

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

Implementation of a Prototype System to Extract Semantic Information from XML Data

古川 夏子[♡] 上村 匡稔[◇] 大河原 俊明^{*}
森嶋 厚行^{*} 杉本 重雄[□]

Natsuko FURUKAWA Tadatoshi KAMIMURA
Toshiaki OHKAWARA Atsuyuki MORISHIMA
Shigeo SUGIMOTO

XML はインターネットにおけるデータ交換のための標準フォーマットとしての地位を確立している。XML データの量が劇的に増加するにしたがい、XML データに対するデータ統合が重要な問題となっている。データ統合における重要な事項として、データ変換プログラムの開発がある。データ変換プログラムの開発には、大きなコストがかかる事が知られている。我々は、XML のデータ変換プログラムの開発を支援するために、XML データから意味情報(クラス図)を抽出するためのシステムを開発を行っている。本稿は、意味情報抽出支援手法の概要およびプロトタイプシステムの設計と実装について説明する。

XML has become a standard format for data interchange on the Internet. As the amount of XML data grows, integration of XML data becomes one of the crucial problems. While the development of data transformation programs is a key issue to achieve data integration, the development requires tremendous efforts in general. We are developing a system for extracting semantic information (class diagrams) from XML data, which is aimed at helping the program development process. The paper explains the overview of our method and the design and implementation of our prototype system.

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。現実に作成・管理される XML データの量が劇的に増加するにしたがい、XML データに対するデータ統合が重要な問題となっている。データ統合の実現は、スキーマの分析や変換プログラムの実装など、大きな開発コストがかかる。我々は、データ統合の重要な要素技術であるデータ変換の問題に着目する。最もわかりやすいデータ変換の例としては、ローカルなデータベースのデータを、統合スキーマに合わせるための変換がある。我々は、意味情報を利用した、データ変換プログラムの開発支援の研究を行っている [1][2][3]。我々

[♡] 学生会員 芝浦工業大学大学院 工学研究科 現在、工学院大学工学部 k104038@ccs.kogakuin.ac.jp

[◇] 正会員 筑波大学大学院 図書館情報メディア研究科

^{*} 学生会員 筑波大学大学院 図書館情報メディア研究科 okwr@slis.tsukuba.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 mori@slis.tsukuba.ac.jp

[□] 非会員 筑波大学 知的コミュニティ基盤研究センター sugimoto@slis.tsukuba.ac.jp

```

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

```

(a) (b)
図 1 2 種類の XML スキーマ $Sch_A(a)$ と $Sch_B(b)$
Fig. 1 Two XML Schemas $Sch_A(a)$ and $Sch_B(b)$

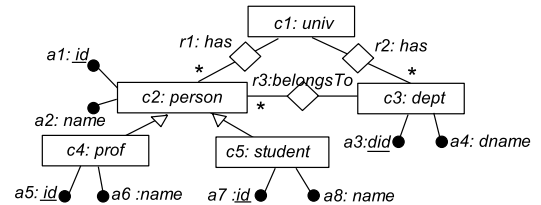


図 2 クラス図 cd_0
Fig. 2 Class Diagram cd_0

のアプローチでは、変換対象となる XML データから意味情報を抽出し、それをデータ変換に利用する。本稿における意味情報とは、直観的には XML データが表す概念モデルに関する情報である。例えば、図 1(b) の XML スキーマ Sch_B が表す意味情報は、構造は異なるものの本質的には図 2 のクラス図として表現できる。本稿では、XML データから意味情報を抽出するためのシステムの設計・実装について説明する。

データの意味情報は、本プロジェクトだけでなく様々なデータ統合・変換プロジェクトで利用されており [4][5]、一般的にも XML からの意味情報抽出は重要な問題である。

本システムの新規性は主に二つある。第一に、クラス図を抽出するだけでなく、XML インスタンスとクラス図を結びつけるための問合せも同時に生成する事である。これにより、概念モデルの各構成要素がどのように XML として表現されているかがわかるため、データ変換への利用が容易である。我々の知る限り、データ (RDB, XML など) からクラス図などを抽出するための他のシステム/手法 [6][7] では、データとクラス図との関連は生成しない。第二に、形式的に定義されたモデル [3] に基づき開発されている事である。これにより、次の利点がある。(1) 意味情報の抽出作業が、形式的に定義されたプリミティブな操作の組合せで行われるため、一貫性のあるアドホックでない枠組みを提供できる。(2) システムのモジュール化、機能拡張が容易である。

2. 写像としてのデータ変換と意味情報の抽出

我々の提案するデータ変換モデルでは、データ変換を写像としてモデル化する (図 3 実線矢印)。このとき、データ変換の問題は次の様に記述する事ができる。

入力: 変換前のスキーマ $scha$, 変換前のインスタンス $I(scha)$, 変換後のスキーマ $schb$ 。

出力: $I(scha)$ にデータ変換写像 F を適用した結果 $F(I(scha))$ 。

したがって、写像 F の発見がデータ変換の問題の鍵である。

データ変換問題の具体例を次に示す。

例 1. 図 1(a)(b) は、2 つの大学における XML データのスキーマ Sch_A と Sch_B である。これらは構造が異なるものの、実はどちらも図 2 のクラス図 cd_0 で表現される同種の情報を表す (Sch_A の XML では、person の type 属性が "s" の時 student, "p" の時 prof を表す)。この時、 $I(Sch_A)$ から $I(Sch_B)$ への写像 $F_{AB}: Sch_A \rightarrow Sch_B$ を求めたい。

2.1 意味写像の導入

我々は、データ変換の問題に対して、意味領域を介した変換というアプローチを取る。具体的には、XML から意味領域への写像 (以下、意味写像) を利用する (図 3 左の点線)。ここで意味領域と

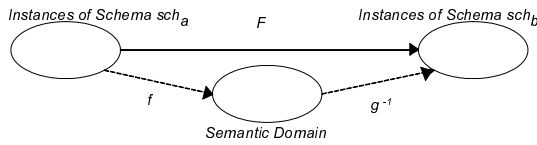


図3 データ変換の写像によるモデル化
Fig. 3 Modeling a Data Transformation as a Mapping

は、直観的にはデータベース設計における概念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、概念レベルでどのようなデータを保持しているかを表す領域である。意味写像は $f: sch \rightarrow cd$ と表す。この写像の定義域は、XML スキーマ sch を満たす XML インスタンスの集合 $dom(sch)$ である。また、値域はクラス図 cd (例えば図2) が規定する値の集合 $dom(cd)$ である。本稿ではクラス図の定義および値の定義は省略し、直観的な説明だけを行う。直観的には、 $v \in dom(cd)$ である値 v は、 cd 中の各ノード(クラス、関連¹、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、 $v \in dom(cd_0)$ (cd_0 は図2参照) は、次の形式をしている。

$[c_1: univ$ 要素, $c_2: person$ 要素の集合, $c_3: dept$ 要素の集合, $\dots, r_1: \dots, r_3: person$ と $dept$ の関係表, $\dots, a_8: \dots]$

意味領域を介した変換には次の利点がある。(1) 意味領域の知識(オントロジなど)を変換写像の構築に利用しやすい。(2) スキーマ間の対応関係だけでは自明でないもの(例1の prof 等)を扱いやすい。

2.2 意味写像

意味写像 f は、 $w \in dom(sch)$ なる XML インスタンス w から値 $v \in dom(cd)$ 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、複数の XQuery 風の式から構成される。例えば、意味写像 $f_A: Sch_A \rightarrow cd_0$ (Sch_A は図1(a), cd_0 は図2参照)において、 cd_0 の dept クラス (c_3) のインスタンス集合を計算する式は、for $\$k$ in /univ/depts/dept return $\$k$ as dept である。

2.3 意味写像を利用したデータ変換写像の構築

意味写像を導入すると、データ変換写像 $F: sch_a \rightarrow sch_b$ は、図3の点線矢印の様にモデル化できる。ここで f および g はそれぞれ意味写像である。

本システムにおける意味情報の抽出とは、意味写像 $f: sch_a \rightarrow cd_a$ (図3左の点線)の構築を行うことである:

[意味情報の抽出] sch_a を基にクラス図 cd_a を導出し、意味写像 $f: sch_a \rightarrow cd_a$ を構築する。例えば、例1においては、 Sch_A を基に cd_0 を導出し、 $f_A: Sch_A \rightarrow cd_0$ を構築する。一般論として、XML スキーマを設計する際には cd_0 の様な概念モデルを想定しながら XML での表現を決定、という過程をとると考えられる。[意味情報の抽出] は、この過程に対する一種のリバースエンジニアリングを行うものである。

意味情報の抽出のために、本システムでは我々が提案するデータ変換モデルで定義している写像の操作系を利用する。次節ではこの概要を説明する。

2.4 写像操作系

本データ変換モデルの写像操作系は、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, 意味領域射影, 意味領域拡張, 値域制限, 名前変更)を持つ。本稿では、特に[意味情報の抽出]で重要な操作子である、意味領域拡張について説明する。他の演算子の説明は[3]にある。

意味領域拡張。意味領域拡張は、意味写像 $f: sch \rightarrow cd$ が与えられたとき、 cd に新たなノード n_i (クラス, 関連, もしくは属性)を追加した cd' への写像 $f': sch \rightarrow cd'$ を作成する操作である(図4(a))

¹汎化は関連ではない。

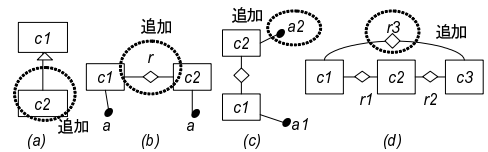


図4 意味領域拡張操作子による cd から cd' への変更
Fig. 4 Changes by the Semantic-Domain Extension Operator

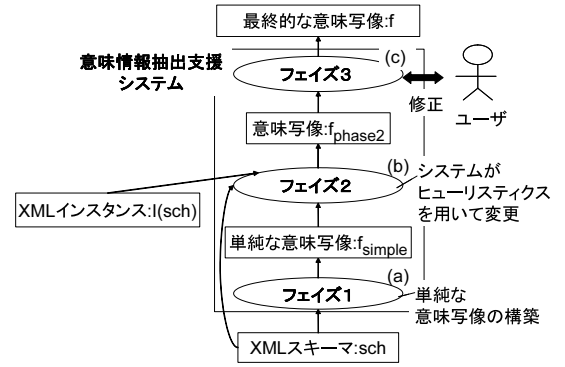


図5 アーキテクチャ
Fig. 5 Architecture

~(d)). 具体的には次のように指定する。意味写像 $f: sch \rightarrow cd$ 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、意味領域拡張の結果 f' は操作子 γ を用いて $\gamma_e^{n_i}[f]: sch \rightarrow cd'$ で求められる。本稿では、式の可読性を高めるため、 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。例えば、本稿で用いられる演算子である、 $C_{E_p, c_1}^2[f]$ 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4(a))。 $R_a^{r(c_1, c_2)}[f]$ は属性値 a の値が等しい事を表す関連 r を作成する(図4(b))。 $A_{c_1, a_1}^{c_2, a_2}[f]$ はクラス c_1 の属性 a_1 を c_2 の属性 a_2 としてコピーする(図4(c))。 $R_{r_1(c_1, c_2), r_2(c_2, c_3)}^{r_3(c_1, c_2, c_3)}[f]$ は関連の合成を作成する(図4(d))。

3. 意味情報抽出支援システム

上述したように、本システムにおける[意味情報の抽出]とは、XML スキーマ sch からクラス図 cd を導出し、意味写像 $f: sch \rightarrow cd$ を構築することである。

本システムのアーキテクチャを図5に示す。基本的な考え方は、意味情報の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。意味情報の抽出は次の3段階で行われる。

(フェーズ1) 入力として XML スキーマ sch を受け取り、単純な意味写像 f_{simple} を出力する(図5(a))。ここで、単純な意味写像 $f_{simple}: sch \rightarrow cd$ の cd は sch と同型の構造を持つ。すなわち、 sch 中の各 XML 要素に対しては cd 中の一つのクラスが対応し、各要素属性に対しては cd 中の一つのクラスの属性が対応する。また、 sch 中の要素の親子関係が cd の関連に直接対応する。例えば、図6(a)は、単純な意味写像 $f_{A, simple}: Sch_A \rightarrow cd_1$ (Sch_A は図1(a)に示す)における cd_1 である。また、図6(b)は意味写像 $f_{A, simple}$ を構成する式の一部である。

(フェーズ2) 入力としてフェーズ1の結果の f_{simple} と、XML インスタンス $I(sch)$ および XML スキーマ sch を受け取り、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_{phase2} を構築し、出力する(図5(b))。ヒューリスティクスには様々なものが考えられるが、以下では、例1の意味情報抽出に必要なものを説明する。

ヒューリスティクス1. XML において、関連はしばしば ID 属性と IDREF 属性を用いて実装される。したがって、逆に ID 属性と IDREF 属性から関連を作成すれば良い。この目的の

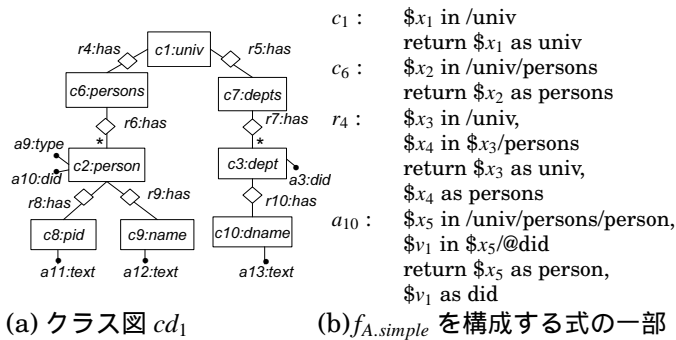


図 6 フェーズ 1 の出力例
Fig. 6 Example of Phase1's Output

ために、同じ属性名 a の ID 属性と IDREF 属性を持つクラス c_1, c_2 の間に関連を追加する。これは、操作子 $R_a^{(c_1, c_2)}[f]$ を用いて行う (図 4(b))。

ヒューリスティクス 2. クラスの属性に対応する情報は、XML においては、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、ある XML 要素がテキストだけを持ち、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他のクラス属性としてコピーする。これは、操作子 $A_{c_1, a_1}^{c_2, a_2}[f]$ を用いて行う (図 4(c))。

ヒューリスティクス 3. XML のスキーマを設計するとき、しばしば、それ自体は特に意味を持たないが、単に他の要素をまとめたり形式を整えるための目的で XML の要素を導入する事がある。そのような要素に対応するクラスの特徴として、他のクラスとの 1 対 1 関連を持つこと、もう一方の関連と 1 対多関連で接続されていること、属性を持たない事がある。したがって、そのようなクラスを次のように除去する。クラス c_1, c_2 の間に関連名 has を持つ関連 r_1 があり、 c_2, c_3 の間には関連 r_2 があるとすると、この時、 r_1 が r_2 の一方の関連が 1 対 1 関連、もう一方の関連が 1 対多関連であり、かつ c_2 が属性を持たないとき、操作子 $R_{r_1(c_1, c_2), r_2(c_2, c_3)}^{r_3(c_1, c_3)}[f]$ を用いて c_1 と c_3 を直接結ぶ関連 r_3 を追加する (図 4(d))。

例 1 のフェーズ 1 で得られた $f_{A.simple}$ に対し、以上のヒューリスティクスを用いて操作子を適用すると、次のような意味写像 $f_{A.phase2} : Sch_A \rightarrow cd_2$ が得られる。

$$f_{A.phase2} \equiv \begin{matrix} R_{has(univ, person)}^{has(univ, persons), has(persons, person)} [\\ R_{has(univ, dept)}^{has(univ, depts), has(depts, dept)} [R_{belongsTo(person, dept)} [\\ A_{person.name}^{person.pid} [A_{person.pid} [A_{dept.dname}^{dept.dname} [f_{A.simple}]]]]] \end{matrix}$$

(フェーズ 3) フェーズ 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。フェーズ 2 の例では、例えば prof クラス等は発見できていない。フェーズ 3 では、ユーザが写像操作子を $f_{A.phase2}$ に適用し (図 5(c))、最終的な意味写像 $f : sch \rightarrow cd$ を得る。以下の式では、例 1 に対するフェーズ 3 の結果 $f_A : Sch_A \rightarrow cd_0$ を得るための式を示す。 cd_0 は図 2 に示されている。ここでのポイントの一つは、フェーズ 2 とフェーズ 3 で共に、本モデルで形式的に定義された操作子を用いているため、システムによる操作と人手による操作が同じ言葉で記述出来る事である。

$$f_A \equiv M_{person.belongsTo: *} [C_{student}^{prof} [C_{@type='s'}^{person} [C_{@type='p'}^{person} [\delta_{pid}^{sid} [f_{A.phase2}]]]]]$$

ここで、本式に現れる操作子のうち、意味領域拡張以外の操作子の説明は次の通りである。まず、名前変更操作子 δ_{pid}^{sid} は、person クラスの属性の名前 “pid” を “id” に変更する。次に、値域制限操作子 $M_{person.belongsTo: *}$ は、関連 belongsTo が 1 対多関連であることを指定している。

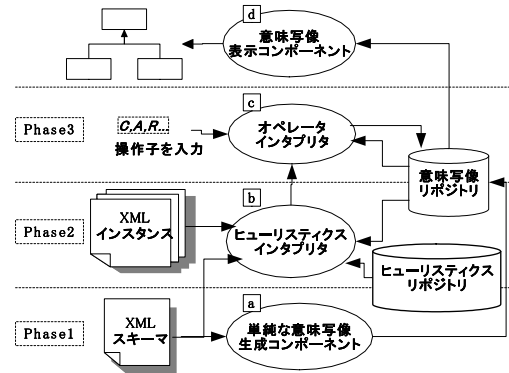


図 7 意味情報抽出支援システムの設計
Fig. 7 Design of the System for Extracting Semantic Information

4. 意味情報抽出支援システムの設計と実装

4.1 システム設計のポイント

本システムの設計にあたっては、次に挙げる 3 つの点に留意した。システムの拡張性: 機能ごとにモジュールを分割し、モジュール間でやりとりするデータの構造の標準化を図ることにより、システムの拡張性を高くする。これにより、モジュールの交換・新モジュールの追加などが容易になる。例えば、4.7 節で説明するプロトタイプシステムではクラス図の表示に Graphviz[9] を用いているが、クラス図の表示に別のツールを用いる場合には、表示部分がモジュール化されているため、モジュールの交換が行いやすい。他ツールとの連携: 標準に準拠することにより、他ツールとの連携を容易にする。例えば、クラス図は、UML における標準である XMI 形式 [8] で出力できるようにする。これによって、本システムで出力した XMI 形式のファイルを各種 UML ツールで利用することや、逆に UML ツールで出力した XMI 形式のファイルを本システムで利用することが可能になる。

ユーザによる作業の効率化の支援: ユーザによる作業の効率化を支援するため、操作子の省略記法を用意することによる入力作業の軽減、および UML による簡潔なクラス図の表示などの取り組みを行う。通常、ユーザがクラス図を操作するためにシステムに入力する式は複雑な記述となるため、ユーザによる入力作業が負担となる。これに対し、ユーザが入力するための省略記法をあらかじめ組み込んだことで、入力作業の軽減を実現する。また、本システムが利用している変換モデルでは、クラス図の表記に ER 図をベースにした形式を用いている。しかし、ER 図は関連や属性をすべてノードで表現するため、複雑になる傾向がある。そこで、クラス図の表示は UML によって行うことにより、簡潔なクラス図の表示を実現する。

4.2 概要

本システムの設計を図 7 に示す。本システムは 4 つのコンポーネントから構成される。(a) 単純な意味写像生成コンポーネント: 3 章で説明したフェーズ 1 を行う。すなわち、XML スキーマから単純な意味写像を生成する。(b) ヒューリスティクスインタプリタ: フェーズ 2 を行う。具体的には、ヒューリスティクスリポジトリ中のヒューリスティクスを用いて、与えられた意味写像に適用する操作子群を決定する。ヒューリスティクスには 3 章で挙げたもの以外にもドメイン固有のヒューリスティクスなど、様々なものが考えられる。したがって、これらヒューリスティクスはハードコーディングせず、リポジトリに保存しそれをインタプリタが実行する設計とする。(c) オペレータインタプリタ: 意味写像と意味写像の操作子を受け取り、適用結果を出力する。(d) 意味写像表示コンポーネント: 意味写像のクラス図等をユーザに提示する。クラス図の表現形式としては、ER 図、UML のクラス図、OWL 等多様な表現で出力可能とする。以下では、各コンポーネントの説明を行う。

4.3 単純な意味写像生成コンポーネント

単純な意味写像生成コンポーネント(図 7(a))は、まず、入力として XML スキーマを受け取り、XML スキーマの要素、要素属性、要素間の親子関係などを解析し、クラス図のクラス、クラス属性、関連に対応させた構文木に変換する。次に、その構文木をクラス図のクラスや属性、関連と対応させて単純な意味写像を生成し、意味写像リポジトリに出力する。

4.4 ヒューリスティクスインタプリタ

ヒューリスティクスインタプリタ(図 7(b))は、XML インスタンスと XML スキーマ、単純な意味写像を受け取り、適用すべきヒューリスティクスを発見する。そして、このヒューリスティクスに基づき具体的に適用する操作子を決定する。これらの作業は、図 8 のヒューリスティクス決定モジュールおよびオペレータ決定モジュールによって行われる。

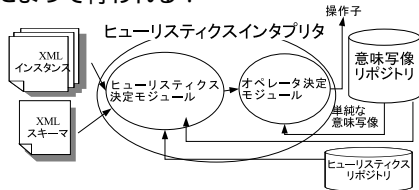


図 8 ヒューリスティクスインタプリタの各モジュール
Fig. 8 Modules of the Heuristics Interpreter

4.5 オペレータインタプリタ

オペレータインタプリタ(図 7(c))は、まず、入力として意味写像とそれに適用するための操作子を受け取る。次に、その操作子を適用して意味写像を変換し、それを意味写像リポジトリに出力する。ここで、ヒューリスティクスインタプリタや直接ユーザから入力として受け取る操作子は、省略記法や複合操作子などで記述された様々な形式が考えられる。その場合には、それをプリミティブな操作子群に展開する処理が行われる。

4.6 意味写像表示コンポーネント

意味写像表示コンポーネント(図 7(d))は、各フェイズ毎に構築された意味写像のクラス図をユーザに表示する。まず、入力として、意味写像リポジトリから受け取った意味写像を、クラス図を表示するための出力形式に変換する。次に、出力形式に変換された意味写像を、クラス図として表示し、出力する。

4.7 プロトタイプシステム

前節までの設計に基づき、意味情報抽出支援システムのプロトタイプを作成した。記述言語は Java であり、約 3800 行のコードで構成される。Java のコードにおけるクラス数は 19 である。入力する XML スキーマは RELAX NG で記述する。現時点では UML のクラス図(XMI 形式)を出力可能である。フェイズ 2 で利用するヒューリスティクスはハードコーディングされている。フェイズ 3 のユーザインターフェースとしては、現在、操作子を直接記述するコマンドラインインターフェースだけをサポートしている。

図 9 は、本プロトタイプシステムの実行画面を示している。上の部分がコマンドラインインターフェース、左下の部分は意味写像を構成する式の表示であり、右下はクラス図の表示である。

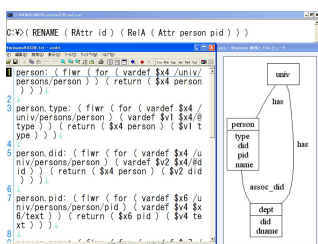


図 9 プロトタイプシステムの実行例
Fig. 9 Screen of the Prototype System

5. おわりに

本稿では、データ変換の応用を目的とした XML データからの意味情報抽出支援システムのアーキテクチャ、意味情報の抽出手法、およびプロトタイプシステムの実装についての説明を行った。今後の課題としては、フェイズ 2 で適用するヒューリスティクスの拡張機構の設計や、ドメインオントロジを利用した意味情報の抽出手法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。また、筑波大学図書館情報学系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。

【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), pp. 9-10, 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, pp. 5-8, 2003 年 10 月.
- [3] 森嶋厚行. 情報統合応用のためのデータ変換モデルの提案. 日本データベース学会 Letters, Vol. 3, No. 1, 2004 年.
- [4] S. Bergamaschi, S. Castano, M. Vincini: Semantic Integration of Semistructured and Structured Data Sources. SIGMOD Record 28(1): 54-59 (1999)
- [5] 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
- [6] S. Ramanathan and J. Hodges. Extraction of object-oriented structures from existing relational databases. ACM SIGMOD Record, 26(1), March 1997.
- [7] M. Mani, D. Lee, R. R. Muntz: Semantic Data Modeling Using XML Schemas. ER 2001: 149-163
- [8] OMG. XMI.
<http://www.omg.org/technology/xml/index.htm>.
- [9] AT&T. Graphviz.
<http://www.research.att.com/sw/tools/graphviz>.

古川 夏子 Natsuko FURUKAWA

現在, 工学院大学工学部在学中. 2004 年 芝浦工業大学大学院工学研究科修士課程修了. 情報処理学会, 日本データベース学会各学生会員.

上村 匡稔 Tadatoshi KAMIMURA

新日鉄ソリューションズ株式会社勤務. 2002 年 筑波大学大学院理工学研究科修士課程修了. 2003~2004 年 筑波大学大学院 図書館情報メディア研究科研究生. 日本データベース学会正会員.

大河原 俊明 Toshiaki OHKAWARA

筑波大学大学院 図書館情報メディア研究科博士前期課程在学中. 2004 年 図書館情報大学 図書館情報学部卒業. 日本データベース学会学生会員.

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了. 博士(工学). 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.

杉本 重雄 Shigeo SUGIMOTO

筑波大学知的コミュニティ基盤研究センター教授. 京都大学工学部情報工学科, 同大学院工学研究科情報工学専攻博士後期課程修了. 工学博士. ACM, IEEE-CS, 情報処理学会他会員.