異種情報源統合のための XML 問合せ最適化と情報源問合せ能力管理

林 孝志[†,☆] 小西一也[†,☆☆] 堀口恭太郎[†]
綱川光明[†] 鈴木源吾[††] 芳西崇[†]

本稿では，異種情報源に対して定義された XML ビューを通じて問合せを行う際の問題を取り扱う．本稿の寄与は主に 2 点ある．1 点目は XML を操作するための代数を採用し，代数の演算間に交換則を適用することで XML ビューに対する問合せを最適化する手法である．本稿では，交換則を適用し，選択演算を情報源側に行わせることで，処理速度の向上を図っている．2 点目は異種情報源を統合するための問合せ能力管理方法である．本稿では，各情報源の問合せ能力を代数の演算列として管理し，XML ビュー問合せから各情報源への問合せを生成する方法を示している．本稿では，関係データベースおよび XPath データベースに格納されている情報の検索について，提案手法の検証を行った．

XML Query Optimization and Wrapping Query Language for Heterogeneous Information Integration

Takashi Hayashi,[†,☆] Kazuya Konishi,[†,☆☆] Kyotaro Horiguchi,[†]
Mitsuaki Tsunakawa,[†] Gengo Suzuki[††] and Takashi Honishi[†]

In this paper, we address the problem of evaluating XML queries over XML views of heterogeneous information sources. This paper makes two main contributions. The first is a technique that optimizes XML queries by applying a commutative law. This technique pushes queries down to the information sources. The second is wrapping the query languages in order to integrate the heterogeneity of the information sources. Query capabilities are described as sequences of internal query representations. We implement a prototype system and verify that the proposed methods work with relational databases and XPath databases.

1. はじめに

近年，インターネット・イントラネットの発展により，大量の情報にアクセスすることが可能となった．これらの情報源は，Web ページ，データベースなど様々である．これら異種情報源を統合検索するアプリケーションへのニーズは，今後ますます増えると予想される．そのようなアプリケーションに対しミドルウェアのデータモデルとして XML を利用することに興味が集まっている．XML は柔軟な形式であり，構造データ/半構造データの両方の内部表現として利用可能だからである．

しかし，XML データモデルを大量・異種情報源の統合検索に利用するためには，克服しなければならない問題がある．第 1 に，大量の情報源を効率的に検索するための XML 問合せ最適化手法が確立されていない点があげられる．第 2 に，異種情報源を統合するために，各情報源の問合せ能力を管理しなければならないという点があげられる．これらの問題点を解決するために，本稿では異種情報源に対して定義された XML ビューを通じて問合せを評価する方法を提案する．

以下，本稿の構成を示す．2 章では，関連研究とその問題点について議論する．3 章では，情報統合のための XML に基づくアプローチを提案する．4 章では，関係データベースおよび XPath データベース（XPath[1)] を問合せ言語として利用しているデータベース）に格納されている情報の検索について，提案手法を検証し，5 章でまとめを行う．

† 日本電信電話株式会社 NTT サイバースペース研究所
NTT Cyber Space Laboratories, NTT Corporation
†† 日本電信電話株式会社第二部門情報推進担当
Information Strategy Planning Section Department II, NTT Corporation
☆ 現在，NTT ビズリンク株式会社
Presently with NTT Bizlink Corporation
☆☆ 現在，株式会社 NTT データ
Presently with NTT DATA Corporation

2. 関連研究

複数の情報源を統合する研究として，マルチデータベース・連邦データベース技術と呼ばれる分野が研究されてきた[2]．この分野では，分散環境下での問合せ最適化や更新問題など様々な研究課題が議論されてきた．そして，近年のインターネットの急速な普及による情報源の多様化という観点からメディエータシステム[3]に注目が集まっている．メディエータシステムでは，各情報源はラッパー（wrapper）と呼ばれるソフトウェアモジュールによって共通のデータモデルに変換される．メディエータは，ラッパー経由で提供される情報を統合するソフトウェアモジュールである．Information Manifold[4]，TSIMMIS[5]などが代表的なメディエータシステムである．いくつかのメディエータシステムでは，データモデルの多様性に対応するために，共通のデータモデルとして XML を採用している．MIX[6]，SilkRoute[7]や MediPresto/XM[8]などがあげられる．これらの XML メディエータシステムはアプリケーションから入力された XML 問合せを解析し，情報源の問合せへと変換し，発行する．そして，情報源から返却された検索結果は XML として統合される．

前章では，XML データモデルを利用して大量・異種情報源にアクセスする際に生じる 2 つの問題点について指摘した．以下では，関連研究をふまえて，これらの問題について議論する．

(A) XML 問合せ最適化に確立された手法がない

問合せ最適化とは，アクセスコストを最小にするための内部表現（代数を構成する演算に相当）の評価順序を求めることである．したがって，問合せ最適化の前に，問合せを解析し，システムに適した内部表現に変換する必要がある．XPERANTO[9),10)]では XML Query Graph Model（XQGM）を内部表現として採用している．しかし，その問合せ最適化手法は，一般的な XML 問合せにとって十分ではない．中間的に生成される結果を抑制し，総検索時間を削減するため，XPERANTO では選択演算を情報源側に行わせ，その処理能力を利用している．この問合せ最適化手法は，リレーショナルモデルでは典型的な手法である[11)]．しかし，XML ビューに対する問合せは検索結果の構造変換をともなうのが一般的である．もし，構造変換をともなう問合せにおいて，選択演算を情報源側に行わせると，必要な情報が欠損する可能性がある（**図 1**）．この問題点を解決するため，本稿では演算間の交換則をふまえた最適化機能を提案する．

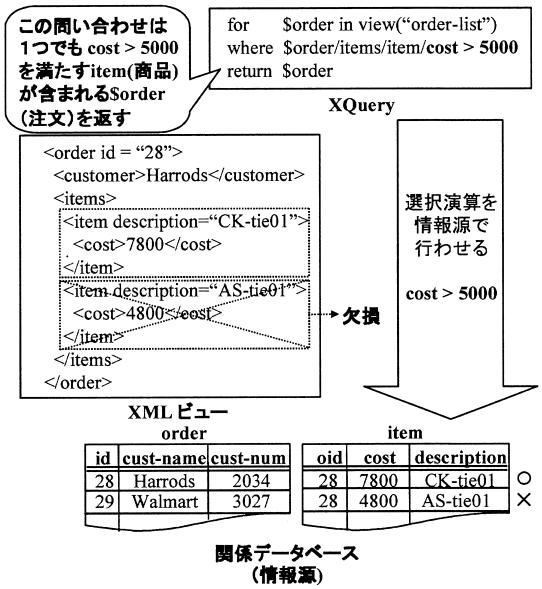

図 1 最適化による情報欠損の例
Fig. 1 A case lost through XML query optimization.

(B) 異種情報源の問合せ能力の管理が困難

異種情報源では，問合せ能力自体も異なる場合がある．たとえば，SQL は表の検索を対象としているが，巡航的（navigational）データ操作を表現するのに適していない．一方，XPath は，XML 中の特定の部分を巡航的に指定する方法であるが，結合などの汎用的な問合せ操作が不足している．したがって，情報源側の処理能力を利用しつつ，異種情報源を統合するためには，各情報源の問合せ能力を管理する必要がある．この問題に関しては，いくつかの解決策が提案されている．Garlic[12)]では，情報源の問合せ能力は対応するラッパー中に記述されている．この方法では，問合せの，どの部分が処理可能かを知るためにメディエータはラッパーと通信する必要があるため，処理の高速化に適さない．YAT[13)]では，各情報源側で評価可能な演算とメディエータの内部表現とを対応させることで，問合せ能力を管理している．しかし，内部表現の 1 つの演算が情報源問合せの 1 つの句に必ずしも対応するとは限らない．この問題を解決するため，本稿では，内部表現の演算列と情報源側の問合せのパターンとを対応させた問合せ能力管理機能を提案する．

XML データモデルを大量・異種情報源の統合検索に利用するためには，これらの問題点の克服は非常に重要である．次の章では，XML に基づく情報統合のためのアプローチを提案する．

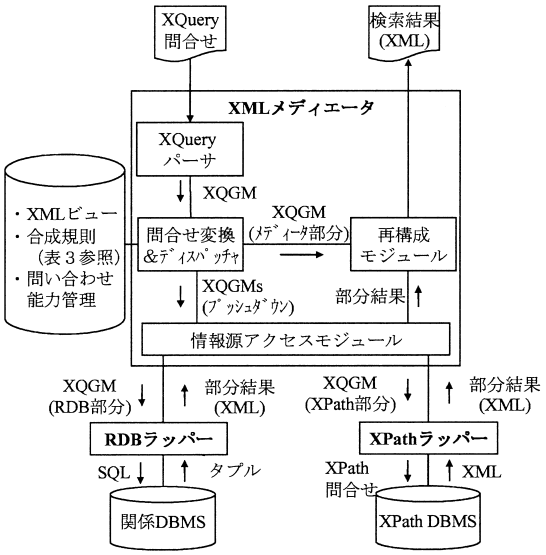

図 2 案システム（XML メディエータシステム）
Fig. 2 Proposed XML Mediator System.

order

id	cust-name	cust-num
28	Harrods	2034
29	Walmart	3027

item

oid	cost	description
28	7800	CK-tie01
28	4800	AS-tie01

データベース名：order_data

```
<order_data>
 <order>
  <row><id>28</id><cust-name>Harrods</cust-name>
  <cust-num>2034</cust-num></row>
  <row><id>29</id> <cust-name>Walmart</cust-name>
  <cust-num>3027</cust-num></row>
   ・・・・
 </order>
 <item>
  <row><oid>28</oid> <cost>7800</cost>
  <description>CK-tie01</description> </row>
   ・・・・
 </item>
</order_data>
```

図 3 関係データベースとそのデフォルト XML ビュー
Fig. 3 Relational database and its default XML view.

3. 提案手法

本稿では，異種情報源の統合のために XML メディエータシステムを採用する（**図 2**）．また，問合せ処理方式は，異種情報源の統合検索を考慮して，XPERANTO の方式を拡張して用いる．提案システムのそれぞれのラッパーは，情報源のデータ構造を直接表現したデフォルト XML ビューを生成する．**図 3** はリレーショナルデータベースとそのデフォルト XML ビューの一例を示している．システムの利用者

```
<order id = “28”>
  <customer>Harrods</customer>
  <items>
    <item description=“CK-tie01”>
      <cost>7800</cost>
    </item>
    <item description=“AS-tie01”>
      <cost>4800</cost>
    </item>
  </items>
</order>
<order id = “29”>
  ・・・・
</order>
```

(a) XML形式の注文リスト

```
1  create view order-list as (
2    for $order in view("order_data")/order/row
3    return
4      <order id = "$order/id">
5        <customer>$order/cust-name</customer>
6        <items>
7          for $item in view("order-data")/item/row
8          where $order/id = $item/oid
9          return
10           <item description = "$item/description">
11             <cost>$item/cost</cost>
12           </item>
13       </items>
14     </order> )
```

(b) ビュー定義XQuery

図 4 ユーザ定義 XML ビュー
Fig. 4 User defined XML view.

がデータベース中のデータを XML 形式の注文リストとして取り出したいと仮定する（**図 4** (a)）．デフォルト XML ビューを望みの XML 形式へと変換するために，XQuery [14] を用いてユーザ定義ビューが生成される（図 4 (b)）．

いったん，ビューが生成されると，XQuery 問合せを対応する情報源に対して発行することが可能となる．XML ビューに対する XQuery 問合せ処理は以下の 3 つの処理から構成される．(1) 問合せ解析，(2) 問合せ変換，(3) 問合せ実行である．処理 (1) と (2) をまとめて問合せ計画と呼ぶ．2 章で述べた問題点を解決するため，以下では，問合せ計画における問合せ最適化（3.1 節）と情報源問合せの能力管理（3.2 節）を中心に提案手法を説明する．

3.1 問合せ最適化

2 章で述べたように，XML 問合せ最適化のためには，問合せの内部表現が必要である．我々は XML Query Graph Model（XQGM）[9] を採用し，異種情報源に対応するため，より一般的な XML 代数へと拡張す

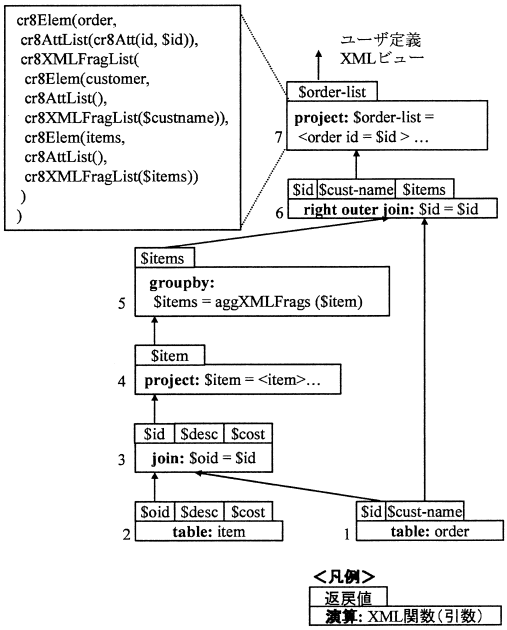

図 5 ビュー定義 XQGM
Fig. 5 View definition XQGM.

表 1 XQGM の演算
Table 1 XQGM operators.

演算	内容
table	関係DBのテーブルを表す
project	入力に基づいた結果の計算
select	入力を選択
(inner,outer) join	２つ以上の入力を結合
groupby	集約関数，グルーピング
orderby	属性値によりソート
union	２つ以上の入力の和集合
unnest	入力の入れ子構造を平坦化
view	ビューを表す
function	XQuery関数を表す

表 2 XML 関数
Table 2 XML functions.

	XML関数	内容
1	cr8Elem (Tag, Atts, Clist)	要素生成
2	cr8AttList (A_1,・・・,An)	属性リスト生成
3	cr8Att (Name, Val)	属性生成
4	cr8XMLFragList (C_1,・・・,Cn)	XMLフラグメントリスト生成
5	aggXMLFrags (C)	集約関数
6	getTagName (Elem)	要素名を返却
7	getAttributes (Elem)	属性リストを返却
8	getContents (Elem)	要素内容を返却
9	getAttName (Att)	属性名を返却
10	getAttValue (Att)	属性値を返却
11	isElement (E)	Eが要素であれば真を返却
12	isText (T)	Tがテキストであれば真を返却
13	unnest (List)	平坦化する

る．2 章で述べた問題点 (A) を解決するため，我々は XQGM の演算間に交換則を適用する．以下では，図を用いてこの拡張点を説明する．

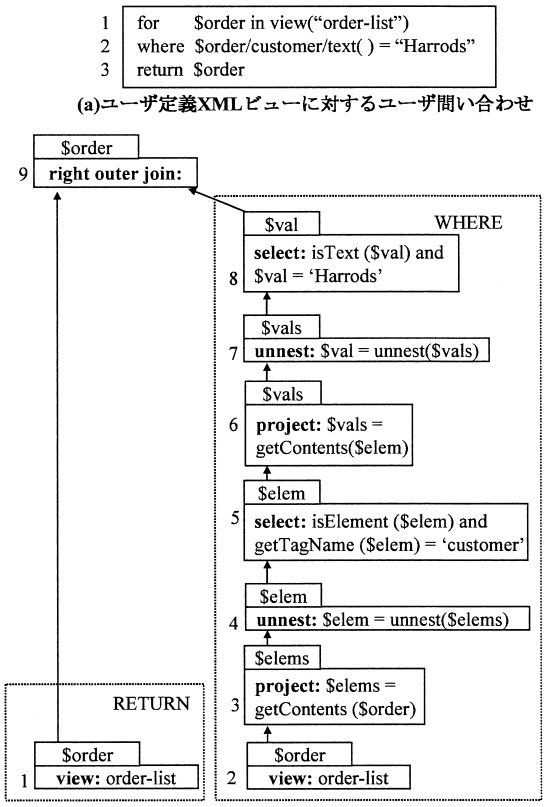

図 6 ユーザ問合せの XQuery と XQGM
Fig. 6 XQuery and XQGM of user query.

(1) 問合せ解析

図 4 (b) で与えられたビュー定義用の XQuery は**図 5** の XQGM のように解析される．解析された XQGM（ビュー定義 XQGM）は**表 1** に示す演算群で定義される．図 5 中の 7 番はその入力にタグ付けすることで XML 要素を生成する．このように XML オブジェクトの生成および操作は**表 2** に示す XML 関数により実行される．また，XPERANTO では，相関関係にある副問合せからの入力を結合する演算として correlated join [10] を定義している．本システムでは，問合せ変換処理の見通しを良くするため，ビュー合成の前に相関関係の分離操作を行う[15]．このため，6 番に示すように correlated join は外結合として表現される．

図 6 (a) は「Harrods」という名称の顧客によるすべての注文を取り出す問合せを表している．図 6 (b) は，図 5 で説明したのと同様の解析により得られたユーザ問合せ XQGM を示している．

(2) 問合せ変換

(i) ビュー合成

ビュー定義 XQGM の XML 生成用関数群（表 2 の

表 3 合成規則
Table 3 Composition rules.

規則	関数	合成対象	合成後
1	getTagName	cr8Elem (Tag, Atts, Clist)	Tag
2	getAttributes	cr8Elem (Tag, Atts, Clist)	Atts
3	getContents	cr8Elem (Tag, Atts, Clist)	Clist
4	getAttName	cr8Att (Name, Val)	Name
5	getAttValue	cr8Att (Name, Val)	Val
6	isElement	cr8Elem (Tag, Atts, Clist)	True
7	isElement	Other than cr8Elem	False
8	isText	PCDATA	True
9	isText	Other than PCDATA	False
10	unnest	aggXMLFrags (C)	C
11	unnest	cr8XMLFragList (C_1,・・・,Cn)	C_1 U ・・・U Cn
12	unnest	cr8AttList (A_1,・・・,An)	A_1 U・・・U An

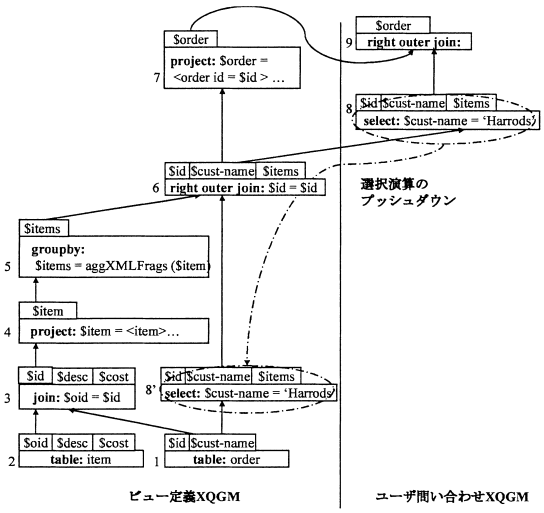

図 7 ビュー合成後の XQGM
Fig. 7 XQGM after view composition.

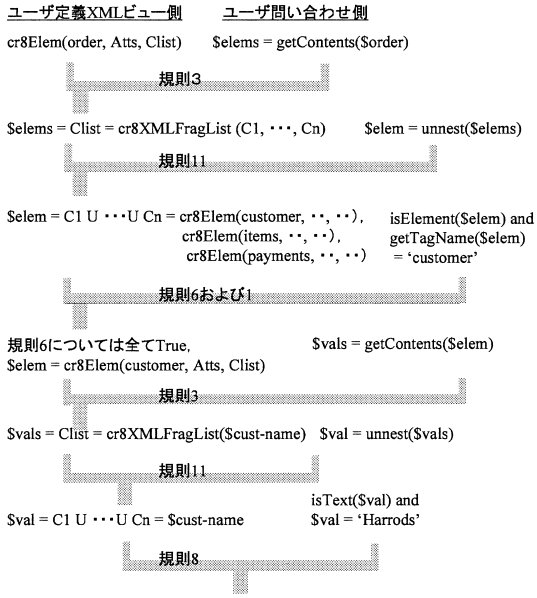

図 8 合成規則の適用例
Fig. 8 Example of applying composition rules.

表 4 選択演算とその他の XQGM 演算との間の交換規則
Table 4 Commutative law between select and other.

演算	交換則の成立/不成立
table	×
project	○
(inner,outer) join	△
groupby	△
orderby	○
union	○
unnest	○
view	○
function	△

○・・・成立, ×・・・不成立, △・・・条件により成立

項番 1 から 5 の XML 関数）とユーザ問合せ XQGM の XML 操作用関数群（表 2 の項番 6 から 13）の合成を行い，検索結果に出現しない XML オブジェクトの生成コストを削減することがビュー合成の目的である．

表 3 は操作用関数群の削除に使われる合成規則を定義している．図 5 に示したビュー定義 XQGM と図 6 (b) に示したユーザ問合せ XQGM との合成結果を**図 7** に示す．表 3 の合成規則に基づいて，図 6 (b) の操作用関数群（3 番〜8 番）が図 5 中の 7 番の対応する関数と合成される様子を**図 8** に示す．ビュー合成の結果，図 5 の 6 番の返戻値（$id，$cust-name，$items）に対し，図 6 (b) の 8 番の選択演算を行う（$cust-name = 'Harrods'）手順となる（図 7 参照）．

(ii) 選択演算プッシュダウン

情報源に処理を任せ，メディエータ側の負荷を軽減するため，情報源側で処理可能な演算を XQGM の下側へ移動（プッシュダウン）する．処理が可能かの判断に関しては次節で述べる．本稿では，XQGM 演算間の交換則に基づいてプッシュダウンを行う．以下では，検索時間を削減するため，選択演算のプッシュダウンに注目する．選択演算とその他の XQGM 演算との間の交換則を**表 4** に示す．ビュー合成の結果，選択演算（図 7 の 8 番）は，右外結合（6 番）の上に位置している．選択演算と右外結合の間には，以下のように条件付きで交換則が成り立つ（表 4 の 3 行目）．

交換則： join(A,B)∘select(A)=select(A)∘join(A,B)
条件： select(A) は A についての選択演算

したがって，説明してきた例では，選択演算は，右外結合の右側の table 演算（1 番）のすぐ上まで問題なくプッシュダウンできる．

次に図 6 (a) 2 行目が “where $order/items/item/cost > 5000” という新しい例を考えてみる．この新しい例では，注文の中に 1 つでも条件を満たす商品

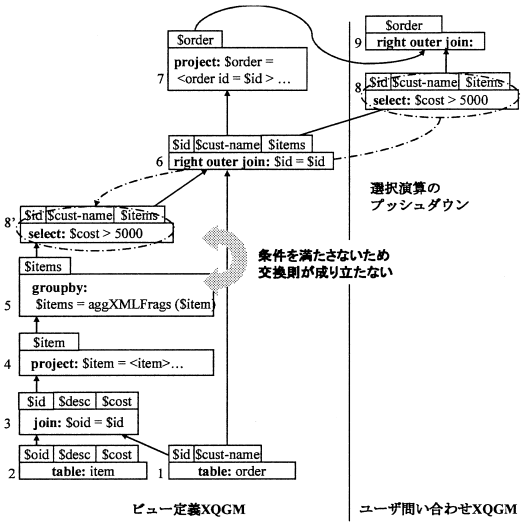

図 9 ビュー合成後の XQGM
Fig. 9 XQGM after view composition.

が含まれれば，述部は真を返す．これは，XQuery の where 句が暗黙的に存在限量だからである（図 1 参照）．ユーザ問合せ XQGM との合成結果を**図 9** に示す．選択演算（8 番）と集約演算（5 番）の間には，以下のように条件付きで交換則が成り立つ（表 4 の 4 行目）．

交換則：groupby ∘ select = select ∘ groupby
条件：select の作用域（scope）が変化しない

この例では，集約演算により作用域が変化している．集約演算の前では，作用域が item 単位なのに対し，集約演算の後では，作用域が items 単位（同一の注文番号で商品を集約）に変化している．もし，選択（$cost > 5000）を table 演算（2 番）の上までプッシュダウンすると，5,000 円以上の商品と一緒に注文された 5,000 円未満の商品まで情報源側の処理で失われてしまう（図 1 参照）．したがって，この例では選択演算（8 番）は集約演算（5 番）の上までしかプッシュダウンできない（図 9 参照）．

このように，演算間の交換則を適用することにより，プッシュダウンによる情報の欠損を防ぎつつ，中間的に生成される結果を抑制することが可能となる．

3.2 情報源問合せの能力管理

3.1 節で示したように XQGM は演算と XML 関数の集合からなる．本稿では，各情報源側で評価可能な演算を用いて情報源問合せの能力を記述する．さらに，提案システムでは，演算列とクエリテンプレートとを対応付ける．1 つの演算が必ずしも，情報源問合せ言語の 1 つの句に変換できるとは限らないからである．

図 10 は，選択演算のプッシュダウンが不可能な場合の，演算列とクエリテンプレートの対応例である．図 10 の XQGM は，SQL の副問合せに変換される．

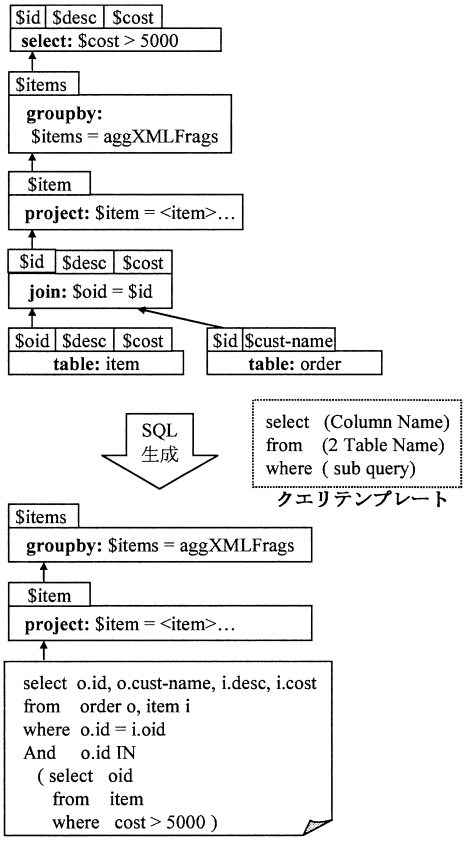

図 10 演算例とクエリテンプレート
Fig. 10 Sequence of operators and query template.

XPath データベースに対する問合せ能力管理を説明するため，他の例を示す．eXcelon [16] のような XPath データベースは，直接，XML ファイルを格納できる．**図 11** (a) に格納された XML ファイル（XML ビュー）を示す．図 11 (b) は，赤を含むすべての商品を抽出する XQuery 問合せと，データベースに対して発行される 2 つの XPath 問合せを示す．XPath では結合演算が評価できないので，メディエータが 2 つの返却された XML オブジェクトを再構成する．

このように情報源の問合せ言語により評価可能な演算の列とクエリテンプレートを対応付けることで，情報源に発行する問合せを生成する．可能な限り演算を任せることで，メディエータ側の負荷軽減が可能となる．

4. 実　　験

提案手法の検証を行うため，MediPresto/XM [8] 上にプロトタイプシステムを構築した．実装には，Java

```
<brand-list>
  <brand>
    <brand-name>Andrew Soccer</brand-name>
    <item><item-name>AS-shirt01</item-name>
     <design>
      <color>red</color>
      <color>blue</color>
      <pattern>stripe</pattern>
     </design>
    </item>
     ・・・・
  </brand>
     ・・・・
</brand-list>
```

(a) XPathデータベースに格納されたXMLファイル

```
1  for $item in view("brand-list")/brand/item
2   where $item/design/color = "red"
3   return
4    <red-item>
5     <brand-name>$item/../brand-name</brand-name>
6     <item-name>$item/item-name</item-name>
7    </red-item>
```

XPath 生成

/axis :: node test [predicate]/

クエリテンプレート

```
/brand-list/brand/item[design/color = "red"]/../brand-name
/brand-list/brand/item[design/color = "red" ]/item-name
```

(b) XQueryとデータベースに発行されたXPath

図 11 XML ファイル，XQuery と発行された XPath
Fig. 11 XML file, XQuery and issued XPath.

と JDK1.2 を利用している．

3.1 節で述べた問合せ最適化，特に，選択演算のプッシュダウンに関して，検索速度の点から性能を評価するため，以下の 3 つの実験を行った．以下で，選択度とは検索対象のデータ量に対し，選択条件を満たすデータ量の割合をさす．

実験 1：選択度 10%でのプッシュダウンを行う場合と行わない場合の比較実験

実験 2：選択度 N/5000 ($1 < N < 5000$) でプッシュダウンを行う場合と行わない場合の比較実験

実験 3：選択度 10%での，総検索時間の内訳

また，3.2 節で述べた演算列とクエリテンプレートの対応による，情報源問合せの能力管理の実行性を確認するため，関係データベースと XPath データベースに対するラッパーを実装し，上記実験に用いた．実験 2 の N および 5000 の単位は，濃度（関係データベース）あるいはノード数（XPath データベース）である．関係データに対する XML ビューを図 4 に示したのと同様に定義した．また，図 6 (a) に示した問合せを用いた．XPath データベースへは図 11 (a) に示したのと同様に XML ファイルを蓄積し，図 11 (b) の問合せを用いた．

システムでは，総検索時間に対し 3 つの要因が影響している（図 2 参照）．第 1 は，情報源問合せ（SQL や XPath）生成時間．これは XQuery 問合せの解析とデータベース管理システムへの接続時間から構成される．第 2 は，部分的な XML 結果を抽出する時間．これは情報源問合せを実行する時間と対応するラッパーにより結果を XML へ変換する時間からなる．第 3 は，メディエータ内での XML 検索結果の再構成時間である．

実験では，メディエータシステムを PentiumIII, 760 MHz，メインメモリ 256 MB，Windows2000 上に実装して検索を実行した．また，関係データベースと XPath データベースとして，それぞれ SQLServer7.0 と XIS3.0（eXcelon）[14] を利用した．これらのデータベースシステムは PentiumII，256 MHz，メインメモリ 256 MB，WindowsNT4.0 の別々のマシンで実行した．

関係データベースに対する，実験 1，2，3 の結果をそれぞれ**図 12**，**図 13**，**図 14** に示す．図 12 から，選択度 10%では，プッシュダウンにより検索が 2 倍高速になっていることが分かる．図 13 は選択度 1%で，検索時間が 71 倍高速になっていることを示している．図 14 はメディエータ内の検索結果再構築時間が総検索時間の大部分を占めていることと，再構築時間は返却されるデータ量に依存していることを示している．

XPath データベースに対する，実験 1，2，3 の結果をそれぞれ**図 15**，**図 16**，**図 17** に示す．図 15 から関係データベース同様，選択度 10%では，プッシュダウンにより検索が 2 倍高速になっていることが分かる．図 16 は選択度 1%で，検索時間が 19 倍高速になっていることを示している．選択演算のプッシュダウンが関係データベースにおいて，より効果的なのは，SQL が結合のような，負荷の高い演算を評価可能だからである．このように，関係データベースと XPath データベースでは，評価可能な演算が異なる．評価可能な XQGM の演算列とクエリテンプレートとの対応により，適切な SQL と XPath が生成され，それぞれの情報源へ発行される．このように，情報源側の問合せ処理能力を利用することで総検索時間を高速化できる．

これらの結果から以下の結論が導かれる．

(1) 再構成時間が総検索時間の大部分を占めている．
(2) 選択演算のプッシュダウンにより中間結果の作成が抑制されるため，総検索時間が削減される．
(3) 問合せ処理能力を有する情報源に，評価可能な処理を委譲することで，システム全体のスループットの向上が図れる．

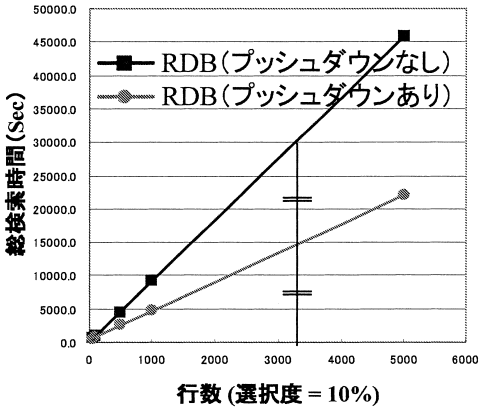

図 12 実験 1（関係データベース）
Fig. 12 Experiment 1 on relational database.

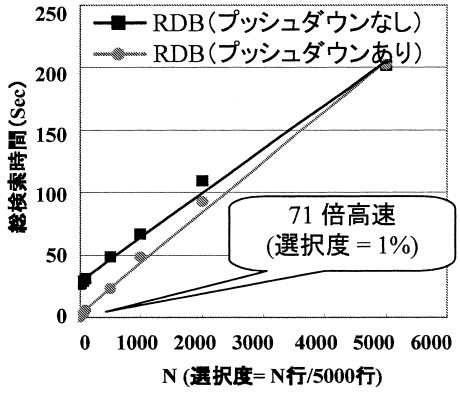

図 13 実験 2（関係データベース）
Fig. 13 Experiment 2 on relational database.

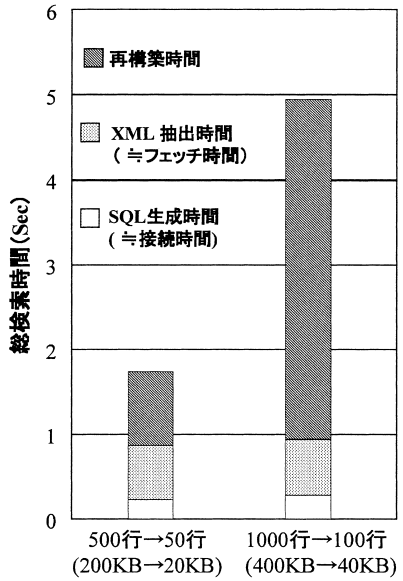

図 14 実験 3（関係データベース）
Fig. 14 Experiment 3 on relational database.

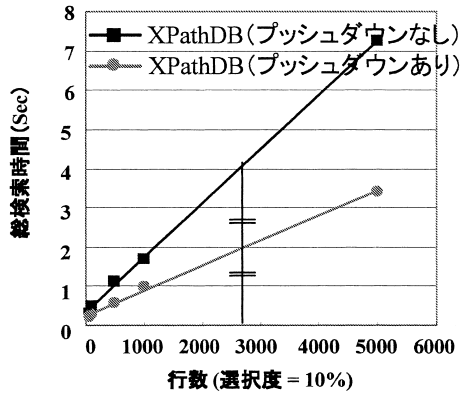

図 15 実験 1（XPath データベース）
Fig. 15 Experiment 1 on XPath database.

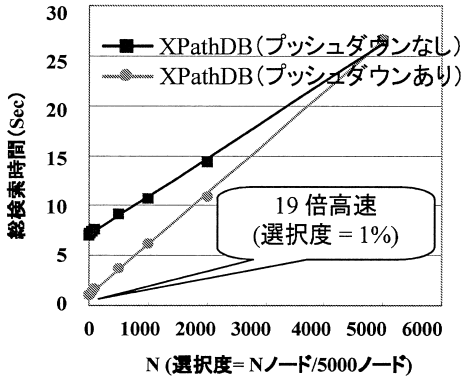

図 16 実験 2（XPath データベース）
Fig. 16 Experiment 2 on XPath database.

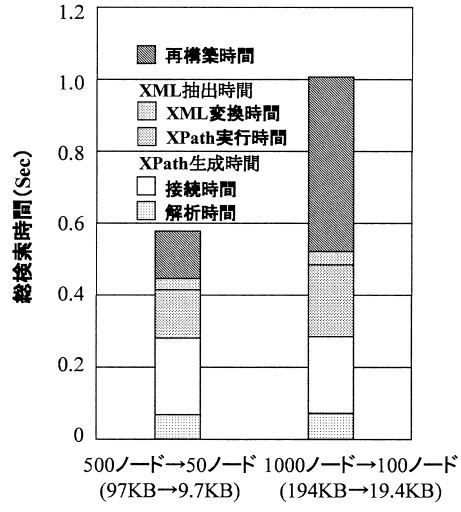

図 17 実験 3（XPath データベース）
Fig. 17 Experiment 3 on XPath database.

5. まとめと今後の課題

本稿では，異種情報源に対して定義された XML ビューを通じて問合せを行う際の 2 つの主要な問題を取り扱った．第 1 は問合せ最適化である．中間結果の抑制という観点から選択演算のプッシュダウンに注目した．本稿では，XML を操作するための代数を採用し，代数の演算間に交換則を適用することにより，必要な情報を欠損することなく演算を情報源側にプッシュダウンできる方法を提案し，実験結果から有効性を確認した．第 2 は情報源問合せの能力管理方法である．負荷分散の観点から，情報源側に発行する問合せに可能な限り処理を任せることとした．これにより，メディエータ側の負荷を軽減することができる．検索実験から，XQGM の演算列とクエリテンプレートを対応させることで，SQL および XPath 問合せを生成できることを確認した．

本稿ではこれら 2 つの問題に焦点を当てたが，我々の目標は異種情報源の統合である．まず，我々は関係データベースと XPath データベースおよび，その他の情報源の統合検索を評価する予定である．さらに，ある種の XQuery は評価するのが困難である．たとえば利用者が定義する関数である．利用者定義の関数は主に 2 つに分類される．1 つは情報源側の演算に関連するもので，もう 1 つは統合された XML ビューに関連するものである．関係データベース上の XML ビューに対する利用者定義関数の評価については文献 15) に詳しい．利用者定義関数を含めた，異種情報源の統合検索は，今後の課題である．

参 考 文 献

1) World Wide Web Consortium: XML Path Language (XPath) 2.0, W3C Working Draft (2001).
2) Sheth, A.P. and Larson, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases, *ACM Computing Survey*, Vol.22, No.3, pp.183–236 (1990).
3) Wiederhold, G.: Mediators in the architecture of future information systems, *IEEE Computer*, Vol.25, pp.38–49 (1992).
4) Levy, A., Rajaraman, A. and Ordille, J.: Querying heterogeneous information sources using source descriptions, *22nd VLDB Conf.*, Bombay, India (1996).
5) http://www-db.stanford.edu/tsimmis/tsimmis.html
6) Baru, C., Gupta, A., Ludascher, B. and Marciano, R.: XML-Based Information Mediation with MIX, *SIGMOD'99*, Philadelphia, U.S.A. (1999).
7) Fernandez, M., Tan, W. and Suciu, D.: SilkRoute: Trading Between Relations and XML, *World Wide Web Conf.*, Toronto, Canada (1999).
8) 鈴木源吾，小西一也，林 孝志，小林伸幸，芳西崇：XML に基づく異種情報源メディエーションシステム：MediPresto/XM，情報処理学会研究報告 2001-DBS-125-60 (2001).
9) Shanmugasundaram, J., et al.: Querying XML Views of Relational Data, *27th VLDB Conf.*, Roma, Italy (2001).
10) Shanmugasundaram, J., et al.: Efficiently Publishing Relational Data as XML Documents, *The VLDB Journal*, Vol.10, No.2–3, pp.133–154 (2001).
11) 滝沢 誠：データベースシステム入門技術解説，ソフト・リサーチ・センター (1992).
12) Haas, L.M., Kossmann, D., Wimmers, E.L. and Yang, J.: Optimizing Queries across Diverse Data Sources, *23rd VLDB Conf.*, Athens, Greece (1997).
13) Christophides, V., Cluet, S. and Simeon, J.: On Wrapping Query Language and Efficient XML Integration, *SIGMOD RECORD*, Vol.29, No.2 (2000).
14) World Wide Web Consortium: XQuery1.0: An XML Query Language, W3C Working Draft (2001).
15) 川田 純，石川佳治，北川博之：リレーショナルデータベース上の XML ビューに対する外部関数を考慮した問合せ処理，情報処理学会論文誌：データベース，Vol.43, No.SIG12 (TOD16), pp.16–37 (2002).
16) eXcelon Corp. http://wwwww.exceloncorp.com/

(平成 15 年 3 月 25 日受付)
(平成 15 年 7 月 6 日採録)

（担当編集委員 掛下 哲郎）

林　　孝志（正会員）

1997 年慶應義塾大学理工学部電気工学科卒業．1999 年同大学院理工学研究科電気工学専攻修士課程修了．同年，日本電信電話（株）勤務．NTT サイバースペース研究所にて XML とデータベースシステムの研究に従事した後，2003 年より NTT ビズリンク（株）ヴィジュアル・コミュニケーション事業部所属．

小西　一也（正会員）

1995 年茨城大学工学部情報工学科卒業．1997 年同大学院理工学研究科情報工学専攻修士課程修了．同年，NTT データ通信（株）（現,（株）NTT データ）入社．2000 年〜2003 年，日本電信電話（株）NTT サイバースペース研究所所属．現在,（株）NTT データ勤務．主として XML メディエータシステムの研究に従事．

堀口恭太郎（正会員）

1990 年早稲田大学理工学部機械工学科卒業．1992 年同大学院機械工学修士課程修了．現在，日本電信電話（株）NTT サイバースペース研究所所属．XML とデータベースシステムの研究に従事．

綱川　光明

1990 年筑波大学第一学群自然学類卒業．同年，日本電信電話（株）勤務．現在，NTT サイバースペース研究所所属．主に，データベース管理技術，異種情報源統合検索技術の研究に従事．現在，メタデータ流通技術の研究に従事．電子情報通信学会会員．

鈴木　源吾（正会員）

1988 年東北大学理学部数学科卒業．1990 年同大学院数学専攻修士課程修了．同年，日本電信電話（株）勤務．NTT サイバースペース研究所にて XML メディエータシステムの研究に従事した後，現在，NTT 第二部門情報推進担当所属．電子情報通信学会，ACM 各会員．

芳西　　崇（正会員）

1981 年九州工業大学工学部電気工学科卒業．1983 年東京工業大学大学院修士課程修了．同年，日本電信電話（株）勤務．マルチメディアのデータベース管理システムと検索技術の研究に従事．現在，NTT サイバースペース研究所情報ベースプロジェクト情報ベースシステム技術グループリーダー．IEEE 会員．