A5–1

XML データベース XRel の実装とその評価

藤井眞吾 † 天笠俊之 † 吉川正俊 †,‡ 植村俊亮 †

† 奈良先端科学技術大学院大学 情報科学研究科
‡ 国立情報学研究所 ソフトウェア研究系

† 〒 630-0101 奈良県生駒市高山町 8916-5
‡ 〒 101-8430 東京都千代田区一ツ橋 2-1-2

{shingo-f, amagasa, yosikawa, uemura}@is.aist-nara.ac.jp

あらまし XML はネットワークを通じて交換される文書やデータの共通フォーマットとして爆発的に普及している．XML で記述される文書数は増大することが予想されるため，大量の XML 文書を扱うことのできる XML データベースの研究，開発は急務である．本研究では，文書型定義に依存しない XML 文書の格納，検索手法である XRel について，その実装と性能評価を行う．まず，任意の XPath 式を XRel のデータベースに問合せるために SQL 式に変換するためのモジュールを Java を用いて実装する手法およびそのモジュールを用いた応用について述べる．次に，XRel の性能評価を，XML データベースの性能評価である Xmark によって行う．

キーワード XRel, XML, 関係データベース, 格納, 検索, 性能評価

An implementation and Performance Evaluation of an XML Database XRel

Huzii Singo†, Amagasa Toshiyuki†, Yoshikawa Masatoshi†,‡, and Uemura Shunsuke†

† Graduate School of Information Science, Nara Institute of Science and Technology
‡ Software Research Division, National Institute of Informatics

†8916-5 Takayama, Ikoma, Nara 630-0101, Japan
‡ 2-1-2 Hitotsubashi, Chiyoda, Tokyo 104-8430, Japan

{shingo-f, amagasa, yosikawa, uemura}@is.aist-nara.ac.jp

Abstract XML is in widespread use as a common format of exchanged documents through the Internet, and is expected to increase. So XML database which can treat massive XML documents is a prime task. In this article we implement and benchmark XRel, which is independent of DTD. First we introduce a method to implement a module to translate a XPath expression to SQL query. Next we benchmark XRel by Xmark, which benchmarks XML database.

Key words XRel, XML, Relational Database, Storage, retrieval, Benchmark

1 はじめに

XML (Extensible Markup Language) は，構造化文書を記述するためのメタ言語であり，ネットワークを通じて交換される文書やデータの共通フォーマットとして普及している．今後 XML で記述された文書の増大が予想されるため大量の XML 文書を効率的に管理・運用できる XML データベースの研究・開発が急務である．

XML データベースの実現手法としては，関係データベース，対象指向 (object oriented) データベースや XML 専用のデータベースを用いる方法等が提案されているが，中でも関係データベースを用いる方法は，1) 最も普及しているデータベースであり稼働中のシステムが多数存在する，2) 大量のデータが関係データベース上に蓄積されている，3) 過去 20 数年に渡る研究による，問合せ最適化や論理作業単位 (トランザクション) 管理等の技術の蓄積があることなどから， XML を既存の計算機資源，情報資源と連携して利用するのに有利である．

関係データベースを用いて XML 文書を格納および検索する手法として XRel [1, 2] がある． XRel は文書型定義 (DTD) に依存せず，任意の整形式の XML 文書を関係データベースに格納することができる手法である．XML 文書を関係データベースに格納するために関係スキーマを DTD を用いて設計する手法があるが， XRel ではこれに対して DTD が必要ない， DTD を変更しても関係スキーマを変更する必要がない等の利点を持つ．

そこで本研究では XRel により格納された XML 文書に対して行われる XPath 式による問合せを SQL 式に変換するモジュールを実装し，その変換モジュールを用いた応用としてシェイクスピア戯曲の検索システムを構築した． また XML データベースの性能評価指標である Xmark を用いて XRel の性能を評価した．

本論文の構成は次のとおりである．2 節では関連研究について述べ， 3 節では XRel の紹介を行う． また 4 節では本研究で実装した XPath 式を SQL 式による XRel 問合せ式に変換するモジュールと，その変換モジュールを用いた応用システムについて述べ，6 節では XRel の XML データベースとしての性能評価について報告する．最後に 7 節ではまとめと今後の課題について述べる．

2 関連研究

2.1 関係データベースに基づいた XML データベースの実現手法

本小節では XML データベースの写像技術について述べる． XML 文書を格納し，検索，更新などの操作をするためのデータベースシステムは，研究用，商用ともに開発が行われている． XML 文書のデータベースへの格納法は大きく分けて， XML 文書をそのまま格納する方法と XML 文書を分解してデータベーススキーマに写像する方法の二通りに分けることができる．

既存のデータベースを用いるのではなく，XML のデータ仕様をもとにして専用のデータベースを構築することが考えられる． XML 文書をそのまま格納する方法を採用するデータベースは，ネイティブ XML データベースと呼ばれることもある． 関係データベースシステムの場合は，基本的に CLOB (Character Large Object) として格納されることが多い [3]． 例えば Oracle 9i では，論理的には新たに XML 型が導入されている[1]．

XML 文書を分解してデータベーススキーマに写像する方法としては，構造写像による方法，モデル写像による方法の二つの方法が考えられる． 構造写像による方法ではデータベーススキーマは XML 文書の論理構造 (あるいは文書型定義が存在する場合は文書型定義) を表現する [4]． また，モデル写像による方法ではデータベーススキーマは XML データの構成要素を表現する [5, 6]. この方法では，任意の XML 文書の木構造を格納するために固定したデータベーススキーマを用いる．

2.2 XML データベースの性能評価技術

本小節では XML データベースの性能評価技術について述べる． XML データベースを実現するための多くの手法が提案されているため，それらの性能を定量的に比較するための性能評価指標が開発されつつある．文献 [7] は XML 性能評価指標が備えるべき測定基準を次のようにまとめている． (1) 一括格納 (bulk loading), (2) 再構成 (reconstruction), (3) 経路横断 (path traversal), (4) 型変換 (casting), (5) 失要素 (missing element), (6) 順序接続 (ordered access), (7) 参照 (references), (8) 結合 (joins), (9) 包含，全文検索 (containment, full-text search)

XML 問合せ処理器の性能評価の枠組として，5 節で説明する Xmark [8] が開発されている． このような問合せ処理器以外の面の評価項目としては，システム基盤，所有の費用がある [7]． システム基盤の評価項目としては「接続規約 (access protocol)」,「検索結果の表示」,「利用者が複数いる場合のデータ処理能力」などがある [9].

3 XRel の概要

XRel は関係データベースを用いた XML 文書の汎用的な格納と検索の手法である． 格納については， XML 文書を構文解析して得られる木構造を節点単位で分割し，節点の型 (要素，属性，文字列) に応じてデータベースの関係表に格納する． この手法を用いると， DTD や要素型に依存することなく，整形式のあらゆる XML 文書を格納することができる． さらに問合せ処理の高速化のために，データベース管理システムで提供されている B+ 木， R 木などの索引機構を利用することができる． 検索については，関係データベース自体への問合せは SQL によりなされるため， XML のための問合せ言語による

[1]"Oracle Technology Network" ⟨URL: http://otn.oracle.com/tech/xml/⟩

```
<book>
    <title>XML and Database</title>
    <authors>
        <author affiliation="NAIST">Yamada Taro</author>
        <author affiliation="RAIST">Sugita Ziro</author>
    </authors>
    <summary>XML stands for Extensible Markup Language</summary>
</book>
```

図 1: XML 本体

問合せ式を SQL 式に変換する必要がある． この格納手法は関係データベースを拡張することなく実現でき，また検索手法については問合せ言語の前処理器を付加することで実現できる．

3.1 XRel による XML 文書の格納

XML 文書は根節点，要素節点，属性節点，文字列節点の 4 種類の節点からなる木構造で表すことができる．XRel では，これらの節点のうち要素 (Element)，属性 (Attribute)，文字列 (Text) のそれぞれの節点に関する表，および要素の経路表現 (Path) に関する表の 4 つの表に格納する． その際，節点のデータは「各要素や属性に関する情報の根要素 (XML 文書の木構造の根節点) からの経路」と「先頭文字からのバイト数」の組合せで表現される．

図 1 で表される XML 文書を XRel により関係データベースに格納したものを表 1– 4 に示す．

表 1: Element 表

docID	*pathID*	*start*	*end*	*index*	*reindex*
1	1	0	238	1	1
1	2	9	39	1	1
1	3	43	168	1	1
1	4	56	103	1	2
1	4	108	155	2	1
1	6	172	230	1	1

表 2: Attribute 表

docID	*pathID*	*start*	*end*	*value*
1	5	57	57	NAIST
1	5	109	109	RAIST

表 3: Text 表

docID	*pathID*	*start*	*end*	*value*
1	2	16	31	XML and Database
1	4	84	94	Yamada Taro
1	4	136	146	Sugita Ziro
1	6	181	220	XML stands for ...

3.2 XRel による XML 文書の検索

XRel では，問合せ処理のため XPath 式を SQL 式に変換して関係データベースで処理させる枠組を提供する．これにより利用者は関係データベースの存在を意識することなしに XML 文書の木構造をもとに XPath, XQuery などの XML 問合せ言語を用いて問合せを行うことができる．

表 4: Path 表

pathID	*pathexp*
1	#/book
2	#/book#/title
3	#/book#/authors
4	#/book#/authors#/author
5	#/book#/authors#@affiliation
6	#/book#/summary

またこれとは別に利用者が直接 SQL 問合せを用いて問合せを行う方法も考えられるが，ここでは触れない．

本論文では XPath 式を SQL 式に変換する手法を実装したので，その詳細を報告する． まず変換手法 [1] について述べる．

XPath 式は次のように表すことができる．

$$A_0\ \{P_0\}+\ A_1\ \{P_1\}+\ \cdots\ A_n\ \{P_n\}*$$

ここで $A_i, P_i (i = 0, \cdots, n)$ はそれぞれ経路表現，述語を表し，$\{\}+, \{\}*$ はそれぞれ一回，零回以上の繰返しを表す．

このような XPath 式について，経路表現 A_0 から経路表現 A_n までを順に有向辺でつなぎ，経路表現 A_i から述語 P_i を有向辺でつないだものを XPathCore グラフという． さらにこの XPathCore グラフ中の経路表現を根節点からいたる経路表現に改めたものを問合せグラフという．

XPath 式を SQL に変換するには，まず中間形式として XPath 式から問合せグラフに変換し，それから問合せグラフを SQL 式に変換する．

例えば，XPath 式

/book[authors/author='Yamada Taro']/title

は，図 2 のような問合せグラフに変換される． 図 2 で，経路表現 A_0, A_1 はそれぞれ「#/book」,「#/book#/title」を表し， A_0 から A_1 への矢印は A_1 が A_0 に包含されることを示す． また述語 P_{00} は A_0 を条件により絞るための述語であり， A_0 から P_{00} への矢印は P_{00} 中の「#/book#/authors#/author」が A_0 に包含されることを示す．

XPathCore グラフに基づき， XPath 式は次の SQL 式に変換される． ここで A_0 に関する表は Element 表の e0 と Path 表の p0， P_{00} に関する表は Text 表の t00 と Path 表の p00， A_1 に関する表は Element 表の e1 と Path 表の p1 である． これらの節点に関する条件はそれぞれ 4, 5 行目，6, 7, 11 行目， 12, 13 行目に示される． また A_0 と P_{00} との包含関係は 9, 10 行目， A_0 と A_1 との包含関係は 15, 16 行目に示される．

```
1:  SELECT DISTINCT e1.docid, e1.start, e1.end
2:  FROM element e0, path p0, text t00, path p00,
3:       element e1, path p1
4:  WHERE p0.pathexp LIKE '#/book'
5:  AND e0.pathid = p0.pathid
6:  AND p00.pathexp LIKE '#/book#/authors#/author'
```

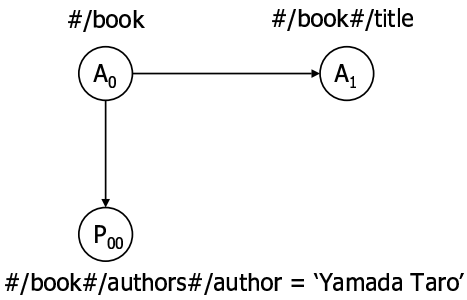

図 2: 問合せグラフ

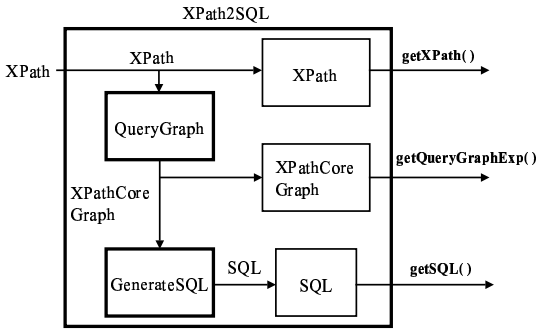

図 3: XPath2SQL

```
 7:  AND t00.pathid = p00.pathid
 8:  AND e0.docid = t00.docid
 9:  AND e0.st < t00.st
10:  AND e0.ed > t00.ed
11:  AND t00.value LIKE 'Yamada Taro'
12:  AND p1.pathexp LIKE '#/book#/title'
13:  AND e1.pathid = p1.pathid
14:  AND e0.docid = e1.docid
15:  AND e0.st < e1.st
16:  AND e0.ed > e1.ed
17:  ORDER BY e1.docid, e1.st;
```

4 問合せ変換モジュールの実装およびその応用

4.1 XPath 式から XRel 問合せ式への変換モジュール

文献 [1, 2] で紹介された XPath による問合せを XRel の SQL による 問合せ式に変換する手法を実装した．プログラム言語は Java を用い，与えられた XPath 式を区切り記号や単語毎に分解するのには字句解析・構文解析を行うための道具である JavaCC (Java Compiler Compiler) を用いた．

モジュール本体のクラス名は XPath2SQL で，入力は XPath 式で与えられる． 与えられた XPath 式を問合せグラフ表現に変換するクラス QueryGraph と QueryGraph により生成された問合せグラフ表現を SQL 式に変換するクラス GenerateSQL とからなる． このモジュールのデータ処理の流れは図 3 のようになる．

出力のための操作関数としては入力された XPath 式を返す String getXPath() ，問合せグラフを返す String getQueryGraphExp() ， SQL 式を返す String getSQL() を用意した．

前節における XPath 式「/book[authors/author='Yamada Taro']/title」を入力として， QueryGraph により問合せグラフに変換される． 問合せグラフは XPath2SQL 内部で次のように表現される．

A[0] : #/book

P[0][0] : compare　#/book#/authors#/author　equal　'Yamada Taro'

A[1] : #/book#/title

そしてこの問合せグラフは GenerateSQL により前節で示した SQL 式に変換される．

4.2 応用

4.2.1 応用システムの概要

4.1 節で紹介した XPath 式から XRel 問合せ式への変換モジュールを用いた応用として XML 文書の検索システムを構築した． この検索システムは，格納された XML 文書の木構造に基づいて利用者が XPath 式により問合せを行い，システムが XPath 式を SQL 式に変換し，その SQL 式を関係データベースに問い合わせ，検索結果を適宜変換して表示する．

このシステムは図 4 に示す主に三つの部品によりなる．

XPath2SQL 入力された XPath 式を SQL 式に変換するモジュール． (Java による)

Database XML 文書を XRel の格納手法を用いて格納した関係データベース．

Translator SQL 問合せの結果として指定される文書部分を XML 文書から抜き出すモジュール． (Java による．)

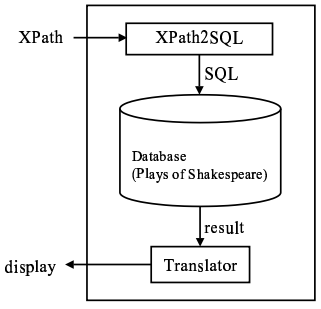

図 4: XML 文書の検索システム

4.2.2 シェイクスピア戯曲の検索システム

この XML 文書の検索システムの応用例としてシェイクスピア戯曲の検索システムを紹介する．使用した XML 文書はシェイクスピアの戯曲を Jon Bosak がタグ付けし

た XML 文書[2]である．このデータは 37 の XML 文書からなり，大きさは総計約 7.65MB である．

次にこのシステムの利用法を示す．

1. 初期画面の「XPath」の文字列入力欄 (text input field) に XPath 式を入力し,「変換」ボタンを押す．
2. SQL 式表示画面になるので，全 37 文書に対する検索か，3 文書に対する検索かを項目選択欄 (radio button) で選択し,「問合せ」ボタンを押す．(図 5 参照)
3. SQL 問合せ結果表示画面になる．XML 文書の部分を切り出すには「表示」ボタンを押す．
4. XML 文書の XPath 式による問合せ箇所の部分文書が表示される．1 に戻る．(図 6 参照)

図 5: SQL 式表示画面

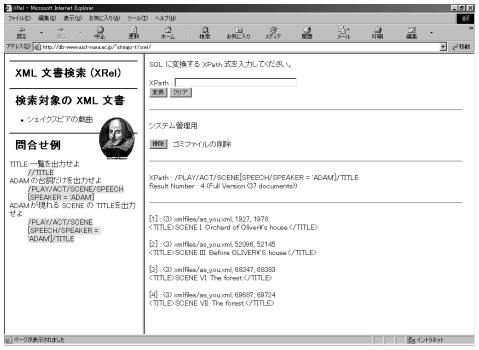

図 6: 結果表示画面

5 Xmark による XRel の性能評価

Xmark 計画 [8][3]は XML データベースの性能評価基準のために，規模変更可能な (scalable) XML 文書の生成と，XQuery による問合せの集合を提供している．ここでは XRel の性能評価を行うために Xmark を用いた．

5.1 Xmark での XML 文書の生成

Xmark で想定する XML 文書はインターネットの競売サイトで実施されるデータベースで利用される文書であり，Xmark ではそのような XML 文書を生成するため xmlgen という XML 文書生成器を設計，公開している．xmlgen の特徴としては，ハードウェアや OS によらず，同一の XML 文書が生成されること，規模変更因子 (scaling factor) により生成する XML 文書の大きさを決定することが挙げられる．

5.2 Xmark での問合せ

Xmark では XML データベースとしての性能を総合的に評価するために，XML の問合せ言語である XQuery を用いた 20 の問合せを提供している．

例えば，XML データベースに対して「順序関係を伴った問合せの処理」を評価するための問合せとして問 2–4 がある．「すべての開催中の競売の初期の増加 (increase) を返す」問合せである問 2 は次のような XQuery で与えられる．

```
  FOR $b
  IN document("auction.xml")/site/open_auctions/op
en_auction
  RETURN <increase> $b/bidder[1]/increase/text()
</increase>
```

5.3 Xmark 問合せの XRel 問合せへの変換

現時点では XQuery による問合せを XRel における SQL 式による問合せに機械的に変換する一般的な手法はない．したがって，Xmark で提案される問合せは手作業で XRel 問合せに変換した．前節における問 2 を XRel 問合せに変換したものを次に示す．

```
  SELECT '<increase>'||t1.value||'</increase>'
  FROM element e0, pth p0, txt t1, pth p1
  WHERE p0.pathexp LIKE '#/site#/open_auctions#/op
en_auction#/bidder'
  AND e0.pathid = p0.pathid
  AND e0.idx = 1
  AND p1.pathexp LIKE '#/site#/open_auctions#/open
_auction#/bidder#/increase'
  AND t1.pathid = p1.pathid
  AND e0.st < t1.st
  AND e0.ed > t1.ed
  ORDER BY t1.st;
```

6 実験

5 節で紹介した Xmark を用いて XRel の性能評価を行った．実験に使用した計算機および環境を表 5 に示す．

表 5: 実験に使用した環境

中央演算装置	Pentium 4 (1.8GHz)
主記憶容量	1GB
基本ソフトウェア	MIRACLE LINUX Standard Edition Version2.0
データベース	Oracle9i Database Standard Edition Release1 (9.0.1) for Linux

6.1 比較対象

Oracle9i では XML にネイティブに対応しているため，当初 XRel を Oracle9i の XML 機能と比較する予定であった．しかしながら，Oracle9i では XML 文書の部分検索機能を有しておらず，Xmark の 20 の問合せを一つも実行できず Xmark による性能評価が不可能であることから XRel を Oracle9i の XML 機能と比較することは断念した．

[2] ⟨URL: http://metalab.unc.edu/bosak/xml/eg/shakes200.zip$rangle$

[3] ⟨URL: http://monetdb.cwi.nl/xml/⟩

そこで XML 文書の問合せ処理エンジンである XML Query Engine[4] (以下「XQEngine」という) を選んだ．その理由は，Xmark で用意されている XQuery 式を上記実験環境で実行可能なシステムがほとんどなかったことが理由である．XQEngine は Java により記述され，XPath, XQL, XQuery による問合せを行うことができる．ただし XQuery の規格に完全には対応しておらず，Xmark で用意された 20 の問合せのうち 6 つの問合せが実行可能であった．

6.2 実験結果

Xmark の XML 文書生成器 xmlgen を用い規模変更因子を 0.01 として[5] XML 文書を生成した．XML 文書の大きさは 1161652 バイト (約 1.2M バイト) である．

Xmark で提供されている 20 の問合せ[6]について実験を行った結果を表 6 に示す．表の XRel, XQEngine の欄は各問合せの実行時間である．実行時間はそれぞれ 10 回行った平均値である．(XRel)/(XQEngine) の欄は各問合せについて XRel での実行時間を XQEngine での実行時間で割った値である．

次節，次々節では，実験結果について XRel 単独で，また XRel と XQEngine とを比較して考察を行う．

表 6: 実験結果

Q	XRel (ms)	XQEngine (ms)	(XQEngine)/(XRel)
1	6.8	48.5	7.13
2	73.2	3509.3	47.9
3	599.7	–	–
4	56.0	–	–
6	8.8	2761.9	314
7	33.2	–	–
8	257.7	–	–
9	4383.0	–	–
13	18.2	129.9	7.14
14	245.8	–	–
15	14.6	214.6	14.7
16	95.0	226.8	2.39
17	256.1	–	–
18	13.9	–	–
19	2807.4	–	–

6.3 XRel 単独の考察

XRel の問合せは，指定する節点に対応する経路を検索し，節点同士の包含関係を調べ，文字列照合，順序節などの条件に合致するかを調べる，といった手順で行われる．そこでそれぞれの問合せについて，経路数，文字列照合の数，順序節の数，文字列比較の数を表 7 に示す．

これらの実行時間と問合せの分析より次のことが分かる．

1. 指定する経路数が多い問合せ (Q3, Q8, Q9, Q14) と，条件に文字列同士の比較を有した問合せ (Q3, Q8, Q9) は実行時間を増加させる．
2. 条件に文字列照合 (Q1, Q4) や順序節 (Q2) を有した問合せは実行時間を減少させる．

表 7: 問合せの分析

Q	時間 (ms)	経路	文字列照合	順序節	文字列比較
1	6.8	3	1	0	0
2	73.2	2	0	1	0
3	599.7	5	0	2	1
4	56.0	4	2	0	0
6	8.8	1	0	0	0
7	33.2	3	0	0	0
8	257.7	4	0	1	1
9	4383.0	9	0	2	2
13	18.2	3	0	0	0
14	245.8	4	1	0	0
15	14.6	1	0	0	0
16	95.0	3	0	0	0
17	256.1	3	0	0	0
18	13.9	1	0	0	0
19	2807.4	3	0	0	0

前者については XRel の問合せでは指定する経路数に対してそれぞれ表が用意される．さらにそれらの結合演算が行われるので，経路数の増加や結合のコストを増加させる文字列同士の比較により実行時間が増加すると考えられる．また後者は，文字列照合や順序節の条件により表の組が結合前に絞られるため，これらの条件を有することにより実行時間が減少すると考えられる．

6.4 XRel と XQEngine との比較による考察

XRel, XQEngine の両方について意味のある実験を行うことができた問合せは Q1, Q2, Q6, Q13, Q15, Q16 の 6 つの問合せである．これら 6 つの問合せについてはすべてにおいて XRel の方が XQEngine よりも高速な処理をしている．ここでは実行時間の差が大きい Q2, Q6 と差が大きくない Q1, Q13, Q16 について検討する．なお差が大きくも小さくもないとした Q15 は，指定する経路が一つの問合せであり，単純な (述語を有さない) XPath 式による問合せの能力の差が十数倍であることが分かる．

Q2 は順序節を用いた問合せであり，順序節を有する問合せは XRel にとって有利にはたらくことが分かる．この理由は XRel では要素表に順序に関するデータ (index, reindex) も格納しており，このデータを有効に利用したことによると考えられる．

Q6 は「//」を用いて経路を指定した問合せであり，指定する経路中に「//」を有した問合せは XRel にとって有利にはたらくことが分かる．この理由は XRel では指定する経路表現の検索が単なる文字列照合であり,「//」はワイルドカード「%」により実現されており，指定する経路検索の計算コストが低くなっているからと考えられる．

Q1, Q13, Q16 は指定する経路が 3 つの問合せであり，指定する経路が増えることは XRel にとって不利にはたらくことが分かる．この理由は指定する経路が 1 つである Q15 に比べて表同士の結合の演算が必要となり，この計算コストが高いからと考えられる．

[4] ⟨URL: http://www.fatdog.com/⟩

[5] 規模変更因子を 0.01 とした理由は，XQEngine で処理可能な要素数が 32768 までという制約があったため．

[6] 問 4 については，規模変更因子を 0.01 にしたことで文字列一致すべきデータが表れなかったので文字列を変更した．

7 まとめと今後の課題

7.1 まとめ

本論文では関係データベースを用いた XML 文書の格納および検索手法である XRel の実装とその評価について述べた．

XRel の実装については， XPath 式から SQL 問合せへの変換モジュールについて述べ，その応用としてのシェイクスピアの戯曲の検索システムを紹介した．XPath 式から SQL 問合せへの変換モジュールを実現することにより，利用者から与えられた問合せ式をシステム側で自動的に SQL 式に変換し，関係データベースへの問合せをすることができる．シェイクスピアの戯曲の検索システムを実現することにより， XML 文書の検索システムとして汎用的に XRel を用いることができる可能性を示した．

さらに， XRel の XML データベースとしての有効性を示すために， Xmark により性能評価の実験を行った．実験の結果，指定する経路の増加にともない表同士の結合回数が増加するため検索効率が悪化するものの，他の XML データベース検索システムである XQEngine と比較するときわめて高速に検索を行うことが示せた． これより XRel は XQEngine に対して XML データベースとして性能が高いことが分かった．

7.2 今後の課題

7.2.1 XQuery 式から SQL 問合せへの変換手法の確立

SQL 式を駆使することにより XML 文書に対し多種多様な問合せが可能である． 実際，本論文では Xmark の XQuery による問合せを手作業により SQL 問合せに変換した． しかし XRel がより汎用的に XML データベースシステムとして用いられるには，XQuery 式を SQL 問合せに自動的に変換する手法を開発する必要がある．

7.2.2 XML 文書の更新

XRel では XML 文書の各要素，属性，文字列の文書内での位置を「ファイルの先頭から何バイト目か」という情報 (リージョン) を用いて格納している． リージョンは XML 文書中での節点の開始位置と終了位置からなる整数値の組であり，文書の更新に対してはあまりよくない [10]．すなわちリージョンの値が正確な節点の位置を記録しているため，更新によってバイト位置がずれてしまうと多くのリージョン値を修正しなくてはならず，このことは，更新が頻繁に起こるシステムでは不利である．そこで文書内容の更新があった場合でも，更新箇所を最小にとどめることのできる手法の開発が必要である． この解決手法として Kha 等は [10] 親節点の開始位置からの相対距離により表される可変リージョンを提案している．

謝辞 本研究の一部は，文部科学省科学技術研究費基盤研究 (課題番号: 11480088, 12680417, 12780309) ，ならびに科学技術振興事業団 (JST) の戦略的基礎研究推進事業 (CREST)「高度メディア社会の生活情報技術」プログラムの支援によるものである．ここに記して感謝を表す．

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*, Vol. 1, No. 1, pp. 110–141, 2001.

[2] 吉川正俊, 志村壮是, 植村俊亮．オブジェクト関係データベースを用いた XML 文書の格納と検索. 情報処理学会論文誌, Vol. 40, No. SIG 6(TOD 3), pp. 115–131, 1999.

[3] G. E. Blake, M. P. Consens, I. J. Davis, P. Kilpeläinen, E. Kuikka, P. A. Larson, T. Snider, and F. W. Tompa. Text / Relational Database Management System: Overview and Proposed SQL Extensions. Technical report, Technical Report CS-95-25, UW Centre for the New OED and Text Research, Department of Computer Science, University of Waterloo, June 1995.

[4] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. J. DeWitt, and J. F. Naughton. Relattional Database for Querying XML Documents: Limitations and Opportunities. *Proceedings of 25th International Conference on Very Large Data Bases*, pp. 302–314, September 1999.

[5] J. Zhang. Application of OODB and SGML Techniques in Text Database: An Electronic Dictionary System. *SIGMOD Record*, Vol. 24, No. 1, pp. 3–8, 3 1995.

[6] D. Florescu and D. Kossmann. Storing and Querying XML Data using an RDBMS. *IEEE Data Engineering Bulletin*, Vol. 22, No. 3, pp. 27–34, September 1999.

[7] A.R. Schmidt, F.Waas, M.L. Kersten, D. Florescu, I. Manolescu, M.J. Carey, I. Manolescu, and R. Busse. Why And How To Benchmark XML Database. *ACM SIGMOD Record*, Vol. 30, No. 3, pp. 27–32, September 2001.

[8] A.R. Schmidt, F.Waas, M.L. Kersten, D. Florescu, I. Manolescu, M.J. Carey, and R. Busse. The XML Benchmark Project. Technical report, Centrum voor Wiskunde en Infomatica, April 2001.

[9] T. Böhme and E. Rahm. XMach-1: A Benchmark for XML Data Management. Technical report, In BTW 2001, 2001.

[10] D. D. Kha, M. Yoshikawa, and S. Uemura. An Efficient Storage for XML Data using Relative Region Coordinate. *Proc. of IEEE 17th International Conference on Data Engineering*, pp. 313–320, 2001.

A Xmark で用意された問合せ式

Xmark で用意された 20 の問合せのうち本研究で行った 15 問合せは次のとおりである．なお Q3, Q4 については用意された XQuery 式に誤りが含まれているため，訂正後の式を掲載する．

Q1: Return the name of the person with ID 'person0'.

```
 FOR $b IN document("auction.xml")/site/people/per
son/[@id="person0"]
 RETURN $b/name/text()
```

Q2: Return the initial increases of all open auctions.

```
 FOR $b IN document("auction.xml")/site/open_aucti
ons/open_auction
 RETURN <increase> $b/bidder[1]/increase/text() </
increase>
```

Q3: Return the IDs of all open auctions whose current increase is at least twice as high as the initial increase.

```
 FOR $b IN document("auction.xml")/site/open_aucti
ons/open_auction
 WHERE  $b/bidder[1]/increase/text() * 2 <=
        $b/bidder[last()]/increase/text()
 RETURN <increase first=$b/bidder[0]/increase/text
() last=$b/bidder[last()]/increase/text()/>
```

Q4: List the reserves of those open auctions where a certain person issued a bid before another person.

```
 FOR $b IN document("auction.xml")/site/open_aucti
ons/open_auction
 WHERE  $b/bidder/personref[@person="person18829"]
 BEFORE $b/bidder/personref[@person="person10487"]
 RETURN <history> $b/reserve/text() </history>
```

Q6: How many items are listed on all continents?

```
 FOR $b IN document("auction.xml")/site/regions
 RETURN COUNT ($b//item)
```

Q7: How many pieces of prose are in our database?

```
 FOR $p IN document("auction.xml")/site
 RETURN count($p//description) + count($p//annotat
ion) + count($p//email)
```

Q8: List the names of persons and the number of items they bought. (joins person, closed_auction)

```
 FOR $p IN document("auction.xml")/site/people/per
son
 LET    $a := FOR $t
 IN document("auction.xml")/site/closed_auctions/c
losed_auction
   WHERE $t/buyer/@person = $p/@id
   RETURN $t
 RETURN <item person=$p/name/text()> COUNT ($a) </
item>
```

Q9: List the names of persons and the names of the items they bought in Europe. (joins person, closed_auction, item)

```
 FOR $p IN document("auction.xml")/site/people/per
son
 LET $a := FOR $t
   IN document("auction.xml")/site/closed_auctions
/closed_auction
   LET $n := FOR $t2
     IN document("auction.xml")/site/regions/europ
e/item
     WHERE $t/itemref/@item = $t2/@id
     RETURN $t2
   WHERE $p/@id = $t/buyer/@person
   RETURN <item> $n/name/text() </item>
 RETURN <person name=$p/name/text()> $a </person>
```

Q13: List the names of items registered in Australia along with their descriptions.

```
 FOR $i IN document("auction.xml")/site/regions/au
stralia/item
 RETURN <item name=$i/name/text()> $i/description
</item>
```

Q14: Return the names of all items whose description contains the word 'gold'.

```
 FOR $i IN document("auction.xml")/site//item
 WHERE CONTAINS ($i/description,"gold")
 RETURN $i/name/text()
```

Q15: Print the keywords in emphasis in annotations of closed auctions.

```
 FOR $a IN document("auction.xml")/site/closed_auc
tions/closed_auction/annotation/description/parlis
t/listitem/parlist/listitem/text/emph/keyword/text
()
 RETURN <text> $a </text>
```

Q16: Return the IDs of those auctions that have one or more keywords in emphasis.

```
 FOR $a IN document("auction.xml")/site/closed_auc
tions/closed_auction
 WHERE NOT EMPTY ($a/annotation/description/parlis
t/listitem/text/emph/keyword/text())
 RETURN <person id=$a/seller/@person />
```

Q17: Which persons don't have a homepage?

```
 FOR $p IN document("auction.xml")/site/people/per
son
 WHERE  EMPTY($p/homepage/text())
 RETURN <person name=$p/name/text()/>
```

Q18: Convert the currency of the reserve of all open auctions to another currency.

```
  FUNCTION CONVERT ($v)
  {
    RETURN 2.20371 * $v -- convert Dfl to Euro
  }

  FOR $i IN document("auction.xml")/site/open_auct
ions/open_auction/
  RETURN CONVERT($i/reserve/text())
```

Q19: Give an alphabetically ordered list of all items along with their location.

```
  FOR $b IN document("auction.xml")/site/regions//
item
  LET $k := $b/name/text()
  RETURN <item name=$k> $b/location/text() </item>
  SORTBY (.)
```