヒット率，稼働比の関係

今回の実験では，全体の稼働比の平均において，候補提示数 m = 5 の時，稼働比が最適値(0.38～0.4)を示し，約 60%の稼動を削減することが可能となった．よって，候補提示数 m = 5 と設定するのが最適と考えられる．要素数約 40 のスキーマ間の変換を行う際，候補提示数を 5 件とすることは，変換ルール作成者が判断を行う量として適していると考えられる．

(2)重み定数 k_{const} とヒット率，稼働比の関係

今回の実験では，全体の稼働比の平均において，重み定数 $k_{const} = 0.2 \sim 0.5$ 付近に設定した場合，ヒット率が高く，稼働比が低い傾向が得られた．

これは，

- 辞書 CB 方式を用いて lsim を算出する際，正解の対応関係とは別の対応関係に高い値が算出されてしまう場合がある
- 辞書に登録されていない単語に関しては，lsim が算出されない場合がある

ため，cssim しか考慮しないことを意味する重み定数 $k_{const} = 0$ や，lsim しか考慮しないことを意味する重み定数 $k_{const} = 1$ に設定すると，正解とは別の変換ルールの wsim に高い値が算出されたり，正解の変換ルールの wsim に低い値が算出されてしまう場合があるためと考えられる．

最適値における重み定数 k_{const} が 0.2～0.5 付近であることからも，タグ名称類似度(lsim)と中間要素が持つ子要素集合同士の類似度(cssim)を組み合わせる本方式の有効性が示せたと考えられる．

5.2. 今後の課題

(1)値の類似性の加味

本稿では，スキーマをベースとした変換候補の提示法を提案しており，3.4.2.節の木構造に基づく類似性において，末端要素同士の類似性はタグ名称のみを判断基準としている．今後は，末端要素が持つ値の傾向の類似性を加えることによって，提示する変換候補の精度向上が期待できると考えられる．

(2)1 要素対 1 要素以外の変換パターンへの対応

本稿では，1 文書対 1 文書変換の中でも，1 要素対 1 要素変換に焦点を当て，その変換ルールの候補を生成するアルゴリズムを提案したが，実際の B2B では，他にも m 要素対 n 要素の変換が頻繁に行われているため，現段階で B2B に適用するにはまだ不十分である．今後は，m 要素対 n 要素の変換ルールの候補生成法に関して，検討を行う必要がある．

(3)wsim の再計算

本稿では，提示した変換候補に対して変換ルール作成者が判断を行う方式を提案したが，変換ルール作成者によって判断が行われた変換ルールを wsim の再計算にフィードバックさせていない．変換ルール作成者が判断した変換ルールは，より正確な wsim を算出するのに有効であると考えられる．そこで，変換元のスキーマの要素 s と変換先のスキーマの要素 t が変換ルールとして変換ルール作成者に確定された場合，

$$wsim(s, t) = 1$$

として 3.4.3.節 Step 3-2 を再実行し，変換ルール作成者に再提示することで提示する変換候補の精度向上が期待できると考えられる．

6. まとめ

本稿では，スキーマの異なる XML 電子伝票の変換に対して，変換ルールの候補を自動的に生成し，変換ルール作成者に選択させる半自動方式を提案した．本方式は，タグ名称類似度と子要素集合の類似度に基づいた変換候補を提示する．タグ名称類似度算出方法に関しては，同じ概念を指す，異なるの表記の複合語に対して，複合語間の意味的な距離を算出可能な辞書 CB 方式を採用した．子要素集合の類似度算出方法に関しては，すべての要素が持つ構造情報を考慮に入れることで，XML の持つ木構造情報を正確に捉えた算出方法を用いていることが特徴である．タグ名称類似度と子要素集合の類似度に基づいた変換候補の中から，変換ルール作成者が変換ルールを選択し確定することによって，変換ルールの正確性が保証された．

また，評価システムを作成し，同業種間でのスキーマ変換において，本方式を用いた場合と従来方式とを比較して，最大約 60%稼動の削減が行えることを確認した．

参考文献

[1]RsettaNet, http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial.

[2]cbXML, http://www.ebxml.org/.

[3]iConnector, http://www.infoteria.com/jp/contents/product/iconnector/.

[4]BizTalkServer, http://www.microsoft.com/japan/biztalkserver/.

[5]DataSpider, http://www.appresso.com/product/dataspider/index.html.

[6]W3C, “XSL Transformations (XSLT) Version 1.0” http://www.w3.org/TR/xslt.

[7]Madhavan, J., P.A. Bernstein, and E. Rahm: Generic Schema Matching with Cupid. VLDB 2001, 49-58.

[8]星野，綱川，町原：DBSENA:マルチデータベース環境における情報資源管理と検索方式，第 114 回情報処理学会データベースシステム研究会 1998.

[9]池田，鈴木，町原，安田：連邦データベースシステムにおけるスキーマ構築の一方式，情報処理学会論文誌 Vol.40 No.SIG 8(TOD 4) 1999.

[10]笠原, 松澤, 石川：国語辞書を利用した日常語の類似性判別，情報処理学会論文誌，Vol. 38, No. 7, 1272-1283, 1997.