社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MPS－72 (10)
2008－BIO－15 (10)
2008／12／17

ユーザ指定関係に基づいた情報保存を満たすXMLスキーマ進化

橋本 健二　石原 靖哲　藤原 融
大阪大学大学院情報科学研究科

概要 本稿ではユーザ指定関係に基づく情報保存を満たすXMLスキーマ進化のための枠組みを提案する．まず，スキーマにおいてどの要素間の関係や文書順序関係を保存するかを指定するための問合せクラスを定義し，それらの問合せをパラメータとしたXMLスキーマ間の情報保存を定義する．次に，XMLスキーマを部分的に書き換えるためのスキーマ更新操作群を定義する．そして，スキーマと保存問合せが与えられたときに，更新操作ごとにその操作によって更新された後のスキーマがその保存問合せに関して元のスキーマの情報を保存するための十分条件を与える．

XML schema evolution preserving information based on user-specified relationship

Kenji Hashimoto　Yasunori Ishihara　Toru Fujiwara
Graduate School of Information Science and Technology, Osaka University

Abstract: This paper proposes a framework for XML schema evolution preserving information based on user-specified relationship. First, we provide a class of queries for specifying relationship to be preserved between elements in an XML schema, and then define information preservation between XML schemas parameterized by given queries in the class. Next, we propose schema update operations each of which partially updates schemas. Then, given a schema and queries to be preserved, for each operation, we present a sufficient condition for the schema updated by the operation to preserve the information of the original schema.

1 Introduction

In recent years, XML has become a promising data format for various applications. XML allows users to develop their own format of XML documents, which is defined by an XML schema. Although XML itself is designed to be schemaless, schemas are useful and necessary for converting and/or integrating XML documents, optimizing queries, and so on.

Schemas are often required to change to accommodate new functionalities and data types in databases. Further, since in many applications one does not want to lose the information of the instance valid to the original schema, a mapping from the document to a new one valid to the updated schema should also preserve information.

In [2], we have defined *preservation of schema's expressive power* for XML schemas based on the tree embedding relation. We have proposed update operation sets for XML schema evolution, and then we address their properties on preservation of schema's expressive power. Though it is ensured that a schema updated by using the operations preserves the original schema's expressive power, this framework strictly disallows schema changes with some loss of the ancestor-descendant relationship or the document order. Ref. [3] has presented a set of primitive operations for restructuring XML schemas and their instances. For each primitive operation on XML schemas, a document transformation from instances valid against the original schema to ones valid to the schema updated by the operation is defined. Though the operation set enables restructuring XML schemas more flexibly than the ones in [2], information preservation is not addressed.

In this paper, we define information preservation between XML schemas based on user-specified relationship, which is more flexible than preservation of schema's expressive power in [2]. This paper proposes schema update operations for XML schema evolution and a sufficient condition for the schema updated by each operation to preserve the information of the original schema.

2 Preliminaries

XML documents are often represented by unranked labeled ordered trees, simply called trees henceforth. For an alphabet Σ, a Σ-*tree* is a tree whose nodes are labeled with symbols in Σ. The set of all nodes of a tree t is denoted by $V(t)$. The root of t is denoted by $root(t)$. For a node $u \in V(t)$, let $\lambda(u)$ denote the label of u. Also, let $chd(u)$ and $dsc(u)$ denote the sets of the children and the descendants of u, respectively.

In this paper, a regular expression may contain the following operators: $*$ (repetition), $+$ (one or more repetition), ? (optional occurrence), and $|$ (alternation), but not the constant $\emptyset$ denoting the

empty language. Let $Occ(r)$ denote the set of the symbols occurring in r. Let $L(r)$ denote the regular (string) language represented by a regular expression r. The regular expression obtained from r by replacing a subexpression r' with r'' is denoted by $r[r' \leftarrow r'']$.

XML schemas are often modeled by regular tree grammars [4]. According to [4], regular tree grammars, regular tree languages, and some related notions are defined.

Definition 1 *A* regular tree grammar *is a 4-tuple* $G = (N, T, S, P)$*, where:*

- N *is a finite set of* nonterminals*,*
- T *is an alphabet, i.e., a finite set of* terminals*,*
- $S \subseteq N$ *is a set of* start symbols*, and*
- P *is a set of* production rules *in the form of* $A \rightarrow a(r)$*, where* $A \in N$*,* $a \in T$*, and* r *is a regular expression over* N*.* A *is the* left-hand side*,* $a(r)$ *is the* right-hand side*, and* r *is the* content model *of this production rule.* □

Let $\mathcal{N}(G)$ and $\mathcal{P}(G)$ denote the set of nonterminals and production rules, respectively.

Definition 2 *Let* $G = (N, \Sigma, S, P)$ *be a regular tree grammar and* t *be a* Σ*-tree. An* interpretation I^t_G of t against G *is a mapping from* $V(t)$ *to* N *with the following properties:*

- *If* $u = root(t)$*, then* $I^t_G(u) \in S$*.*
- *For each* $u \in V(t)$ *and its children* $u_0, u_1, \ldots, u_m$*, there exists a production rule* $A \rightarrow a(r)$ *in* G *such that*
  - $I^t_G(u) = A$*,*
  - $\lambda(u) = a$*, and*
  - $I^t_G(u_0) I^t_G(u_1) \cdots I^t_G(u_m) \in L(r)$*.* □

For a regular tree grammar $G = (N, \Sigma, S, P)$ and a tree $t \in \mathcal{T}_\Sigma$, if there exists an interpretation of t against G, then t is *valid* against G. Let $TL(G)$ denote the set of trees valid against G, i.e., the *tree language generated* by G.

Without loss of generality, we assume that no two production rules have the same nonterminal in the left-hand sides. Moreover, we assume that every regular tree grammar contains only useful nonterminals, i.e., every nonterminal can generate a nonempty language and can be reachable from a start symbol. Given a regular tree grammar G, the *dependency graph* $Gr(G)$ of G is a directed graph $(\mathcal{N}(G), E)$ such that $E = \{(A, B) \mid A \rightarrow a(r) \in \mathcal{P}(G), B \in Occ(r)\}$.

For $A \in \mathcal{N}(G)$, let $Symb(A)$ denote the terminal a such that $A \rightarrow a(r) \in \mathcal{P}(G)$. Also, for a regular expression r over $\mathcal{N}(G)$, let $Symb(r) = \{Symb(A) \mid A \in Occ(r)\}$. Then, a subclass of regular tree grammars, called single type tree grammars [4], is defined as follows:

Definition 3 *Two different nonterminals* A_1 *and* A_2 *are said to be* competing *with each other if* $Symb(A_1) = Symb(A_2)$*. A regular tree grammar is* single-type *if in the content model of each production rule no two competing nonterminals occur and no two start symbols are not competing with each other.* □

In this paper, we use a subclass $\mathcal{G}_S$ of single type tree grammar such that the content model of each production rule is duplicate-free and in star normal form [1]. A regular expression is *duplicate-free* if the same symbol does not occur more than once. A regular expression r is in *star normal form* if for each starred subexpression r^*_s of r, $\epsilon \notin L(r_s)$ and for every symbol a and any words u, v, and w, $\{au, v, vaw\} \not\subseteq L(r_s)$. For example, $(a|b)^*$ is in star normal form but $(a^*b^*)^*$ is not.

3 Information preservation

We define *information preservation* between XML schemas based on user-specified relation.

Relation between elements We define a class R_G of R-expressions with respect to an XML schema $G \in \mathcal{G}_S$ to specify relations to be preserved between elements in G.

Definition 4 *The syntax of* R-expressions *with respect to* G *is as follows:*

$$R_G ::= \text{“(”}\ A_p\ \text{“}\sharp\text{”}\ r\ \text{“,”}\ A_1\ \text{“,”}\ A_2\ \text{“)”}$$

where A_p*,* A_1*, and* A_2 *are distinct nonterminals in* $\mathcal{N}(G)$ *and* r *is a subexpression of the content model* r_p *of* $A_p \rightarrow a(r_p) \in \mathcal{P}(G)$*.*

Definition 5 *For* $t \in TL(G)$ *and* $e = (A_p \sharp r, A_1, A_2) \in R_G$*, define*

$$\begin{aligned} e(t) = \{&(v_1, v_2) \mid v_p \in V(t), I^t_G(v_p) = A_p, \\ &v'_1 \text{ and } v'_2 \text{ occur in the subsequence} \\ &\text{of the children of } v_p \text{ that matches } r, \\ &v_1 \in dsc(v'_1), v_2 \in dsc(v'_2), \\ &I^t_G(v_1) = A_1, I^t_G(v_2) = A_2\}. \end{aligned}$$

Also, for $R' \subseteq R_G$*, let* $R'(t) = \bigcup_{e \in R'} e(t)$*.*

Document order between elements We use a subset N^o of the set $\mathcal{N}(G)$ of nonterminal symbols to specify a document order to be preserved between elements in G.

Definition 6 *Let* O^t_{doc} *denote the document order on* $t \in TL(G)$*. For* $N^o \subseteq \mathcal{N}(G)$*, define* $O^{N^o}_G(t)$ *as the least fixed point of the following function* f*:*

$$\begin{aligned} f(O) = \{&(v_1, v_2) \mid A \in N^o, v_p \in V(t), I^t_G(v_p) = A, \\ &v_1, v_2 \in chd(v_p), (v_1, v_2) \in O^t_{doc}\} \\ \cup\{&(v_c, v) \mid v_c \in chd(v_p), (v_p, v) \in O\} \\ \cup\{&(v, v_c) \mid v_c \in chd(v_p), (v, v_p) \in O\}. \end{aligned}$$

Information preservation

Definition 7 *Suppose that two schemas $G_1, G_2 \in \mathcal{G}_S$, a family $\mathcal{R}_1$ of disjoint finite sets of R-expressions with respect to G_1, and a subset N_1^o of $\mathcal{N}(G_1)$ are given. We say that G_2 preserves the information of G_1 with respect to $(\mathcal{R}_1, N_1^o)$ if $\exists f_d : TL(G_1) \to TL(G_2)$, $\exists \mathcal{R}_2$, $\exists f_R : \mathcal{R}_1 \to \mathcal{R}_2$, $\exists N_2^o \subseteq \mathcal{N}(G_2)$, $\forall t \in TL(G_1)$, $\exists M \subseteq V(t) \times V(f_d(t))$, the following three conditions are satisfied:*

1. $\forall R \in \mathcal{R}_1\ \forall v_1, v_2 \in R(t)$, $\exists v_1', v_2' \in V(f_d(t))$ $(\{(v_1, v_1'), (v_2, v_2')\} \subseteq M)$,
2. $\forall R \in \mathcal{R}_1\ \left(F^M(R(t)) = f_R(R)(f_d(t))\right)$, *where* $F^M(C) = \{(v_1', v_2') \mid (v_1, v_2) \in C, \{(v_1, v_1'), (v_2, v_2')\} \subseteq M\}$, *and*
3. $\forall v_1, v_2 \in V(t)$, $\forall v_1', v_2' \in V(f_d(t))$ $(\{(v_1, v_1'), (v_2, v_2')\} \subseteq M \wedge (v_1, v_2) \in O_{G_1}^{N_1^o}(t) \Rightarrow (v_1', v_2') \in O_{G_2}^{N_2^o}(f_d(t))\ )$.

4 Schema update operations

In this section, we define schema update operations for XML schema evolution. In the following definition of each operation, let $G_2 = (N_2, T_2, S, P_2)$ be the tree grammar in $\mathcal{G}_S$ obtained by applying the operation to $G_1 = (N_1, T_1, S, P_1) \in \mathcal{G}_S$. Also, we provide a sufficient condition that G_2 preserves information of G_1 with respect to a given $(\mathcal{R}_1, N_1^o)$.

For each operation, we consider as f_d in Definition 7 a natural document transformation such that only portions of an input tree concerned with target rules of the operation are updated according to the operation. For each $t \in TL(G_1)$, a relation M between nodes of t and $f_d(t)$ comply with its transformation policy. Also, we consider the following mapping as f_R: for each $R \in \mathcal{R}_1$ such that $e \in R$ includes symbols concerned with target rules of the operation, f_R maps R to R' such that for each $t \in TL(G_1)$, $R'(f_d(t))$ corresponds to $R(t)$ with respect to M. The condition of each operation is a sufficient condition for such a mapping f_R to exist.

Insert [Def.] Let $p = (A \to a(r)) \in P_1$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r'), X \to x(\epsilon)\}$ such that r' is a regular expression obtained by inserting X before or after a subexpression r_s of r and replacing ϵX and $X\epsilon$ with X, where $X \notin N_1$ and $x \notin Symb(r)$.

Delete [Def.] Let $p = (A \to a(r)) \in P_1$ and $X \to x(\epsilon)$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r')\}$ such that r' is a regular expression obtained by replacing X with ϵ if X occurs without concatenating any other nonterminal symbol in r; otherwise r' is obtained by deleting X from r.
[Condition] For each $R \in \mathcal{R}_1$, there is no expression $(A_q \sharp r_q, A_1^t, A_2^t) \in R$ such that, in $Gr(G_1)$, some path from a start symbol in S to A_1^t or A_2^t through an edge from A_q to A_r includes an edge from A to X, where $A_r \in Occ(r)$.

M-extend [Def.] Let $p = (A \to a(r)) \in P_1$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r[r_s^m \leftarrow r_s^{m'}])\}$, where r_s^m is a subexpression of r, m'=* if m is ? or +; m' =?, +, or * if m is none.
[Condition] For each $R \in \mathcal{R}_1$, if $(A \sharp r_p, A_1, A_2) \in R$, then r_s is a subexpression of r_p, or r_s does not overlap r_p.

C-extend [Def.] Let $p = (A \to a(r)) \in P_1$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r'), X_1 \to x_1(\epsilon), \ldots, X_n \to x_n(\epsilon)\}$, where $X_1, X_2, \ldots, X_n \notin N_1$ and $x_1, x_2, \ldots, x_n \notin Symb(r)$, such that r' is a regular expression obtained by replacing a subexpression r_s of r with $(r_s|r_x)$ or $(r_x|r_s)$ and r_x is a regular expression consisting of only $X_1, X_2, \ldots, X_n$.
[Condition] For each $R \in \mathcal{R}_1$, if $(A \sharp r_p, A_1, A_2) \in R$, then r_p is a subexpression of r_s, or r_p does not overlap r_s.

Restrict [Def.] Let $p = (A \to a(r)) \in P_1$. Suppose that r has a subexpression in the form of r_s^m, where m =+ or m =*. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r[r_s^m \leftarrow r_s^{m'}])\}$, where if m =+, then m' is none, and if m =*, then m' =?.
[Condition] For each $R \in \mathcal{R}_1$, there is no expression $(A_q \sharp r_q, A_1^t, A_2^t) \in R$ such that, in $Gr(G_1)$, some path from a start symbol in S to A_1^t or A_2^t through an edge from A_q to A_r includes an edge from A to X, where $A_r \in Occ(r)$.

Swap [Def.] Let $p = (A \to a(r)) \in P_1$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r[r_1 r_2 \leftarrow r_2 r_1])\}$ where $r_1 r_2$ is a subexpression of r.
[Condition] (1) $A \notin N_1^o$, and (2) for each $R \in \mathcal{R}_1$, if $(A \sharp r_p, A_1, A_2) \in R$, both or neither of r_1 and r_2 are subexpressions of r_p.

Split-path [Def.] Let $p_A = (A \to a(r_A))$ and $p_B = (B \to b(r_B)) \in P$ where B occurs in r_A. Suppose that there exists a rule other than p_A in P whose content model contains B. Then, $P_2 = (P_1 - \{p_A\}) \cup \{A \to a(r_A[B \leftarrow B']), B' \to b(r_B)\}$, where $B' \notin N_1$.

Join-path [Def.] Let $p_A = (A \to a(r_A))$, $p_B = (B \to b(r_B))$, and $p_B' = (B' \to b(r_B)) \in P$, where B' occurs in r_A but B does not in r_A. Then, $P_2 = (P_1 - \{p_A\}) \cup \{A \to a(r_A[B' \leftarrow B])\}$.
[Condition] (1) N_1^o includes both or neither of B and B', and (2) for each $R \in \mathcal{R}_1$, $(A_p \sharp r_p, A_1, A_2) \in R$ where A_p(A_1, and A_2, respectively) is B if and only if $(A_p \sharp r_p, A_1, A_2) \in R$ where A_p (A_1, and A_2, respectively) is B'.

Divide-rules [Def.] Let P_A be the set of all the production rules whose content models contain B. Let $p_B = (B \to b(r))$. Then, $P_2 = (P_1 - P_A) \cup \{B \to b(r_1), B' \to b'(r_2)\}$ $\cup \{A \to a(r[B \leftarrow (B|B')]) \mid (A \to a(r)) \in P_A\}$ such that $L(r_1) \cup L(r_2) = L(r)$ and $L(r_1) \cap L(r_2) = \emptyset$, where $b' \notin T_1$.
[Condition] For each $R \in \mathcal{R}_1$ and each $(B\sharp r_p, A_1, A_2) \in R$, (1) r_p is a subexpression of r_1 or $Symb(r_p) \cap Symb(r_1) = \emptyset$, and (2) r_p is a subexpression of r_2 or $Symb(r_p) \cap Symb(r_2) = \emptyset$.

Combine-rules [Def.] Let P_A be the set of all the production rules whose content models contain $(B|B')$. Let $p_B = (B \to b(r_1))$ and $p'_B = (B' \to b'(r_2)) \in P$. Then, $P_2 = (P_1 - P_A) \cup \{B'' \to b(r)\} \cup \{A \to a(r'[(B|B') \leftarrow B'']) \mid (A \to a(r')) \in P_A\}$, where $L(r) = L(r_1) \cup L(r_2)$ and $B'' \in N_1$.
[Condition] (1) N_1^o includes both or neither of B and B', and (2) for each $R \in \mathcal{R}_1$, (2-1) both or neither of $(A_p\sharp r_p, B, A_2)$ and $(A_p\sharp r_p, B', A_2)$ (respectively $(A_p\sharp r_p, A_1, B)$ and $(A_p\sharp r_p, A_1, B')$) are in R, (2-2) if $(B\sharp r', A_1, A_2) \in R$ (respectively $(B'\sharp r', A_1, A_2) \in R$), then (2-2a) r' is a subexpression of r, and (2-2b) $Symb(r') \cap Symb(r_2) = \emptyset$ (respectively $Symb(r') \cap Symb(r_1) = \emptyset$), or $(B'\sharp r', A_1, A_2) \in R$ (respectively $(B\sharp r', A_1, A_2) \in R$).

Bundle [Def.] Let $p = (A \to a(r)) \in P_1$. Then, $P_2 = (P_1 - \{p\}) \cup \{X \to x(r''), A \to a(r[r'' \leftarrow X])\}$ such that r'' is a subexpression of r, where $X \notin N_1$ and $x \notin Symb(r)$.
[Condition] For each $R \in \mathcal{R}$, if $(A\sharp r_p, A_1, A_2) \in R$, r_p is a subexpression of r'' or r'' is a subexpression of r_p.

Collapse [Def.] Let $p = (A \to a(r'))$ and $X \to x(r'') \in P_1$ where X occurs in r' and $Symb(r') \cap Symb(r'') = \emptyset$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r'[X \leftarrow r''])\}$.
[Condition] (1) Both or neither of A and X are in N_1^o, and (2) for each $R \in \mathcal{R}$, $(X\sharp r_p, A_1, A_2) \notin R$, $(A_p\sharp r_p, X, A_2) \notin R$, and $(A_p\sharp r_p, A_1, X) \notin R$.

Fact [Def.] Let $p_1 = (A \to a(r))$ and $p_2 = (X \to x(r')) \in P_1$ such that X occurs in r, r' is of the form $r_f r''$ where r_f is a proper subexpression of r', and $Symb(r) \cap Symb(r_f) = \emptyset$. Assume that an edge from A to X is a unique edge to X in $Gr(G_1)$. Then, $P_2 = (P_1 - \{p_1, p_2\}) \cup \{A \to a(r[X \leftarrow r_f X']), X' \to x(r'')\}$, where $X' \notin N_1$.
[Condition] (1) For each $R \in \mathcal{R}$, if $(X\sharp r_p, A_1, A_2) \in R$, r_p is a subexpression of r_f or r_f is a subexpression of r_p, and (2) A and X are not in N_1^o.

Defact [Def.] Let $p_1 = (A \to a(r))$ and $p_2 = (X \to x(r')) \in P_1$ such that $r_f X$ occurs in r, and $Symb(r_f) \cap Symb(r') = \emptyset$. Assume that an edge from A to X is a unique edge to X in $Gr(G_1)$. Then, $P_2 = (P_1 - \{p_1, p_2\}) \cup \{A \to a(r[r_f X \leftarrow X']), X' \to x(r_f r)\}$, where $X' \notin N_1$.
[Condition] (1) For each $R \in \mathcal{R}$, if $(X\sharp r_p, A_1, A_2) \in R$, then r_p is a subexpression of r_f or $r_f X$ is a subexpression of r_p, and (2) neither A nor X is in N_1^o.

Nest [Def.] Let $p = (A \to a(r))$ such that r has a subexpression r_h^* or r_h^+, and r_h has a subexpression r_g or $r_g^?$. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r[r_g \leftarrow r_g^+, r_g^? \leftarrow r_g^*])\}$.
[Condition] (1) A is not in N_1^o, and (2) for $R \in \mathcal{R}_1$ and $(A_p\sharp r_p, A_1, A_2) \in R$, (2a) r_h^* or $r_h^?$ is a subexpression of r_p, or (2b) not both of r_h^* and $r_h^?$ overlap r_p.

Unnest [Def.] Let $p = (A \to a(r))$ such that r has a subexpression r_h^* or r_h^+, and r_h has subexpression r_g^* or r_g^+. Then, $P_2 = (P_1 - \{p\}) \cup \{A \to a(r[r_g^+ \leftarrow r_g, r_g^* \leftarrow r_g^?])\}$.
[Condition] (1) A is not in N_1^o, and (2) for each $R \in \mathcal{R}_1$, if $(A\sharp r_p, A_1, A_2) \in R$, then (2a) r_h^* or $r_h^?$ is a subexpression of r_p, or (2b) not both of r_h^* and $r_h^?$ overlap r_p.

5 Conclusion

We have proposed schema update operations for XML schema evolution and presented a sufficient condition for the schema updated by each operation to preserve the information with respect to a given $(\mathcal{R}, N^o)$.

Our future work is to make the method of specifying information to be preserved more expressive. Another future work is to extract information to be preserved from queries and transformations for documents valid to the original schemas.

References

[1] A. Brüggemann-Klein and D. Wood, One-unambiguous regular languages, Information and Computation, Vol. 142, No. 2, pp. 182–206, 1998.

[2] K. Hashimoto, Y. Ishihara, and T. Fujiwara, Update operation sets for schema evolution in XML database and their properties on preservation of schema's expressive power, IEICE Trans. Inf. & Syst., Vol. J90-D, No. 4, pp. 990–1004, 2007 (in Japanese).

[3] J. Liu, H. Park, M. W. Vincent, and C. Liu, A formalism of XML restructuring operations, ASWC 2006, pp. 126–132, 2006.

[4] M. Murata, D. Lee, M. Mani, and K. Kawaguchi, Taxonomy of XML schema languages using formal language theory, ACM TOIT, Vol. 5, No. 4, pp. 660–704, 2005.