

情報統合支援を目的としたXMLの意味情報の抽出

Extracting Semantic Information from XML Data to Support Information Integration

古川 夏子[▼] 森嶋 厚行^{*}

Natsuko FURUKAWA Atsuyuki MORISHIMA

XMLはデータ交換のためのデファクトスタンダードとしての地位を既に確立している。XMLデータの量が劇的に増加するにしたいが、情報統合やデータ再構成の問題の重要性が増大するのは明らかである。我々はリバースエンジニアリング的な発想で、既存のXMLデータの情報統合の支援を行う問題に焦点を当てる。具体的には、XMLのスキーマやインスタンスから高水準の意味情報を作成する手法を提案する。

XML has become the de facto standard for data interchange on the Internet. As the amount of XML data grows, there are increased needs for information integration and restructuring. We address the problem of supporting such processes by “reverse engineering” XML data; This paper proposes a framework for extracting high-level semantic information from XML schemas and instances.

1. はじめに

XMLはインターネットにおけるデータ交換のためのデファクトスタンダードとしての地位を既に確立している。現実には作成・管理されるXMLデータの量が劇的に増加するにしたいが、次のような問題の重要性が増すことは明らかである。

(1) 情報統合の問題: 異なる組織が類似の種類XMLデータを個別に持っているが、それぞれスキーマ(DTDやXML-*Schema*など)が異なる。これらを統合したい。

(2) データ再構成の問題: 情報システムの移行や拡張にとともに、XMLのスキーマを変更したい。また新旧スキーマ間のマッピングとそれらの間でデータの移行を行いたい。

これらの情報統合やデータの再構成を行うには、オントロジなどの意味情報の利用が有効であることが知られている[2][4][7]。しかし、XMLの意味情報は、リレーショナルデータベースやオブジェクト指向データベースなどに比べ、同種の情報を表すスキーマのバリエーションがより多くなるため、一般に問題が複雑になる。その理由として、(1)正規化の理論やスキーマの設計論が確立していないこと、(2)要素の入れ子構造や選択構造など多様な構成子を持つこと、(3)クラスなどの概念がなく、意味情報に1対1対応しないことなどの要因があげられる。

例えば、図1のUMLクラス図で表される情報をXMLで表現する場合を考える。このとき、例えば図2(a)(b)のようなスキーマ

で表現することが可能である。これらは同じ情報をかなり異なる方法で表現している。たとえば、(a)は教員と学生を異なる型の要素で区別しているのに対し、(b)では同じ型の要素を属性を用いて区別している。

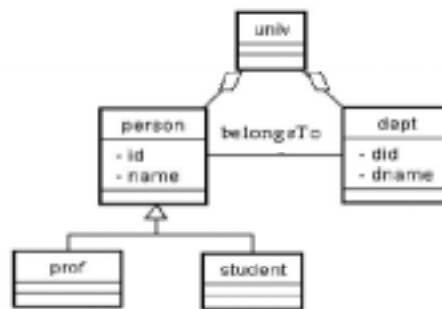


図1 UMLクラス図の例

Fig. 1 A UML class diagram

univ=(dept*)	univ=(dept*, person*)
dept=(did, dname, people)	dept=@(did: ID, dname)
people=(prof*, student*)	person=(pid, @type, name,
prof=(pid, name)	@did: IDREF)
student=(sid, name)	
(a)	(b)

図2 図1のクラス図のための2種類のスキーマ

Fig. 2 Two Different Schemas for the Class Diagram in Fig. 1

我々はリバースエンジニアリング的な発想で、既存のXMLデータの統合や再構成の支援を行うフレームワークの問題に焦点を当てる。本フレームワークでは、既存のXMLスキーマやインスタンスから図1のような意味情報を作成する。我々は、このようなフレームワークの鍵となる技術としてCX図(Class-XML mapping Diagram)を提案する。CX図は、XMLの意味情報がXMLインスタンスにどのようにマッピングされているかを表す表現である。情報統合やデータ再構成などの応用を想定しており、XMLとのマッピング情報を維持したままCX図の操作を行うオペレータが定義されている。本稿では、CX図とそのオペレータ、CX図の作成支援システムについて説明する。このアプローチを用いたデータ再構成については[8]で議論している。

2. Class-XML mapping Diagram

CX図の構成要素はUMLのクラス図のサブセットであり、クラス、関係(関連、汎化)、属性から構成される。集約は名前“has”を持つ関連として表現される。これらの各構成要素には、XSBE(XML-Semantics Binding Expression)が付加されている。これは、各構成要素とXMLインスタンスの関係を表現する式である。図3はCX図の例である。

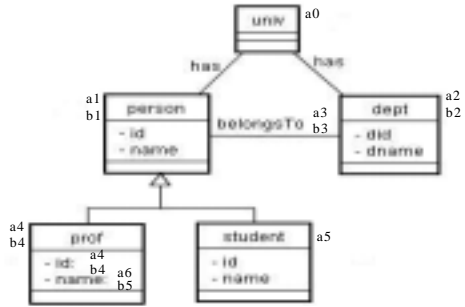
図3の(a)、(b)はそれぞれ図2(a)、(b)のスキーマで規定されるXMLデータと意味情報を結びつけるXSBEである。XSBEは各クラス、関連、属性に付加される¹。本稿では、クラスのXSBEはクラスの脇に記述する。関連のXSBEは、関連の横に並べて記述している。属性のXSBEは、各属性の横に記述している。UMLのクラス図では汎化においては上位クラスからの継承される属性を省略するが、CX図では全て記述する。

一般にXSBEは図4の構文を持ち、かつ付加対象の種類(クラ

[▼] 学生会員 芝浦工業大学大学院工学研究科
m102186@sic.shibaura-it.ac.jp

^{*} 正会員 筑波大学知的コミュニティ基盤研究センター
amorishima@acm.org

¹ 図では一部の構成要素に対してのみXSBEを記述している。



a0:\$j_1\$ in /univ a1:a4 union a5 a2:\$k_1\$ in /univ/dept/did
 a3:a7 union a8 a4:\$k_2\$ in /univ/dept/people/prof/pid
 a5:\$k_3\$ in /univ/dept/people/student/sid
 a6:\$x_1\$ in /univ/dept/people/prof,\$v_1\$ in \$x_1\$/name, \$k_2\$ in \$x_1\$/pid
 a7:\$x_2\$ in /univ/dept, \$k_1\$ in \$x_2\$/did, \$k_2\$ in \$x_2\$/people/prof/pid
 a8:\$x_3\$ in /univ/dept, \$k_1\$ in \$x_3\$/did, \$k_3\$ in \$x_3\$/people/student/sid
 (a)

b1: \$k_1\$ in /univ/person/pid b2: \$k_2\$ in /univ/dept/did
 b3: \$x_1\$ in /univ/person, \$k_1\$ in \$x_1\$/pid, \$k_2\$ in univ/dept/@did
 where \$x_1\$/@did = \$k_2\$
 b4: \$x_2\$ in /univ/person, \$k_3\$ in \$x_2\$/pid, where \$x_2\$/@type = prof
 b5: \$x_3\$ in /univ/person, \$v_1\$ in \$x_3\$/name, \$k_3\$ in \$x_3\$/pid,
 where \$x_3\$/@type = prof (b)

図3 CX 図の例

Fig. 3 Examples of CX diagrams

ス、関連、属性など)によって、記述に制約がある。各XSBEはリレーションを表現する。例えば、図3(a)の関連 belongsTo のXSBE a3が表すリレーション(R(a3)と表記)は\$Sk_1, Sk_2\$に対応する属性を持つリレーションである²。XSBEのunion演算は各XSBEが表すリレーションが和両立で無ければならない。

```
<xobe> ::= (<xobe>) union (<xobe>)
          | <expr> { , <expr> } [ <whereExpr> ]
<expr> ::= <var> in [ <var> ] <locationPath>
<whereExpr> ::= where <predicates>
```

図4 XSBE の構文

Fig. 4 XSBE's syntax

付加対象の種類ごとのXSBEの制約は次の通りである。

クラスのXSBE. (制約1) 変数として、ID変数(\$k_j\$ もしくは \$j_j\$) を唯一つ持つ。ここで\$Sk_j\$はこのクラスのインスタンスのキー値に束縛される変数である。図3(a)において、クラス profのXSBEはID変数 \$Sk_2\$ を持つ。また\$j_j\$は、このクラスのインスタンスのアイデンティティ要素に束縛される変数である。ここでアイデンティティ要素とは、クラスの各インスタンスと1対1対応する要素である。例えば、図3において、a0の\$j_1\$はクラス univのアイデンティティ要素 univに束縛される。(制約2) リレーション Rのリレーション属性のうち、ID変数に対応するものをID(R)と表記する。このとき、あるクラスのXSBE b から求められるリレーション $\pi_{ID(R(b))}(R(b))$ は、そのクラスのインスタンスすべてのID値(キー値もしくはアイデンティティ要素)の集合となる。例えば、図3(a)においてクラス profのリレーション $\pi_{sk_2}(R(a4))$ は、pidの集合となる。

関連のXSBE. (制約1) 変数として、接続されたクラス全てのID変数を持つ。(制約2) ある関連のXSBE b から求められるリレーション $\pi_{ID(R(b))}(R(b))$ の各タプルは、接続されたクラス間

の関連のインスタンスである。例えば、図3(a)の関連 belongsTo のリレーションは、 $\pi_{sk_2, sk_1}(R(a3))$ となる。このリレーションにタプル (k,I) が含まれる時、k で表されるpersonがI で表されるdeptに所属することを表す。

属性のXSBE. (制約1) 変数として、属性が所属するクラスのID変数と、値変数(\$v_j\$)を一つ持つ。(制約2) リレーション Rのリレーション属性のうち、値変数に対応するものをV(R)と表記する。このとき、クラスcの属性aのXSBE b から求められるリレーション $\pi_{VD(R(b)), V(R(b))}(R(b))$ は、タプル(cのID値,aの値)の集合である。例えば、図3(a)のクラス profの属性 nameは、ID変数\$Sk_2\$と値変数\$v_1\$を持ち、XSBE a6のリレーションは $\pi_{sk_2, v_1}(R(a6))$ となる。このリレーションにタプル(m,n)が含まれる時、m で表されるprofのnameがn である事を表す。キー属性の場合は、 $ID(R(b)) = V(R(b))$ となるため、XSBEが表すリレーションは単項リレーション $\pi_{VD(R(b))}(R(b))$ とする。

CX図は次のように定義される。

定義. CX図は4つ組(N,R,name,b)である。ここでN=C∪A, Cはクラス集合、Aは属性集合、R:N×Nはクラス間の関係の集合である。name:(N∪R)→Stringはクラス、属性、および関係の名前をあらわす関数である。また、b:(N∪R)→BEは各ノードおよび関連に付随したXSBEを求める関数である。BEは図4で規定されたXSBEの集合である。

3. オペレータ

本節では、CX図のオペレータについて説明する(図5)。これらを適用すると、CX図の構造とXSBEが変更される。

主な目的	演算子(*は複合演算子)
クラスの操作	SupC, SubC, RC
関連の操作	CR, UR, NR, RR
属性の操作	CA, RA
その他	Key, RN, A2R, C2A*

図5 オペレーター一覧

Fig. 5 Operators

以下の説明において、 $xcd = (N, R, b, name), N = C \cup A$ とする。

3.1 クラスの操作

SupC 演算. 与えられたxcdに関して、 $c_1, c_2 \in C$ はクラスであるとする。SupC(c_1, c_2)(xcd)は、xcdにR(c') = R(c₁)∪R(c₂)となるようなクラスを追加する演算である(図6)。形式的には、(N',R',b',name') = SupC(c_1, c_2)(xcd)と定義する。ただし、

$$N' = C' \cup A'$$

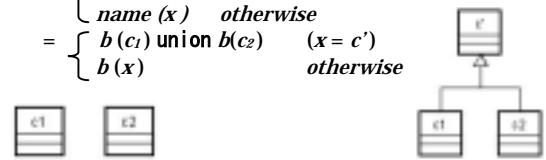
$$C' = C \cup \{c'\}$$

$$A' = A$$

$$R' = R \cup \{(c_1, c'), (c_2, c')\}$$

$$name'(x) = \begin{cases} \epsilon & (x = c') \\ isa & (x = (c_1, c') \text{ or } x = (c_2, c')) \\ name(x) & otherwise \end{cases}$$

$$b'(x) = \begin{cases} b(c_1) \text{ union } b(c_2) & (x = c') \\ b(x) & otherwise \end{cases}$$



(a) オペレータ適用前 (b) オペレータ適用後

図6 SupC 演算例

Fig. 6 SupC operation

² XSBE に union 演算がある場合は最初のオペランドの変数名がリレーションの属性名となる。

SubC 演算. 与えられた $cxid$ に関して $c \in C$ はクラスであるとする。また、 p を条件式とする。 $\text{SubC}_{(c,p) \rightarrow c'}(cxid)$ は、 $cxid$ に $\mathbf{R}(c') = \text{op}(\mathbf{R}(c))$ となるようなクラスを追加する演算である。

RC 演算. 与えられた $cxid$ に関して $c \in C$ はクラスであるとする。 $\text{RC}_c(cxid)$ は、 $cxid$ から c および c に接続された関連および属性を除去する演算である。

3.2 関連の操作

CR 演算. 与えられた $cxid$ に関して、 $r_1 = (c_1, c_2), r_2 = (c_2, c_3) \in R$ は関連であるとする³。 $\text{CR}_{(r_1, r_2) \rightarrow r'}(cxid)$ は、 $cxid$ に $\mathbf{R}(r') = \pi_{\neg \text{common}(\mathbf{R}(r_1), \mathbf{R}(r_2))}(\mathbf{R}(r_1) \bowtie \mathbf{R}(r_2))$ となるような関連を追加する演算である(図7)。ここで、 $\neg \text{common}(S, T)$ はリレーション S, T の属性のうち、 S, T の双方には現れないものである。形式的には、 $(N', R', b', name') = \text{CR}_{(r_1, r_2) \rightarrow r'}(cxid)$ と定義される。ただし、

$$N' = C \cup A$$

$$R' = R \cup \{(c_1, c_3)\}$$

$$name'(x) = \begin{cases} \varepsilon & (x = (c_1, c_3)) \\ name(x) & otherwise \end{cases}$$

$$b'(x) = \begin{cases} Join(b(r_1), b(r_2), p) & (x = (c_1, c_3)) \\ b(x) & otherwise \end{cases}$$

ここで、 $Join(b_1, b_2, p)$ は $\mathbf{R}(r_1)$ と $\mathbf{R}(r_2)$ を条件 p で結合するXSBEである。 p は、 $\mathbf{R}(r_1)$ と $\mathbf{R}(r_2)$ に共通のID変数 k_i に関する $K_i = K'_i$ の並びである。

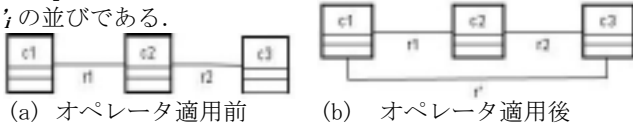


図7 CR 演算例
Fig. 7 CR operation

UR 演算. 与えられた $cxid$ に関して $c_1, c_2 \in C$ はクラスであり、 s は関連の名前を表す文字列とする。 $\text{UR}_{s \rightarrow (c_1, c_2, r)}(cxid)$ は、 $cxid$ に $\mathbf{R}(r') = \cup_{r_i \in R_s} \mathbf{R}(r_i)$ となるような c_1 と c_2 間の関連 r' を追加する演算である。ここで、 R_s は、名前前に s を持つ関連 $r_s = (c_k, c_l)$ の集合である。 c_k は、 c_1 もしくはそのサブクラスであり、 c_l は c_2 もしくはそのサブクラスである。

NR 演算. 与えられた $cxid$ に関して $c \in C$ はクラスであるとする。 $\text{NR}_{c \rightarrow r'}(cxid)$ は、 c に接続されるすべてのクラスを結ぶ n 項関連 r' を作成し $cxid$ に追加する(図8)。ここで、 $\mathbf{R}(r') = \pi_{\text{common}(\mathbf{R}(r_i))}(\cup_{r_i \in R_c} \mathbf{R}(r_i))$ となる。 R_c は c に接続する関連の集合である。

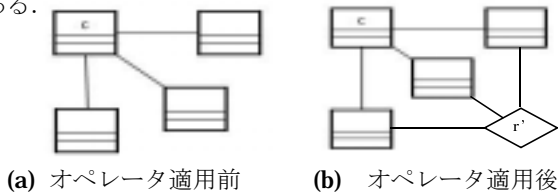


図8 NR 演算例
Fig. 8 NR operation

RR 演算. 与えられた $cxid$ に関して $r \in R$ とする。 $\text{RR}_r(cxid)$ は、 $cxid$ から r を除去する演算である。

3.3 属性の操作

CA 演算. 与えられた $cxid$ に関して $c_1 \in C$ はクラス、 $a \in A$ は属性であるとする。 a が属するクラス c_2 は、 c_1 と1対1関連 $r \in R$ で接続されていなければならない。この時、 $\text{CA}_{a \rightarrow (c_1, a)}(cxid)$ は、 a を c_1 の属性 a' としてコピーする演算である(図9)。こ

で、 $\mathbf{R}(a') = \pi_{K(c_1), V(a)}(\mathbf{R}(r) \bowtie \mathbf{R}(a))$ となる。

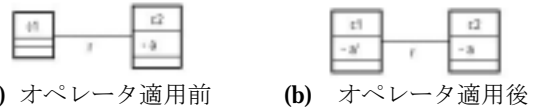


図9 CA 演算例
Fig. 9 CA operation

RA 演算. 与えられた $cxid$ に関して $a \in A$ とする。 $\text{RA}_a(cxid)$ は、 $cxid$ から a を除去する演算である。

3.4 その他

Key 演算. 与えられた $cxid$ に関して、 $c \in C$ はクラス、 $a \in A$ は c の属性であるとする。 $\text{Key}_{c,a}(cxid)$ は、 c のID変数を S_{kj} から S'_{kj} に変更する。ここで S'_{kj} は a の属性値 v に束縛される。この演算はクラス図の構造には影響を与えないが、 c および c に直接接続された全ての関連・属性のXSBEに影響を及ぼすため、やや複雑な操作となる。演算結果において、元の a に対応するものを a' 、 c に対応するものを c' 、 c に接続された属性 a_i に対応するものを a'_i 、 c に接続された関連 r_i に対応するものを r'_i とする。このとき、結果のXSBEが計算するリレーションはそれぞれ次のようになる。

- $\mathbf{R}(a') = \delta_{V(a) \rightarrow K_i}(\pi_{V(a)}(\mathbf{R}(a)))$
- $\mathbf{R}(c') = \delta_{V(a) \rightarrow K_i}(\mathbf{R}(a'))$
- $\mathbf{R}(a'_i) = \delta_{V(a) \rightarrow K_i}(\pi_{\neg \text{common}(\mathbf{R}(a), \mathbf{R}(a_i))}(\mathbf{R}(a) \bowtie \mathbf{R}(a_i)))$
- $\mathbf{R}(r'_i) = \delta_{V(a) \rightarrow K_i}(\pi_{\neg \text{common}(\mathbf{R}(a), \mathbf{R}(r_i))}(\mathbf{R}(a) \bowtie \mathbf{R}(r_i)))$

ここで、 $\delta_{V(a) \rightarrow K_i}$ は $V(a)$ 中の属性名をキー属性名に変換する。

RN 演算. 与えられた $cxid$ に関して、 $x \in N \cup R$ はクラス、属性、および関連のいずれかであるとする。また s は文字列であるとする。この時、 $\text{RN}_{x,s}(cxid)$ は x の名前を s に変更する。

A2R 演算. 属性から関連を作る演算である(図10)。与えられた $cxid$ に関して、 $c_1, c_2 \in C$ はクラス、 $a_1, a_2 \in A$ はそれぞれ c_1, c_2 の属性であるとする。(すなわち、 $(c_1, a_1), (c_2, a_2) \in R$ である。)この時、 $\text{A2R}_{(c_1, a_1), (c_2, a_2) \rightarrow r'}(cxid)$ は新たな関連 r' を $cxid$ に追加する。ただし、 $\mathbf{R}(r') = \pi_{K(a_1), K(a_2)}(\mathbf{R}(a_1) \bowtie_{V(a_1) = V(a_2)} \mathbf{R}(a_2))$ となる。

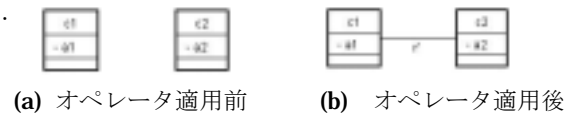


図10 A2R 演算例
Fig. 10 A2R operation

C2A 演算. クラス c をクラス c' の属性 a' に変換する複合演算である。 $c, c' \in C, a \in A, (c, a), (c, c') \in R, name((c, c')) \neq "isa", name(c) = s$ とする。このとき、 $\text{C2A}_{c \rightarrow (c', a)}(cxid) = \text{RC}_c(\text{CA}_{a \rightarrow (c', a)}(\text{RN}_{a,s}(cxid)))$ と定義する。

4. 意味情報の作成支援

本章では、本システムを用いた意味情報の作成の概要について説明する。意味情報の作成は次の3段階から構成される。(フェイズ1)システムがXMLのスキーマを基に単純な規則を用いてデフォルトのCX図を作成する。(フェイズ2)システムがヒューリスティクスと3章で説明したオペレータを用いてフェイズ1で作成したデフォルトのCX図を変更する。(フェイズ3)利用者がオペレータを用いてフェイズ2の結果を変更する。

4.1 フェイズ1

図2(b)にフェイズ1を適用した結果(図11)を用いて説明する。

³ 本論文では、関連 r_k がクラス c_i, c_j を結ぶ時、 $r_k = (c_i, c_j)$ と記述する。

クラスの生成. スキーマに現れる各要素毎に、一つのクラスを作成する. 例えば, 図11のクラス **person**は図2(b)の要素 **person**に対応する. クラス **person**のXSBEは $\$i_1$ in /univ/personとなる.

属性の生成. スキーマの要素属性毎にクラス属性を生成する. 例えば, 図 11 のクラス **person** に属性 **did** が生成される. この属性の XSBE は $\$i_1$ in /univ/person, $\$v_1$ in $\$i_1$ /@did となる.

関連の生成. 要素間に親子関係があるとき, それらの要素に対応したクラス間を二項関連で結び, 関連名を“has”とする. 例えば, 図2(b)の要素 **univ** と **person**間には親子関係であるため, 図11のクラス **univ**と **person**間に関連 **has**が生成される. 関連のXSBEは, $\$i_0$ in /univ, $\$i_1$ in $\$i_0$ /personとなる.

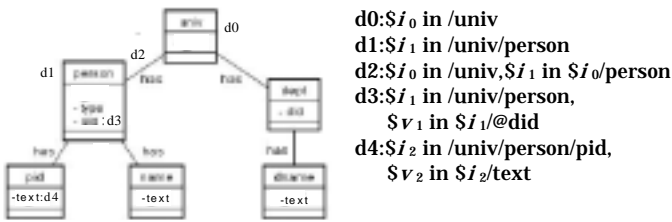


図11 フェイズ1の出力
Fig. 11 Output of phase1

4.2 フェイズ2

フェイズ2では, システムがヒューリスティクスを利用してフェイズ1の結果を変更する. 変更には3章で定義したオペレータを利用する. ヒューリスティクスの一部を次に示す.

- ・ 同じ名前を持つID属性とIDREF属性はA2Rオペレータを用いて関連に変更する.
- ・ テキストノードしか持たないクラスはC2Aオペレータを用いて属性に変換する.
- ・ ID属性を持つクラス c は, Keyオペレータを用いて c のID変数をキー変数と変更する.

フェイズ2では図11の cx_d1 に対して次の変換を行う.

$$cx_d2 = (Key_{c_2,a_6}(RA_{a_5}(A2R_{(c_1,a_5),(c_2,a_6)}r'(C2A_{c_5-(c_2,a_3)}(C2A_{c_4-(c_1,a_2)}(C2A_{c_3-(c_1,a_1)}(cx_d1)))))))$$

ただし, c_1 はperson クラス, c_2 はdept クラス, c_3 はpid クラス, c_4 はname クラス, c_5 はdname クラス, a_5 は c_1 のdid 属性, a_6 は c_2 のdid 属性である.

4.3 フェイズ3

フェイズ3では, フェイズ2の結果に対して利用者がオペレータを適用し, 最終的なCX図を得る. フェイズ2の結果から図3(b)を求めるためには, 次の演算を行う必要がある.

$$cx_d3 = (RN_{r'},"belongsTo"(RN_{c_7,"student"(RN_{c_6,"prof"(SubC_{c_1-(c_7,p_2)}(SubC_{c_1-(c_6,p_1)}(RN_{a_1,"id"(Key_{c_1,a_1}(RA_{a_4}(cx_d2))))))))))$$

ただし, p_1 は $\$k_1$./@type="prof", p_2 は $\$k_1$./@type="student", a_4 は c_1 のtype 属性である.

5. 関連研究

DTDやXML SchemaなどのXMLスキーマ言語から, 意味情報を生成する研究はいくつか存在する. Melloらの研究[3] [4]ではDTDをConceptual Schemaに変換するための規則を定義している. また, XMLのスキーマ言語の一つであるXGrammar[5]では, EERダイアグラムとの相互変換方法が与えられている. これらが本研究と異なる点は次の通りである.

- (1)意味情報の構成要素とXMLインスタンス間との関係の記述を明示的に保持しない.
- (2)意味情報のオペレータが定義されていない.
- (3)作成される意味情報が, 与えられたスキーマとほぼ1対1対応する.

C-Web [1]では, XSBEと同様, 意味情報の構成要素とXMLインスタンス間とのマッピングを表すための言語を提供している. しかし言語の表現力がXSBEより小さく, XSBEでのみ記述可能なマッピングが存在する. また, C-Webでは意味情報の自動生成については議論していない.

6. おわりに

本稿では, XMLデータの意味情報の表記法および意味情報の作成支援システムについて説明した. 今後の課題としては, ヒューリスティクスの適用手順の明確化, システムの実装などがあ

【謝辞】

ゼミなどでご議論いただきました芝浦工業大学工学部情報工学科古宮誠一教授に感謝いたします. 本研究の一部は文部科学省科学研究費補助金若手研究(B)(課題番号15700108)による.

【文献】

- [1] B. Amann, I. Fundulaki, M. Scholl, C. Beeri, Anne-Marie Vercoustre: "Mapping XML Fragments to Community Web Ontologies.", WebDB 2001: 97-102
- [2] S. Bergamaschi, S. Castano, M. Vincini: "Semantic Integration of Semistructured and Structured Data Sources.", SIGMOD Record 28(1): 54-59 (1999)
- [3] R. S. Mello, C. A. Heuser: "A Rule-Based Conversion of a DTD to a Conceptual Schema.", ER 2001: 133-148
- [4] R. S. Mello, C. A. Heuser: "A Bottom-Up Approach for Integration of XML Sources.", Workshop on Information Integration on the Web 2001: 118-124
- [5] M. Mani, D. Lee, R. R. Muntz: "Semantic Data Modeling Using XML Schemas.", ER 2001: 149-163
- [6] E. Pitoura, O. A. Bukhres, A. K. Elmagarmid: "Object Orientation in Multidatabase Systems.", ACM Computing Surveys 27(2): 141-195 (1995)
- [7] P. F. Patel-Schneider, J. Simeon: "The Yin/Yang Web: XML Syntax and RDF Semantics.", WWW 2002: 443-453
- [8] 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): 9-10 (2003).

古川 夏子 Natsuko FURUKAWA

芝浦工業大学大学院工学研究科修士課程在学中. XML処理・管理, XMLの情報統合などに興味を持つ. 情報処理学会, 日本データベース学会各学生会員.

森嶋 厚行 Atsuyuki MORISHIMA

筑波大学知的コミュニティ基盤研究センター助教授. 1998年筑波大学大学院工学研究科修士. 博士(工学). 情報統合, XMLとデータベース, Webの一貫性管理, Webアーカイブなどに興味を持つ. ACM, IEEE-CS, 情報処理学会, 電子情報通信学会, 日本データベース学会各正会員.