

## 一貫性制約を考慮したデータ管理のための XML-RDB マッピング手法

太田 壮祐<sup>†</sup> 森嶋 厚行<sup>‡</sup> 天笠 俊之<sup>††</sup> 只石 正輝<sup>‡</sup>筑波大学 図書館情報専門学群<sup>†</sup> 筑波大学大学院 図書館情報メディア研究科<sup>‡</sup> 筑波大学大学院 システム情報工学研究科<sup>††</sup>

## 1. はじめに

これまで、XML データを RDB にマッピングする手法が数多く研究されてきた。これらのマッピング手法の多くはモデル写像<sup>1)</sup> や構造写像<sup>2)</sup> 等の技法を用いて XML の木構造に従ったマッピングを行うものであり、データ一貫性制約を考慮していない。例えば、図 1 の auction.dtd という DTD に従った XML データ auction.xml を構造写像でマッピングすると、auction(ID, No, article, price, seller.name, seller.rating) というリレーションスキーマが生成される。しかし、このリレーションスキーマでは同じ出品者 (seller.name) が出品する商品が複数存在すると、seller.name と seller.rating の値が複数のタプルに現れる。これは関数従属性を考慮して正規化したりレジョンではないからである。その結果、このリレーションでは更新不整合が起こる可能性がある。

関連研究として、XML データに存在する関数従属性を考慮した XML-RDB マッピング手法には RRXS<sup>3)</sup> がある。本論文では、XML データに存在する関数従属性だけでなく包含従属性<sup>4)</sup> も考慮したマッピング手法を提案する。

## 2. 提案手法

## 2.1 システム概要

図 2 は、提案システムの概要である。本システムは、XML データの集合  $XML$ 、XML データに存在する関数従属性の集合  $XFD$ (詳細は 2.2 節で説明する)、XML データに存在する包含従属性の集合  $XIND$ (詳細は 2.3 節で説明する) を入力とし、与えられた制約を満たしたりレジョンの集合  $R$  を出力する。本システムでは XML データの全てを RDB にマッピングする事は行わず、 $XFD$  で指定された XML 要素やその属性のみを RDB にマッピングする。マッピングされなかった部分は XML テンプレート集合  $XML'$  として出力する。

本システムは次の手順で処理を行う。(1)  $XFD$  と  $XIND$  を用いてスキーマのマッピングを行いリレーションスキーマの集合  $RS$  を生成する。(2)  $XML$  を、スキーマ  $RS$  に従うインスタンス  $R$  と  $XML'$  に変換する。本稿では主に (1) のマッピング手法について説明する。

## 2.2 XML Functional Dependencies

XML Functional Dependencies (以下 XFD) は論文<sup>3)</sup> で定義された XML のための関数従属性である。auction.xml において、No の値が決まれば article の値が決まる、とい

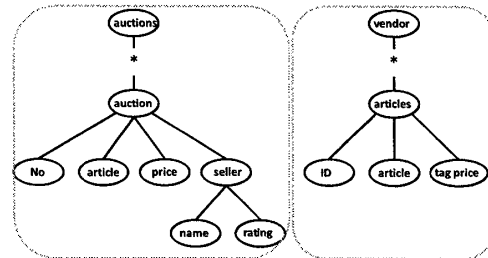


図 1 auction.dtd(左) と vendor.dtd(右)

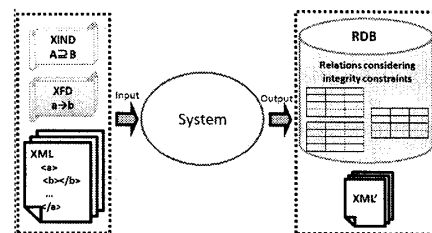


図 2 マッピングシステムのアーキテクチャ

う XFD を定義したい場合は次のように記述する。

$$\text{\$x in auction.xml/auctions/auction} \quad \dots(1)$$

$$\text{\$x/No/value()} \rightarrow \text{\$x/article/value()} \quad \dots(2)$$

(1) で XPath 式により抽出した auction ノードを変数  $\text{\$x}$  に束縛し、さらに  $\text{\$x}$  を用いて、(2) で各ノードに対する関数従属性を記述している。

## 2.3 XML Inclusion Dependencies

本稿では、XML データを対象とした包含従属性の表記として XML Inclusion Dependencies (以下 XIND) を導入する。XIND は XML パス式を用いて表記する。例えば、auction.xml の article の値は全て、図 1 の vendor.dtd に従った XML データ vendor.xml の article の値に含まれていなければならない、という XIND は次のように記述する。

$$\text{vendor.xml/vendor/articles/article/value()}$$

$$\supseteq \text{auction.xml/auctions/auction/article/value()}$$

## 2.4 マッピング手法

本稿で提案するマッピングは次の手順で行う。

ステップ 1 入力として与えられた XFD を利用し、データベーススキーマ (リレーションスキーマの集合)  $RS$  を生成する。

ステップ 2 入力として与えられた XIND を利用し、 $RS$  を XIND を考慮したデータベーススキーマ  $RS'$  に変形する。

(1) については RRXS の手法に従い、関数従属性の左辺をリレーションの主キー属性、右辺をそのリレーションの属性として各リレーションスキーマを生成する。次節では、ステップ 2 の詳細を説明する。

ステップ 2 の詳細。ステップ 2 では、SQL データベースで一般にサポートされている外部キー制約を利用して XML データの包含従属性を維持できるよう、データベーススキーマ  $RS$  を  $RS'$  に変形する。外部キー制約は包含従属性の一種である。

An XML-RDB Mapping Method Considering Integrity Constraints  
 Sosuke Ota<sup>†</sup> Atsuyuki Morishima<sup>‡</sup> Toshiyuki Amagasa<sup>††</sup>  
 Masateru Tadaishi<sup>‡</sup>  
 Sch. of Library and Information Science, Univ. of Tsukuba.<sup>†</sup>  
 Grad. Sch. of Library, Information and Media Studies, Univ. of Tsukuba.<sup>‡</sup>  
 Grad. Sch. of Systems and Information Engineering, Univ. of Tsukuba.<sup>††</sup>

すなわち、2つのリレーションスキーマ  $R(K_R, \dots, A, \dots)$ ,  $S(K_S, \dots, B, \dots)$  がある時、外部キー制約  $R[K_R] \supseteq S[B]$  は、属性  $B$  の全ての値が、属性  $K_R$  の値に必ず存在していなければならないという制約を表す。一般的な RDBMS は指定された外部キー制約をリレーションが満たすようチェックする機能を持っている。

しかし、SQL データベースでサポートされる外部キー制約には、参照先の属性が主キーもしくは一意キーでなければならないという制限が存在する。したがって、 $R[A] \supseteq S[B]$  のように  $A$  が主キーでは無い一般の包含従属性は、SQL データベースではサポートしていない。XIND  $e_1 \supseteq e_2$  の両辺は XPath で表現された XML データの要素集合あるいは属性集合であるが、マッピング後のリレーションにおける包含従属性  $U[A_{e_1}] \supseteq V[A_{e_2}]$  における属性  $A_{e_1}$  が必ずしも一意キーとは限らない。したがって、単純な方法では、与えられた XIND を SQL データベースにおける外部キー制約で実装することはできない。

したがって、本手法では、与えられた XIND に対応するリレーションスキーマ上での包含従属性が  $R[A] \supseteq S[B]$  である時、その両辺をリレーション  $R$  と  $S$  の主キーでそれぞれ置き換えることによって出来る外部キー制約  $R[K_R] \supseteq S[K_S]$  に置き換える。

これが可能な条件は、リレーション  $T = R \bowtie_{K_R=K_S} S$  において  $T[A] = T[B]$  が成立する事である。この時、 $K_R$  と  $K_S$  がそれぞれ主キーであることから  $R$  において  $K_R \rightarrow A$  であり、 $S$  において  $K_S \rightarrow B$  である。したがって、 $R[K_R] \supseteq S[K_S]$  ならば  $R[A] \supseteq S[B]$  を満たす。

したがって、ステップ 2 では、(1) この条件を満たすため  $S$  の属性に  $R$  の主キーである  $K_R$  を追加する。さらに、(2) 冗長性を除去するため、リレーション  $S$  から属性  $B$  を除去する。この属性が必要な場合は  $T$  を計算して  $A$  を参照する (図 3)。図 3 では 6, 7 行目で XIND の両辺に対応するリレーション属性を求め、13 行目で (1)、14 行目で (2) の処理を行っている。

### 2.5 リレーションインスタンスへの変換の問題

ステップ 2 の手法で生成されたスキーマに従うインスタンスを生成する際に、 $S$  の各タプルに対して、追加された属性  $K_R$  の値を求める必要がある。これは次のように行う。すなわち、 $S$  の各タプルに対して、 $S$  のタプルと  $R$  のタプルを比較し、もし  $S[B]$  の値と  $R[A]$  が同じであれば、その  $R$  のタプルの  $R[K_R]$  を  $S[K_R]$  に挿入する。候補となる主キー値が一意に決まらない場合の対応は複数考えられるが、一番簡単な対応は、ユーザに問合せを行い解決するというものである。

次の例を用いて説明する。まず、図 1 の DTD に従った XML データ auction.xml, vendor.xml に対し、ステップ 1 で次のようなりレーションスキーマ auction(No, article, price, sellerID), articles(ID, article, tag, price) が生成されたとする (利用された XFD は省略する)。これに対し 2.3 節の XIND が与えられた時、図 4 に示すリレーションが生成される。

```

1. Procedure Step2 {
2.   input: RS, XIND
3.   output: RS'
4.   RS'=RS
5.   let XIND={dep_1  $\supseteq$  ref_1, dep_2  $\supseteq$  ref_2, ..., dep_n  $\supseteq$  ref_n}
6.   for each "dep_i  $\supseteq$  ref_i"  $\in$  XIND {
7.     let R[A] = CorrespondingAttr(dep_i) in RS'
8.     let S[B] = CorrespondingAttr(ref_i) in RS'
9.     S.addAttribute(R.primarykey)
10.    S.removeAttribute(B)
11.  }
12. return RS';
13. }
    
```

図 3 ステップ 2 のアルゴリズム

auction				articles		
No	price	sellerID	articlesID	ID	article	tag price
10001	3000	s11	1	1	cleaner	5000
10850	2200	s11	2	2	fan	2000
				...		

図 4 アルゴリズム適用後のリレーション

## 2.6 制限

ステップ 2 において、一般の包含従属性を外部キー制約に変換する手法を提案したが、これが適用可能なケースは、包含従属性の両辺に対応する属性が、URA<sup>4)</sup> における Universal Instance において同一の属性と見なされ、別々に格納したときに冗長と見なされる場合のみである。例えば、あらゆる文字列を格納する属性 *string* と、学生の名前リストを格納する属性 *name* 間の包含従属性  $string \supseteq name$  は入力される XIND としてあり得るが、これらの情報は冗長でなく、本手法では対応できない。しかし、実用上はこのような包含従属性が与えられることは少ないと考えられる。

## 3. まとめと今後の課題

本稿では、XML データに存在する包含従属性を考慮した RDB へのマッピング手法を提案した。今後の課題としては、理論的な解析や XFD や XIND の発見支援と組み合わせた仕組みの開発があげられる。

## 謝辞

本研究の一部は科学研究費補助金若手研究 (B)(#20700076) による。

## 参考文献

- 1) M. Yoshikawa, T. Amagasa, T. Shimura and S. Uemura: "XRel: A path-based approach to storage and retrieval of XML documents using relational databases", ACM Transactions on Internet Technology (TOIT), 1(1): 110-141, 2001.
- 2) Jayavel Shanmugasundaram, Kristin Tufte, Gang He, Chun Zhang, David De Witt, Jeffrey Naughton: Relational Databases for Querying XML Documents: Limitations and Opportunities. the 25th VLDB Conference: 302-314, 1999.
- 3) Yi Chen, Susan Davidson, Carmem Hara, Yifeng Zheng: RRRXS: Redundancy reducing XML storage in relations. the 29th VLDB Conference: 189-200, 2003.
- 4) Serge Abiteboul, Richard Hull, Victor Vianu: Foundations of Databases. Addison-Wesley 1995.