5S-1

知識処理に基づく XML データベースシステムの試作

駅昌弥 † 大囿忠親 ‡ 新谷虎松 ‡
† 名古屋工業大学 知能情報システム学科 ‡ 名古屋工業大学大学院 工学研究科 情報工学専攻
e-mail: {eki, ozono, tora}@ics.nitech.ac.jp

1 はじめに

XML データの管理には，リレーショナルデータベースや XML データベースが用いられることが多い．リレーショナルデータベースは，取り扱う XML データが形式の決まっている構造データの場合には検索性能が良いが，構造の定義が困難な非構造データでは，検索性能が大幅に低下する．また，データ構造に変更があった場合，それに合わせて表構造を変更しなければならないという欠点がある．XML データベースにおいて一般的に問題とされるのは，処理速度が遅い点やデータ量が膨大になる点，大規模なデータに対しての検索が難しい点などである．また，構造化されたデータの処理速度は，リレーショナルデータベースに比べて極めて低速である．全文検索は，該当する構造やノードのみを取得することしかできず，柔軟な問い合わせを行えない．

これらの問題を解決するために，本論文では，一般的な XML 文書，特に事前に仕様が確定していない，及び仕様の変更が頻繁に起こるようなスキーマレスの XML 文書を述語や知識テーブル [1] に変換し，論理型言語である Prolog に準ずる問い合わせ言語を用いて知識推論を行うことで，XML の柔軟性を損なわずに高速な検索を行うシステムを提案する．

2 大規模知識処理に基づく XML データベースシステム

2.1 システムの概要

図 1 は，本システムの構成を示す図である．ユーザは，その環境に従って XML データベースに対して 3 通りのアプローチをとることができる．Javascript，Perl，PHP を用いてブラウザからリクエストを送る場合は，Java サーブレットに対してリクエストを送る．Java プログラムに組み込みたいときは，Java プログラムからリクエストを送る．コマンドラインで動かしたいときは，コマンドラインからリクエストを送る．前者 2 つは，ソケットをリクエストし，そのレスポンスを表示する．

図 2 は，XML データベースの構成を示す図である．XML データベースでは，データベースごとにデータを格納することができる．最初に，データベース名を決め，そのデータベースに格納する XML 文書を選択する．次に，その XML 文書を知識テーブルや述語に変換し，それらの情報をデータとして書き出す．完成したデータベースに対して，検索などの問い合わせを行ったり，XML 文書の追加や知識の定義などを行う．

2.2 知識テーブル

本システムでは，知識推論に基づく XML データベースを実現するために，知識テーブル [1] を利用する．知識テーブルは，Prolog を用いて知識推論を行うのに適している．

Building an XML Database System Based on Knowledge Processing
Masaya EKI, Tadachika OZONO, Toramatsu SHINTANI
Dept. of Intelligence and Computer Science, Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya 466-8555 JAPAN

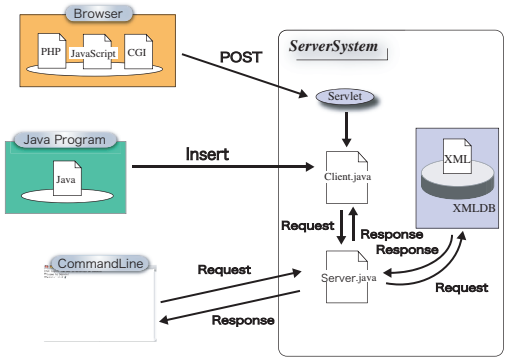


図 1: システム構成図

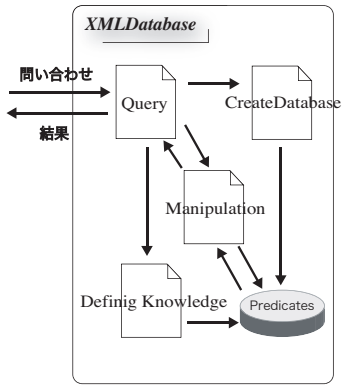


図 2: XML データベース構成図

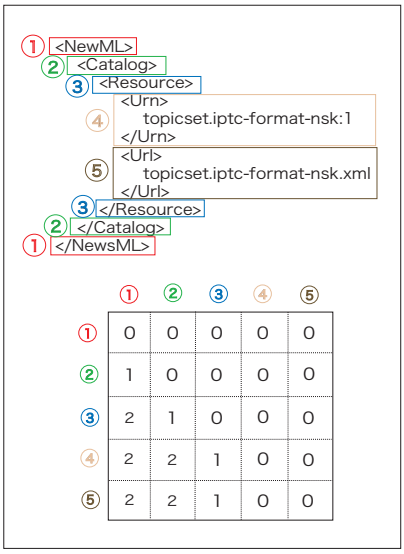
知識テーブルは，木構造やネットワークモデルなど有向グラフにおいて，知識の関係をテーブルで表現するものである．XML を有向グラフであるとすると，ルートから葉ノードへ辿る有向グラフであると考えられる．

図 3 に XML の知識テーブル表現の例を示す．知識テーブルのテーブル要素に書かれている数字は，横列の ID から見て縦列の ID との関係を表している．この数字は ‘1’,‘2’,‘0’ の 3 種類で表され，それぞれ，直接の子ノード，孫以下のノード，無関係なノードという意味である．例えば，図 3 の知識テーブルではノード 2 とノード 3 が親子関係であることがわかる．

2.3 本システムにおける問い合わせ言語

本システムで用いられる問い合わせ言語は，論理型言語である Prolog に準ずる．変数はタグに大文字が許されている (本来の XML の定義では小文字のみ) ので，Prolog の変数と異なって ‘_ (アンダースコア)’ で始まる英数字の文字列である．例えば，_X, _hohe, _12345 などは全て変数である．

以下の問い合わせの例は，‘hoge’ という値を持つノードと，‘hogehoge’ という値を持つノードが共通の親を持つと



	①	②	③	④	⑤
①	0	0	0	0	0
②	1	0	0	0	0
③	2	1	0	0	0
④	2	2	1	0	0
⑤	2	2	1	0	0

図 3: 知識テーブル

き，それらのノードを出力する問い合わせである．

X = (element() = 'hoge'),
Y = (element() = 'hogehoge'),
parallel(_X, _Y), parent(_Z, _X).

言語の特徴として，変数やアトムとパターンマッチングを行うことで，知識推論を可能とする．また，ユーザが新たな述語を定義することができるので，個々のユーザにとって扱いやすいデータベースを実現できる．

2.4 実行時間

図 4 は，ノード検索とタグ構造検索の実験結果である．どちらの検索も，全ノードの 20%〜25%を取得する問い合わせを 100 回問い合わせ，それらの平均を取った値が表に記されている．実験に用いられた XML 文書は，ニュースの表現に用いられる NewsML の文書を仕様した．実験に使用した計算機は，ノート型の PC で，OS は WINDOWS XP Professional，CPU は 1.8 GHz Pentium4 Processor，メモリは SDRAM 1 GB である．検索速度は XML 文書のサイズではなくノードの数に依存するので，この実験ではノード毎の検索速度を測定した．

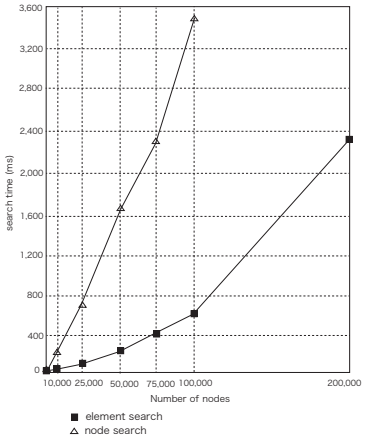


図 4: 知識テーブル

3 本システムの応用事例

図 5 は同じ “NIT college” の学生を別のタグ構造で表現した例である．一般に，ユーザやアプリケーションによって，同じ種類の意味だが異なったタグ構造が作られる．そのため，図 5 のタグ構造の上側しか知らないユーザが，上下共に含む XML 文書に “NIT college” の学生についての問い合わせを行うと，上側の構造のみが検索結果として出力される場合がある．この検索結果は，ユーザにとって不完全である．

```
<college>
  <NIT>
    <student>
      <name>Masaya-Eki</name>
      <number>12345678</number>
      <address>Japan</address>
    </student>
  </NIT>
</college>
```

```
<college name=" NIT" >
  <student>
    <firstname>Masaya</firstname>
    <lastname>Eki</lastname>
    <age>21</age>
  </student>
</college>
```

図 5: 同じ種類の意味だが構造の異なる XML 文書

例えば，図 5 のタグ構造が格納されている XML データベースが存在し，ユーザが “Masaya-Eki” という学生の住所と年齢が知りたいと思ったとき，図 5 の上下の構造を理解しているならば上下の構造に対して問い合わせを行うことで住所と年齢を取得できる．しかしながら，同じ種類の意味の構造が複数パターンあることを知らないユーザは必要な情報を取得することができない．本システムでは，大学生に関する知識を定義し，知識に問い合わせを行うことで，ユーザは実構造を知らなくてもマッチしたタグ構造全てにアクセスできる．

4 おわりに

本論文では，高速な検索及び柔軟な問い合わせを可能とする XML データベースについて述べた．本アプローチでのデータベースは，内部構造をユーザが知らなくても検索を行うことができる．ここでは，Prolog に準ずる問い合わせ言語を用いることで知識推論を実現した．本データベースは，既存の XML データベースの操作 (検索やノード追加など) を行えるだけでなく，ユーザが定義した知識を用いた推論を行うことができる．知識推論を用いることにより，同じ種類の意味でタグ構造が異なるデータに対して，ユーザはそれら全てのデータにアクセスし，問い合わせを行うことができる．また，様々な知識をユーザが定義し，それらの知識を組み合わせることにより，格納されているデータから法則や規則性を発見することができる．

参考文献

[1] Toramatsu Shintani, “Knowledge Table: An Approach to Speeding up the Search for Relational Information in Knowledge Base”, Journal of Information Processing, Vol. 13, No4, 1990.

[2] Dunren Che, Karl Aberer, M. Tamer özsu, “Query optimization in XML structured-document databases”,VLDB 2006.

[3] Margaret G. Kostoulas, Morris Matsa, Noah Mendelsohn, Eric Perkins, Abraham Heifets , “XML Screamer: An Integrated Approach to High Performance XML Parsing, Validation and Deserialization”, WWW 2006.