

ルール記述に基づく XML データからの意味情報抽出手法

大河原 俊明 † 森嶋 厚行 †† 杉本 重雄 ††

近年の情報技術の急速な発展に伴い、ソフトウェアの実行環境の変化が速くなっている。また、Web サービスなどのソフトウェア間連携が現実のものになろうとしており、ソフトウェアの、動的な環境変化への対応が重要な問題となっている。我々は、これらを背景に、環境変化への対応において重要であるデータ変換の問題に着目し、XML データを対象として、データ変換プログラムの構築の効率化を図るためのフレームワークの研究開発を行っている。一般に、データ変換を実現するためには変換プログラムが必要であるが、変換プログラムの構築は自明ではない。我々のアプローチの特徴の 1 つは、データ変換プログラムの構築にあたって、データの表す意味を表現したクラス図を利用することである。本稿では、XML データからクラス図を抽出するための一手法を提案する。本手法の特徴は、クラス図抽出のためにルール記述を用いることである。

A Method for Extracting Semantic Information from XML Data based on Rule Descriptions

TOSHIAKI OHKAWARA †, ATUYUKI MORISHIMA ††
and SHIGEO SUGIMOTO ††

With rapid development of information technologies in recent years, change of the execution environment of software has become quick. Moreover, the cooperation among software components such as Web service tends to become actual, correspondence to a dynamic environmental change of software becomes one of the crucial problems. We focus on data conversion problem which is important in the correspondence to environmental change against background of these. And we are doing research and development of the framework for construction of data conversion programs which target at XML data for increase in efficiency. Although conversion programs are generally required in order to realize data conversion, construction of data conversion programs is not trivial. One of the features of our approach is using class diagrams expressing the meaning which data's expresses in construction of data conversion programs. In this paper, a method for extracting a class diagram from XML data is proposed. The feature of this method is using the rule description for extraction of class diagrams.

1. はじめに

近年の情報技術の急速な発展に伴い、ソフトウェアの実行環境の変化が速くなっている。また、Web サービスなどのソフトウェア間連携が現実のものになりつつあり、ソフトウェアの、動的な環境変化への対応が重要な問題となっている。我々はこれらを背景に、環境変化への対応において重要な問題の一つであるデータ変換に着目し、研究を進めている。

ここで我々の定義するデータ変換の問題とは、入力としてスキーマ S_A , S_B , およびスキーマ S_A のインスタンス I_A が与えられたとき、出力としてスキーマ

S_B のインスタンス I_B を求めるものである。これを実現するためには、 I_B を求めるための変換プログラムが必要であるが、その開発は一般に自明ではない。これまで、データ変換の問題は、情報統合などの文脈において研究が行われてきていたものの、「時折起こる問題」としてアプローチされることが通常であり、常に起こりうる問題としての注目はそれほど集められてこなかった。

我々は、XML データを対象として、データ変換プログラムを効率良く開発するためのフレームワークの研究開発を行っている¹⁾²⁾³⁾。本フレームワークの特徴は、次の通りである。(1) データ変換プログラムの構築のための形式的モデルに基づくこと。(2) 単純なスキーママッチングに基づくデータ変換⁹⁾以外のデータ変換にも対応するため、意味情報を利用したデータ変換プログラムの構築を行うこと。本稿における意味情報とは、直観的にはデータベース設計における概念

† 筑波大学 図書館情報メディア研究科

Grad. Sch. of Info. and Media Studies, Univ. of Tsukuba.

†† 筑波大学 知的コミュニティ基盤研究センター

RCKC, Univ. of Tsukuba.

```

univ = (depts, people)
depts = (dept*)
dept = (@did:ID, dname)
people = (person*)
person = (@did:IDREF,
          (prof | student))
prof = (@id:ID, name)
person = (@id:ID, name)
  
```

(a)

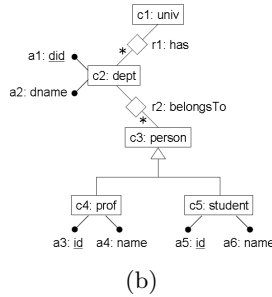


図 1 XML スキーマ S_a (a) とその意味情報 D_a (b)

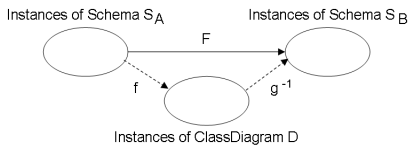


図 2 意味情報を介したデータ変換

モデルの層に対応する。すなわち、実際のデータ表現の詳細を捨象し、概念モデルでどのようなデータを保持しているかを表す情報である。例えば、図 1 (a) で表される XML スキーマの意味情報は図 1 (b) で表すことができる。

我々が提案しているデータ変換モデル⁴⁾では、データ変換を写像 $F: S_A \rightarrow S_B$ としてモデル化する (図 2 実線矢印)。また、本モデルでは XML からクラス図によって表される意味領域への写像 (意味写像と呼ぶ) を導入する。これにより、変換元のスキーマ S_A 、および変換先のスキーマ S_B が与えられたとき、意味写像 $f: S_A \rightarrow D_A$ と $g: S_B \rightarrow D_B$ を用いて、 S_A から S_B への変換は $F: S_A \rightarrow S_B \equiv f \circ g^{-1}$ と表現される (図 2 点線矢印)。ここで、 D_A 、および D_B は意味領域を表すクラス図 (例えば、図 1 (b)) である。

意味領域を介した意味写像による変換には次の利点がある。(1) 意味情報の知識 (オントロジなど) を変換写像の構築に利用できる可能性がある。(2) スキーマ間の直接の対応関係だけからでは自明でない関係を扱いやすい。例えば、図 1 (a) の XML スキーマでは prof 要素と student 要素は独立して存在する。しかし、他のスキーマが同じ情報 (図 1 (b)) を表すとしても同じように表現されるとは限らない。例えば、これらは独立した要素として存在せず、person 要素だけが用意されており、prof と student の区別は属性の値によって行われるかもしれない。このような場合にも、どちらのスキーマからも同じクラス図が抽出できれば、対応関係は明らかになる。

本稿では、上記における意味写像 f を半自動的に

構築する一手法を提案する。我々は、これまでの研究で変換モデルの提案を行い⁴⁾、意味情報抽出支援システムの第一次プロトタイプシステムの開発を行ってきた³⁾。しかしこれまでは、意味情報抽出のために利用されているルールは固定であり、かつ利用されていたアルゴリズムはそれらのルールに特化したものであった。本稿では、意味情報抽出のためのルールの一般的な記述方法、および任意のルールが与えられたときの意味情報抽出手法について提案する。

本稿で提案する手法の概要は次である：(1) 意味写像構築のためのヒューリスティクスを表すルール群を用意し、(2) そのルール群を適用して作成される意味写像 (群) を f の候補とし、列挙する。

本稿の構成は次の通りである。2 章では、我々が開発している意味情報抽出支援システムの概要を説明する。3 章では、本稿で提案する、ルール記述に基づいた意味情報抽出手法について説明する。4 章では関連研究について述べる。5 章ではまとめを行う。

2. XML データからの意味情報抽出支援システムの概要

2.1 意味写像

我々が提案しているモデルでは、データ変換を意味写像の組合せとしてモデル化する。意味写像は $f: S \rightarrow D$ と表す。意味写像 $f: S \rightarrow D$ の定義域は、XML スキーマ S を満たす XML インスタンスの集合 $dom(S)$ である。意味写像 $f: S \rightarrow D$ の値域は、図 1 (b) で表されるようなクラス図によって規定される値の集合 $dom(D)$ である。直観的には、 $v \in dom(D)$ である値 v は、 D 中の各ノード (クラス、関連、および属性) のインスタンス集合を持つタグ付きレコードである。例えば、 $f: S_a \rightarrow D_a$ (S_a は図 1 (a)、 D_a は図 1 (b)) における D_a は、次の形式をしている。ここで、 $c1$ (univ 要素) や $c2$ (dept 要素) など c というラベルが付いたものはそれぞれのノードの ID となっている。

[c1: univ 要素, c2: dept 要素の集合, c3: person 要素の集合, ..., r1: ..., r2: dept 要素と person 要素の関係表, ..., a6: ...]

意味写像 $f: S \rightarrow D$ は、 $w \in dom(S)$ なる XML インスタンス w から値 $v \in dom(D)$ を計算する。具体的には、 f は、 v 中の各ノードに対するインスタンス集合をそれぞれ計算するために、複数の XQuery 風の式で構成される。例えば、意味写像 $f: S_a \rightarrow D_a$ において、 D_a の person クラス ($c3$) のインスタンス

汎化は関連ではない。

集合を計算する式は, for \$k in /univ/dept/person
return \$k as person である .

2.2 意味写像を利用したデータ変換写像の作成

我々の提案するデータ変換モデルにおけるデータ変換写像 $F: S_A \rightarrow S_B$ (図 2 実線矢印 F) の作成は, 次の手順で行われる .

- [1. 意味情報の抽出] 意味写像 $f: S_A \rightarrow D_A$, および $g: S_B \rightarrow D_B$ を作成する . ここで D_A , および D_B は, それぞれ S_A , および S_B の意味情報を表すクラス図である . 例えば, $f: S_a \rightarrow D_a$ は, 大学の情報を表す XML スキーマ S_a (図 1 (a)) のインスタンスをクラス図 D_a (図 1 (b)) のインスタンスに変換するための意味写像 (図 2 点線矢印 f) である .
- [2. マッチング] 一般に, D_A と D_B には不一致 (conflicts) が存在する . そこで f と g をそれぞれ変形し, 意味写像 $f': S_A \rightarrow D'_A$ と $g': S_B \rightarrow D'_B$ を作成する . ここで, D'_A と D'_B はそれぞれ不一致を解消した後のクラス図である .
- [3. 逆写像の作成] $g'^{-1}: D'_B \rightarrow S_B$ を求める .
- [4. 合成] データ変換写像 $F: S_A \rightarrow S_B \equiv g'^{-1} \circ f'$ を求める .

2.3 写像操作系

本データ変換モデルにおける写像操作系は, 写像を操作するための 7 つの操作子 (逆写像 f^{-1} , 意味合成 $f^{-1} \circ g$, ラベル変更, 意味射影, 意味拡張, 値域制限, 名前変更) を持つ . 本稿では, 特に [1. 意味情報の抽出] で重要な操作子である, 意味拡張について説明する . 他の操作子の説明は, 文献⁴⁾にある .

意味拡張: 意味拡張は, 意味写像 $f: S \rightarrow D$ が与えられたとき, D に新たなノード n_i (クラス, 関連, あるいは属性) を追加した D' への写像 $f': S \rightarrow D'$ を作成する操作子である (図 3 (a) ~ (c)) . 具体的には次のように指定する . 意味写像 $f: S \rightarrow D$ が与えられたとする . n_i をクラス c , 関連 $r(c_1, \dots, c_m)$, 属性 a のいずれかとし, e を導出の方法を指定するパラメータとする . このとき, f' は操作子 γ を用いて $\gamma_e^{n_i}[f]: S \rightarrow D'$ で求められる . 本操作系では, 式の可読性を高めるために, γ の代わりに C, R, A と表記し, 追加するノードの種類を明示する . 以下では, 本稿に現れるいくつかの操作子についてのみ説明する . $C_{c_1}^{c_2}[f]$ は, 選択条件 p を用いてクラス c_1 からサブクラス c_2 を導出する (図 3 (a)) . $A_{c_1, a_1}^{c_2, a_2}[f]$ はクラス c_1 の属性 a_1 をクラス c_2 の属性 a_2 と

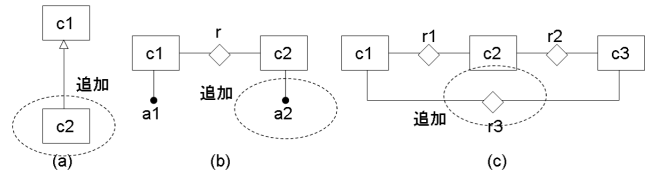


図 3 意味拡張操作子による D から D' への変更

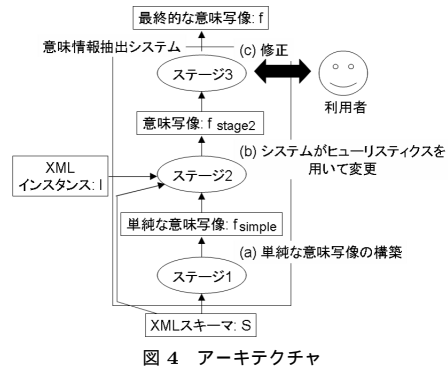


図 4 アーキテクチャ

してコピーする (図 3 (b)) . $R_{r_1(c_1, c_2), r_2(c_2, c_3)}^{r_3(c_1, c_3)}[f]$ は関連の合成を作成する (図 3 (c)) .

2.4 意味情報抽出支援システムの概要

本システムは, [1. 意味情報の抽出] を支援するシステムである . すなわち, XML スキーマ S からクラス図 D を導出し, 意味写像 $f: S \rightarrow D$ を作成することを支援する .

2.4.1 アーキテクチャ

本システムのアーキテクチャを図 4 に示す . 基本的な考え方は, 意味情報抽出のうち自動化できる部分はシステムが行い, それ以外は人が実行できるような環境を提供することである . 意味情報抽出は次の 3 段階で行われる .

- (ステージ 1) 入力として XML スキーマ S を受け取り, 単純な意味写像 f_{simple} を出力する (図 4 (a)) . ここで, 単純な意味写像 $f_{simple}: S \rightarrow D$ の D は S と同型の構造を持つ . すなわち, S 中の各 XML 要素に対して D 中の一つのクラスが対応し, S 中の要素の親子関係が D の関連に直接対応する . ステージ 1 の出力は図 5 である .
- (ステージ 2) 入力として f_{simple} と XML インスタンス I を受け取る . ヒューリスティクスを用いて写像操作子を f_{simple} に適用し, 新たな意味写像 f_{stage2} を作成し, 出力する (図 4 (b)) .
- (ステージ 3) ステージ 2 で作成される意味情報は必ずしも十分であるとは限らない . 例えば, 図 1 (b) の dept 要素と person 要素間の関連の名前が be-

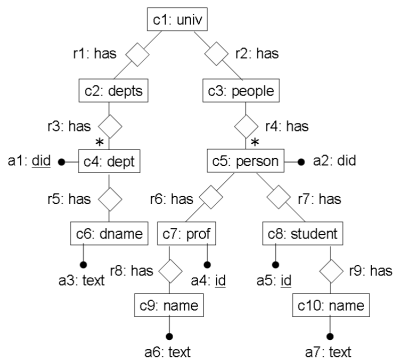


図 5 ステージ 1 の出力

longsTo であるということは、システムにはわからない。そこで、ユーザが写像操作子を f_{stage2} に適用し (図 4 (c)), 最終的な意味写像 $f : S \rightarrow D$ を得る。

3. ルール記述に基づいた意味情報抽出

ここでは、2 章で述べたステージ 2 の詳細を説明する。この 3 章で説明するルール記述に基づく意味情報抽出手法が本稿のメインであり、これまでの我々の研究³⁾との差異である。

例として、図 1 (a) の XML スキーマから図 1 (b) の意味情報を導出する場合を考える。

3.1 意味情報抽出のためのルール記述

ルールの記述は if 節, derive 節, dependencies 節から構成される。if 節にはルール適用の条件となる述語を記述し、これが成立した場合、derive 節に記述される意味情報の新しい構成要素と、その構成要素を含んだ値域を持つ新しい意味写像を作成する。derive 節の as の後ろには、新しい構成要素をどのようにして導出するかを記述する。また、dependencies 節には if 節に記述した条件が成立した場合の、構成要素間の導出関係を記述する。導出関係とは、左辺に記述された構成要素があれば、右辺の構成要素を導き出せる、という関係である。

例として、図 6 で示されるヒューリスティクスを表すルールを説明する。このヒューリスティクスは次のようなルールである。XML においてクラスの属性に対応する情報は、しばしば要素属性ではなく独立した要素として表現される。したがって、XML 要素 (図 6 (c2)) がテキストだけを持ち (図 6 (a1)), 子要素や属性を一切持たなければ、その要素に対応するクラスの内容をそのクラスと 1 対 1 関連で接続される他のクラス (図 6 (c1)) の属性としてコピーする (図 6 (a2))。これは、操作子 $A_{c_2, a_1}^{c_1, a_2}[f]$ を用いて行う (図 3 (c))。

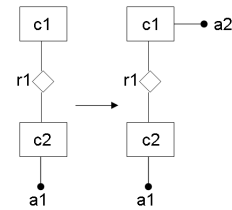


図 6 テキスト要素をそのクラスの親クラスの属性としてコピー

述語	意味
class(c)	c というクラスが存在
rel(c1, c2, r)	クラス c1 とクラス c2 間の r という関連が存在
attr(c, a)	クラス c の a という属性が存在
isa(c1, c2)	クラス c1 is-a クラス c2 が成り立つ関係が存在
name(γ , "...")	要素 γ (クラス, 関連, あるいは属性) の名前は ... である
type(a, type)	属性 a のタイプ (ID, IDREF, etc.) は type である
relation(γ)	要素 γ (クラス, 関連, あるいは属性) のリレーション (表) が存在
multiplicity(m, c, r)	ある関連 r のクラス c 側の多重度は m である
participation(c, r)	あるクラス c が関連 r に対して参加制約を満たしている
exclusion(r1, r2, ..., rm)	ある関連 r1 ~ rm が排他制約を満たしている
hasOneBaseNode(c, node)	あるクラス c は一つのベースノード node を持つ
hasTwoBaseNode(c, node1, node2)	あるクラス c は 2 つのベースノード node1, node2 を持つ
hasNoAttr(c)	あるクラス c は属性を持たない

表 1 述語の意味

図 7 はこのヒューリスティクスをルールによる記述で表したものである。if 節には、条件として、(1) 関連 r1 で接続されているクラス c1, c2 があり、c2 はテキストノード a1 しか持っていない、(2) r1 は 1 対 1 関連である、(3) c2 と a1 は XML スキーマに明示的に現れている、という条件を記述する。if 節の条件が成立した場合、derive 節に記述したクラス c1 の属性 a2 という、クラス c2 の属性 a1 から導出された構成要素が追加され、その構成要素のインスタンスが $\pi_{ID, v}(r1 \bowtie a1)$ で計算される、意味写像を得ることができる。ここで、v は要素属性の値を格納するリレーショナル属性名として、本モデルで利用しているものである。dependencies 節には、例えば、クラス c1, クラス c2, 関連 r1, 属性 a2 から属性 a1 が導出可能である、ということを記述する。

本手法で用いる述語を表 1 で示す。表中のベースノードとはステージ 1 で出力されるノードである。つまり、XML スキーマに明示的に存在する要素である。

if:

```
class(c1)
class(c2)
rel(c1, c2, r1)
attr(c2, a1)
type(a1, text)
multiplicity(1, c1, r1)
multiplicity(1, c2, r1)
hasOnlyOneBaseNode(c2, a1)
```

derive:

```
attr(c1, a2) as  $\pi_{ID,v}(r1 \bowtie a1)$ 
```

dependencies:

```
c1c2r1a1  $\rightarrow$  a2
c1  $\rightarrow$  c2
c1c2  $\rightarrow$  r1
c1c2r1a2  $\rightarrow$  a1
```

図 7 図 6 におけるルール記述

3.2 3 段階意味情報抽出

本手法におけるステージ 2 は、以下で説明するような 3 段階の操作から構成される。まずフェイズ 1 で、新しい意味情報の構成要素を発見し、意味写像を作成し、導出関係を導き出す。次にフェイズ 2 で、フェイズ 1 で得られた導出関係から minimal cover を求める。最後にフェイズ 3 で、フェイズ 2 で複数の minimal cover が得られた場合のランキングを行う。以下で詳細に説明する。

3.2.1 フェイズ 1

フェイズ 1 では、ルールを適用し、新しい意味情報の構成要素（クラス、関連、および属性）の発見を行い、意味写像を作成する。ルールにより一つの構成要素が発見されるたびに、また 1 からルールの適用を繰り返す。新しい意味情報の構成要素が発見されると、それに伴い、その構成要素間の導出関係も導き出される。構成要素の発見ができなくなったらフェイズ 1 の終了である。

3.2.2 フェイズ 2

フェイズ 2 では、フェイズ 1 のルール適用により導き出された導出関係に基づき、必要な構成要素を導出するための minimal cover を求める。具体的には、dependencies 節の右辺に記述された構成要素が極小になるような minimal cover を求める。必要な構成要素とは、導出関係を利用することで、その構成要素だけでフェイズ 1 で発見したすべての構成要素を導出で

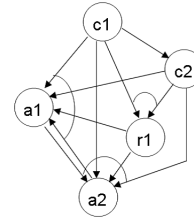


図 8 図 7 の導出関係の and-or グラフ

きるような構成要素である。複数の minimal cover が得られた場合には、それぞれを意味写像 f の候補とする。

フェイズ 2 の操作を行うアルゴリズムを図 9 で示す。このアルゴリズムは、導出関係を and-or グラフとして表したものを入力とする。例えば、図 7 の dependencies 節に記述される導出関係を and-or グラフ G で表すと、図 8 となる。出力はこのグラフのノード $Nodes(G)$ の部分集合 Ans である。

図 9 (a) のアルゴリズムは、以下の手順で出力となるノードの集合を求める。(1). and-or グラフが入力として与えられると、まず導出関係には現れず独立して存在するノードの集合を除去する (図 9 (a)2 行目)。(2). 矢印が来ないノードを除去する (図 9 (a)5~7 行目)。(3). (b) の minimal cover を求めるアルゴリズムを適用する (図 9 (a)8~10 行目)。(4). ノードの集合 Ans を得る (図 9 (a)9 行目)。

図 9 (b) のアルゴリズムは、以下の手順で出力となる minimal cover を求める。(1). $\mathcal{P}(Node(G))$ における各集合 N に対して、 N が導出関係 F を用いることで N^+ を導けるかを調べる (図 9 (b)3~9 行目)。 N^+ とは G 中に存在するすべてのノードである。(2). (1) が true のとき、関連端がない関連がないか、クラスに属さない属性がないかのチェックを行う (図 9 (b)5 行目)。(3). (2) が true なら N を minimal cover とする (図 9 (b)6 行目)。

3.2.3 フェイズ 3

フェイズ 2 で求める minimal cover は、一般には一意には決まらない。そのためにその候補群がある基準によりランキングする。この操作を行うのがフェイズ 3 である。ランキングを行うためのスコアリングは表 2 に基づいた減点法で行う。例えば、クラスが一つ存在するごとに 3 減点する。このランキングにおいては、スコアが高い方がランクが高くなる。

3.3 例への適用

まず、図 1 (a) の XML スキーマを入力として、ステージ 1 の操作を行う。ステージ 1 の出力は図 5 である。

```

1 Set (Set Nodes)
  Phase2(Graph G, Set Dependencies F) {
2   Ans := G 中の独立して存在するノードの集合;
3   G0 := G - Ans;
4   Ans' := G0;
5   for each x s.t. x は G0 中の矢印が来ないノード {x} {
6     Ans' := Ans' - {x};
7   }
8   for each mc ∈ minimal_covers(G0 ∪ Ans', G, F) {
9     Ans = Ans ∪ mc;
10  }
11 }

```

(a)

```

1 Set (Set Nodes)
  minimal_cover(Graph G1, Graph G, Dependencies F)
  {
2   Ans := empty;
3   for each Set N ∈ P(Node(G)) {
4     if(N+ on F = Node(G)) {
5       if(関連に矛盾がない(N) ∧ 属性に矛盾がない(N))
6         Ans := Ans ∪ {N};
7     }
8   }
9   }
10  return Ans;
11 }

```

(b)

図 9 フェイズ 2 のアルゴリズム (a) と minimal cover を求めるアルゴリズム (b)

減点対象	減点数
クラス	3
サブクラス	0
属性	2
キー属性	0
関連	1

表 2 スコアリング

フェイズ 1: フェイズ 1 の操作を行うと、ステージ 1 で出力される意味情報に、新しい意味情報の構成要素が追加され、出力として図 10 が得られる。また、図 11 に示すように、構成要素間の導出関係も得られる。

フェイズ 2: フェイズ 1 で出力された導出関係における minimal cover を、図 9 のアルゴリズムにより求めると次のような組合せが得られる。ここで $(x | y)$ という記述は、 x が y のどちらかであることを示す。したがって、この例では、minimal cover は $2^5 = 32$ 通りになる。

$c1c4c5c7c8a1a4a5(a8 | c6a3)(a9 | c9a6)$
 $(a10 | c10a7)(r10 | c2r3)(r12 | a2)$

フェイズ 3: フェイズ 2 で出力された minimal cover を図 2 に基づきスコアリングする。この例において、スコアが最高となるのは次であり、これをクラス図として示したものが図 1 (b) である。

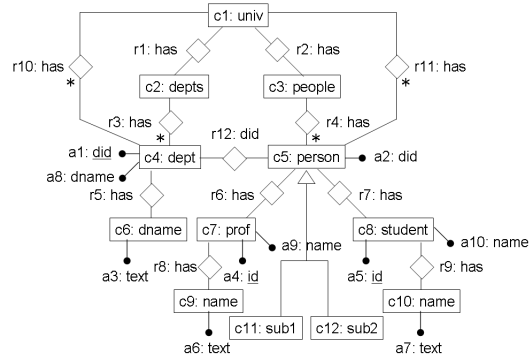


図 10 フェイズ 1 の出力

```

c4c5a1a2 → r12
c4c5r12a1 → a2
c4c6r5a3 → a8
c4 → c6
c4c6 → r5
c4c6r5a8 → a3
c7c9r8a6 → a9
c7 → c9
c7c9 → r8
c7c9r8a9 → a6
c8c10r9a7 → a10
c8 → c10
c8c10 → r9
c8c10r9a10 → a7
c1c2c4r1r3 → r10
c1c2 → c2
c1c2 → r1
c1c2c4r1r10 → r3
c1c3c5r2r4 → r11
c1 → c3
c1c3 → r2
c1c3c5r2r11 → r4
c1c4c5r10r12 → r11
c1 → c4
c1c4 → r10
c5 → c7c8c11c12

```

図 11 フェイズ 1 によって導き出された導出関係

$c1c4c5c7c8r10r12a1a4a5a8a9a10$

4. 関連研究

XML から UML のクラス図や ER 図などの概念モデルを抽出する研究はいくつか行われている⁶⁾⁷⁾⁸⁾が、これらは、(1) 変換規則が固定である、(2) インスタンスの変換を考慮していない、という点で本手法と異なる。特に (1) の理由により、これらの手法の設計時に想定されていないようなスキーマを対象とした場合に対応することは、簡単ではない。一方、本手法ではルールの追加は比較的容易にできる。

データベースを対象としたリパースエンジニアリングの支援が可能な CASE ツールも存在する⁵⁾が、これは特に自動的なリパースエンジニアリングを想定したものではない。

5. おわりに

本稿では、データ変換プログラムの構築支援を目的とした、XML データからの意味情報抽出の一手法を

提案した。本手法の特徴は、意味情報抽出の際に利用されるルールを、ユーザなどが自由に追加できることである。今後の課題としては、共通に利用可能なルールを集めたルールリポジトリの構築や、本手法の有効性を示すためのベンチマークの開発などが挙げられる。

謝 辞

ゼミなどでご議論いただきました筑波大学図書館情報メディア研究科田畑孝一教授、阪口哲男助教授、永森光晴講師に感謝いたします。本研究の一部は日本学術振興会科学研究費補助金若手研究 (B)(課題番号 15700108) による。

参 考 文 献

- 1) 古川夏子, 森嶋厚行. 意味情報を用いた XQuery 問合せ作成支援システムの開発. 情報処理学会第 65 回全国大会講演論文集 (3), 2003 年 3 月.
- 2) 古川夏子, 森嶋厚行. 情報統合を目的とした XML の意味情報の抽出. 日本データベース学会 Letters, Vol.2, No.2, 2003 年 10 月.
- 3) 古川夏子, 上村匡稔, 大河原俊明, 森嶋厚行, 杉本重雄. XML データからの意味情報抽出支援プロトタイプシステムの実装. 日本データベース学会 Letters, Vol.3, No.1, 2004 年 6 月.
- 4) 森嶋厚行. データ変換プログラムのモデル化と構築手法の提案. 日本データベース学会 Letters, Vol.3, No.1, 2004 年 6 月.
- 5) Jean-Luc Hainaut, Jean Henrard, Didier Roland, Vincent Englebert, Jean-Marc Hick: Structure Elicitation in Database Reverse Engineering. WCRE 1996: 131-140.
- 6) Mikael R. Jensen, Thomas H. Moller, Torben Bach Pedersen: Converting XML DTDs to UML diagrams for conceptual data integration. Data Knowl. Eng. 44(3): 323-346 (2003).
- 7) Ronaldo dos Santos Mello, Carlos A. Heuser: A Rule-Based Conversion of a DTD to a Conceptual Schema. ER2001: 133-148.
- 8) Murali Mani, Dongwon Lee, Richard R. Muntz: Semantic Data Modeling Using XML Schemas. ER 2001: 149-163.
- 9) L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernandez, R. Fagin: Translating Web Data. VLDB 2002: 598-609.