データベースシステム　122－20
(2000. 7. 27)

拡張可能 XML 問合せ言語 X^2QL における外部関数の評価モデル

品川 徳秀 †　北川 博之 ‡　石川 佳治 ‡

† 筑波大学 工学研究科　‡ 筑波大学 電子・情報工学系

siena@dblab.is.tsukuba.ac.jp　{kitagawa,ishikawa}@is.tsukuba.ac.jp

305-8573 茨城県つくば市天王台 1-1-1
筑波大学 電子・情報工学系 データベース研究室
TEL/FAX 0298-53-5051

あらまし　近年、XML をはじめとする構造化文書データベースの高度利用への要求が増大しており、その基本的な処理記述言語として各種の問合せ言語が提案されている。これらの多くは比較的強力な文書構造操作機能を持つものの、十分な記述内容処理機能を提供していない。しかし、構造化文書の高度利用への要求に対応し、大量の文書から目的とする情報を効率的に獲得するためには、高度な記述内容処理機能の利用が必須である。この観点から、我々は XML-QL をベースとした、ユーザ定義の外部関数による機能拡張性を持つ拡張可能 XML 問合せ言語 X^2QL を提案してきた。本稿では、X^2QL の概要と、XML-QL に対する拡張部分である外部関数の評価モデルを示す。

キーワード　XML, 問合せ言語, 機能拡張性, 外部関数, X^2QL

A Processing Model of Foreign Functions in An eXtensible XML Query Language X^2QL

Norihide Shinagawa †　Hiroyuki Kitagawa ‡　Yoshiharu Ishikawa ‡

† Doctoral Program in Engineering, University of Tsukuba　‡ Institute of Information Sciences and Electronics, University of Tsukuba

siena@dblab.is.tsukuba.ac.jp　{kitagawa,ishikawa}@is.tsukuba.ac.jp

Institute of Information Sciences and Electronics, University of Tsukuba
1-1-1 Tennohdai, Tsukuba, Ibaraki 305-8573, Japan
TEL/FAX +81-298-53-5051

Abstract　Recently, management of structured documents such as XML documents and their databases has become more and more important. Some query languages for XML have been proposed. Most of them provide very limited contents processing capability in contrast with their powerful structure manipulation capability. The contents processing capability is also important to acquire target information efficiently. From this view point, we have proposed X^2QL, which features inclusion of user-defined foreign functions to process document contents in the context of XML-QL-based document structure manipulation. In this paper, we describe an overview of X^2QL and the processing model of foreign functions.

keywords　XML, query language, extensibility, foreign function, X^2QL

1 はじめに

近年、各種文書データベースの構築やその利用の高度化が急速に進みつつある。特に、インターネットの著しい普及に伴い、XML をはじめとする構造化文書データベースの構築とその利用の高度化の要求が増大している [1]-[2]。構造化文書データベースに対する基本的な処理記述言語として、これまでに各種の問合せ言語が提案されている [3]-[11]。

XML を対象とした代表的な問合せ言語としては、XML-QL [3] や XQL [4] 等がある。また、Lorel [5]、UnQL [6]、StruQL [7]、YATL [8] 等は半構造データを対象とした問合せ言語であるが、XML 構造化文書もその適用対象として想定している。しかし、これらは多様化、複雑化する文書処理要求に対しては、以下のように必ずしも十分な機能を提供しているとは言えない。

構造化文書データベースに対する処理記述では、タグの階層構造で与えられる文書構造に対する処理と文字列テキストで与えられる記述内容に対する処理の、両者を扱う必要がある [1]。前述の問合せ言語の多くは、高水準な宣言的記述による高い問合せ記述能力を持っており、特に XML-QL や半構造データ問合せ言語の幾つかは文書構造に対する比較的強力な再構成機能を持つ。

一方で、これらは記述内容処理に関しては極めて限定された機能しか提供していない。例えば、指定した語句やパターンに合致する部分文字列を含む文書要素を選択するといったように、記述の内容に踏み込んだ処理は行なわれていない。特に、大量の構造化文書から目的とする情報を効率的に獲得するためには、より内容に踏み込んだ処理、例えば、与えられたキーワード群との意味的な類似度に基づく文書要素の絞り込みやランキング、文書要素の要約や話題抽出等の記述内容処理機能が重要となる [12]-[16]。また、構造操作に関しても、宣言的に記述する事が困難な応用処理も存在する。このような場合には、より詳細かつ具体的な記述能力が要求される。

これらの機能は必ずしも汎用的なものではなく、一般には処理の適用対象である文書要素の特性や問合せの目的に依存する。例えば、要約生成には様々な手法が提案されているが、各々の利点、欠点があり、目的に適したものを利用する必要がある [12]。また、文書間の類似度計算等についても同様である [13]。それゆえ、それらの機能を網羅的に提供する事は現実的ではない。この問題を解決するには、問合せ言語に柔軟な機能拡張性を持たせる事で、利用者がその目的に応じた必要な処理機能を導入可能にする必要がある。

我々はこのような問題に対し、構造化文書に対する問合せ言語が持つ文書構造操作機能に加え、ユーザ定義の外部関数による機能拡張性を備えた拡張可能 XML 問合せ言語 X^2QL を提案してきた [17]-[18]。X^2QL は XML-QL をベースとして拡張した問合せ言語である。ユーザは、必要に応じて外部関数を導入する事で、問合せとしての宣言的な記述が困難な処理を外部化し、外部関数呼出しを通じた拡張機能として利用可能となる。また、外部関数は文書要素に固有のメソッドとして定義可能であり、文書要素独自の処理を実現可能となっている。本稿では、X^2QL の XML-QL に対する拡張部分である外部関数に焦点を当て、その型の扱いや、文書要素との関連付け方法、処理モデルを示す。

以下では、まず 2 節で文書構造操作と高次の記述内容処理の両者を用いて実現される、構造化文書に対する問合せ例を示す。次に、3 節で X^2QL について概説し、4 節で外部関数の評価モデルについて述べる。5 節で関連研究について言及し、6 節でまとめと今後の課題を示す。

2 構造化文書に対する問合せ例

本節では、文書構造操作に加え、記述内容処理の機能を必要とする構造化文書問合せの例を示す。3 節では、本節の例を用いて X^2QL を説明する。以下では、必要に応じて〈識別子〉で “識別子” という文書要素名もしくはその文書要素を表記する。また、属性についても同様に表記する。

ここでは、ある新聞の記事データベースとして構成された XML 文書を考える。各記事は、掲載日、記事分類、見出し、本文を含み、文書の DTD は次で与えられるものとする。

```
<!-- 入力文書の DTD -->
<!ELEMENT 文書      記事+>
<!ELEMENT 記事      (掲載日, 記事分類, 見出し, 本文)>
<!ELEMENT 記事分類  #PCDATA>
<!ELEMENT 見出し    #PCDATA>
<!ELEMENT 本文      段落+>
<!ELEMENT 段落      #PCDATA>
<!ELEMENT 掲載日    (年, 月, 日)>
<!ELEMENT 年        #PCDATA>
<!ELEMENT 月        #PCDATA>
<!ELEMENT 日        #PCDATA>
```

例 1　上記の XML 文書に対し、1999 年以降に掲載された記事を記事分類に従って分け、各記事に要約を付加する。また、各記事中の記事分類は、重複するため削除する。その出力文書の DTD は以下のものとする。

```
<!-- 出力文書の DTD -->
<!ELEMENT 文書    分類+>
<!ELEMENT 分類    記事+>
<!ATTLIST 分類    分類名 CDATA REQUIRED>
<!ELEMENT 記事    (掲載日, 見出し, 本文)>
<!ELEMENT 見出し  #PCDATA>
<!-- 入力の DTD における〈本文〉以降はそのまま-->
```

これを実現するためには、〈年〉が 1999 以上の値を持つ〈記事〉の抽出、その〈記事分類〉による〈記事〉

のグルーピング、各⟨記事⟩からの⟨記事分類⟩の除去という文書構造操作と、⟨記事⟩からの⟨要約⟩の生成という記述内容処理が必要とされる。

例 2　読者が記事分類が「経済」である記事のうち、キーワード群に沿った上位 N 件以内のものを抽出したいとする。各⟨記事⟩はキーワード群との類似度によるランキングによって順位が決定される。これは、⟨記事分類⟩に基づく選択に加え、類似度の計算のために外部関数によって導入される記述内容処理を必要とする。この出力は、入力の DTD に従う。

3　X²QL における問合せ記述

本稿で提案する問合せ言語 X²QL は、XML に対する代表的な問合せ言語の一つである XML-QL [3] をベースとしている。また、問合せ中にはユーザによって外部プログラムとして与えられる外部関数が利用可能であり、これによって高い機能拡張性が要求される処理へ対応する事が可能である。3.1 節では、本稿の説明を理解する上で必要な構文について説明し、4 節で、外部関数について説明する。

3.1　問合せ記述

X²QL の基本構文は次で与えられる。

```
where      パターン式および変数束縛 [in 対象]
         [, パターン式および変数束縛 [in 対象]]*
         [, 述語]*
[rank-by   入力順基準式   top 選択件数]
[order-by 出力順基準式   [descending]]
construct 出力生成式
```

これは、XML-QL の基本構文に rank-by〜top 節を追加したものとなっている。また、後述の外部関数を適宜利用可能である点も XML-QL との主な違いである。

前節の例 1 は、⟨記事⟩における⟨要約⟩の付加と⟨記事分類⟩の除去を省くと次のように記述される。出力は⟨分類⟩の列を内容とする⟨文書⟩であり、⟨日別⟩は同じ値の⟨掲載日⟩を持つ⟨記事⟩の列である。

```
where     <文書> $x </>
construct <文書>
  where     <記事> <記事分類> $c </>
              <掲載日> <年> $y </> </>
            </> element_as $a in $x, $y>=1999
  order-by  $c
  construct <分類 ID=CategoryID($c) 分類名=$c>
              $a </>
            </>
```

where 節　問合せ条件として適合すべき文書構造やそれが満たすべき条件を指定し、必要に応じて変数の束縛を指定する節である。変数はその記述位置に応じて、文書要素, 文書要素内容, 文書要素名, 属性名, 属性値のいずれかに束縛され、それに関する述語によってその値の満たすべき条件が記述される。特に、パターン式中の element_as $x と content_as $x は、それぞれその直前の文書要素およびそのタグを除去した文書要素の内容で変数 $x を束縛する。尚、この変数の束縛は適合する全ての組合せに対して行なわれる。

冒頭の記述例では、一つ目の where 節で $x を⟨文書⟩の内部で束縛し、二つ目の where 節で $a を $x 中の各⟨記事⟩で、$y をその⟨年⟩の内容で、$c を⟨記事分類⟩の内容で束縛する。また、述語 $y>=1999 によって、1999 年以降のもののみが選択される。

construct 節　where 節のパターン式中で束縛された変数のそれぞれの組合せに対して生成される出力の構成を指定する節である。問合せ結果は、この記述の変数を展開して得られる文書要素の列となる。ここで、文書要素等が束縛された変数が属性値等のように文字列が期待される場所で参照される場合には、タグを取り除いて得られる文字列に展開される。construct 節中には where〜construct 節を入れ子にする事ができる。

また、生成する文書要素の属性 ⟨ID⟩ を、その引数と返値が一対一対応するスコーレム関数で与える事ができる。この時、親を共有し、同じ ⟨ID⟩ を持つ文書要素は、一つの文書要素実体に集約される。グルーピングはこれを利用して実現される。

冒頭の例では、副問合せの処理結果を内容とする、ただ一つの⟨文書⟩が生成される。また、副問合せでは⟨記事⟩ ($a) を内容とする⟨分類⟩の列が生成される。それらは、⟨記事分類⟩の内容である $c を値とする属性⟨分類名⟩が与えられる。また、スコーレム関数 CategoryID() によって⟨記事⟩が⟨分類⟩にグルーピングされる。

order-by 節　construct 節で生成される文書要素の出力順を指定する節である。出力は出力順基準式で指定された値をキーとして昇順に、descending が指定された場合にはその逆順にソートされる。

冒頭の例では、出力の⟨分類⟩を $c の値によって辞書順にソートして出力を行なう。

rank-by〜top 節　where 節で束縛された変数群の処理順を指定する節である。入力順基準式に基づいた順序で変数群が順位付けされ、上位の指定された選択件数のものが construct 節の適用対象として選択される。また、この処理順序は出力の順序に影響せず、order-by 節が指定されていなければ、入力文書中での出現順序に従って出力される。尚、前述の通り、これは X²QL に独自の節である。

記述内容処理においては、文書要素の間の類似度や、その重要度等に基づいてランキングをし、その上位の N 件のみを選択するという事がしばしば行なわれる。これをサポートするため、X²QL では rank-by〜top 節を提供する。また、文書においては、その出

現順序が重要であり、このような選択のために破壊的な操作を行なえない。そこで、rank-by～top 節は出力順には影響を及ぼさないものとなっている。

3.2 関数としての問合せ利用

XML-QL では、次の構文を用いて必要に応じて関数を定義する事ができる。

```
function 関数名 ( 引数リスト )
where～construct 問合せ
end
```

即ち、引数リストで与えられる非束縛変数を含む問合せとして記述可能な関数のみが定義可能であり、その問合せ結果が返値となる。これは柔軟かつ詳細な処理を行なう関数を記述できない。これは X^2QL でも利用可能であるが、機能拡張性を補うものではない。

3.3 外部関数

高度な文書処理においては、必要とされる機能は多岐に渡るため、組込み関数を拡充するというアプローチで網羅的に提供する事は本質的に不可能である。そこで、X^2QL においては、Java 等の汎用プログラミング言語による外部プログラムとして**外部関数**を与える事で、これらの問題を解決する。

外部関数は、不特定の対象に適用可能な**一般関数**と、特定の文書要素名を持つ文書要素に対して限定的に適用される**文書要素固有メソッド** (以下、メソッド) とに分けられる。これにより、文書要素毎に異なる処理を同名のメソッドとして定義可能である。

2 節の例を考える。例 1 では、要約生成を実現するための外部関数が必要とされる。その実装の例として、⟨ 記事 ⟩の⟨ 見出し ⟩に含まれる語を重要語とみなして要約を生成するというような処理が考えられる。これは⟨ 記事 ⟩に固有のメソッドとして与えるのが適切である。また、例 2 ではユーザが与えたキーワード群と⟨ 記事 ⟩との類似度を計算する外部関数を利用する事でランキング処理が可能になる。ここでは、一般関数として定義するものとする。その類似度の計算方法としては様々なものが提案されているが [14]、本研究のアプローチによってユーザの目的に適したものを利用可能である。

外部関数を利用するためには、**外部関数定義**と**外部関数実装**を与える。ここで外部関数定義とは、外部プログラムとして与えられる外部関数実装を、問合せ中で利用可能にするための情報の記述を指し、定義構文を以下に示す。

```
function  型 関数名 ( 引数リスト )
defined-by "関数実装を含む URI"

function  型 文書要素名. メソッド名 ( 引数リスト )
defined-by "関数実装を含む URI"

引数リスト ::= 型 引数名 [, 型 引数名 ]*
```

外部関数はその適用対象によって呼び分けられる。特に、メソッドの場合には、文書要素に関連付けられた固有の外部関数が呼び出される。これを実現するために、外部関数の定義では型を明示的してシグネチャを与える。関数の引数および返値として有効な型は、問合せ処理時に扱われる値を特徴付ける element (文書要素), content (要素内容), string (文字列), number (数値文字列), boolean (真偽値) の五つと、任意の文書要素型である。文書要素型は、文書要素名をその型名とする。

ここで、element 型と content 型は、element_as および content_as によって束縛された変数の型に相当し、個々の文書要素型は element 型の下位型として扱われる。また、number, boolean は、それぞれ数値、真偽値を表現した特別な文字列であり、表現形式は XML Schema [20] における double, boolean のそれに従う。即ち、number は IEEE 浮動小数点数であり、boolean は真値が true または 1、偽値が false または 0 である。型の扱いの詳細は 4.1 節で述べる。

メソッドの呼出しは、問合せ中で “文書要素変数. メソッド名 (引数リスト)” という形式で記述される。この時、変数に格納されている文書要素の名前を実行時に検査し、外部関数定義に従ってその文書要素型に関連付けられた適切なメソッドが呼ばれる。

上記の例で必要とされる要約生成関数を⟨ 記事 ⟩のメソッド abstract()、類似度計算関数を一般関数 sim_cosine() とし、Java によって外部関数実装が与えられるものとすると、例えば次のように外部関数定義が与えられる。

```
function  要約 記事.abstract()
defined-by "http://fqdn/path/
            pkg.article#abstract"

function  number sim_cosine(
                  element e, string y )
defined-by "http://fqdn/path/
            common.vecspace#cosine"
```

⟨ 記事 ⟩の文書要素固有メソッド abstract() は、Java におけるパッケージ pkg のクラス article のメソッド abstract() によって実装が与えられ、返値として⟨ 要約 ⟩が返される事が示されている。一般関数 sim_cosine() は、任意の文書要素と文字列として与えられるキーワード群から類似度を計算する。

3.4 問合せ記述例

3.3 節の外部関数を用いた、2 節の例 1, 2 の問合せの記述例を以下に示す。

問合せ記述例 1

```
where      <文書> $x </>
construct <文書>
```

表 1: 型変換の対応

変換元	string	number	boolean	content	element
string	—	$n_s(x)$	$b_s(x)$	$c_s(x)$	—
number	$s_n(x)$	—	$b_n(x)$	$c_s(s_n(x))$	—
boolean	$s_b(x)$	$n_b(x)$	—	$c_s(s_b(x))$	—
content	$s_c(x)$	$n_s(s_c(x))$	$b_s(s_c(x))$	—	$e_c(x)$
element	$s_e(x)$	$n_s(s_e(x))$	$b_s(s_e(x))$	$c_e(x)$	—

```
where     <記事>
            <記事分類> $c </>
            <掲載日> <年> $y </>
                     </> element_as $d
            <見出し> </> element_as $h
            <本文> </>   element_as $b
          </> element_as $a in $x, $y>=1999
order-by  $c
construct <分類 ID=CategoryID($c) 分類名=$c>
            <記事> $d $h $a.abstract() $b </>
          </> </>
```

問合せ記述例 2

```
where     <文書> $x </>
construct <文書>
  where     <記事> <記事分類> $c </>
            </> element_as $a in $x, $c='経済'
  rank-by   sim_cosine($a, keywords)
  top       N
  construct $a
            </>
```

問合せ記述例 1 において、3.1 節の例と同様に、出力の⟨ 文書 ⟩内には、副問合せで適合した全ての⟨ 記事 ⟩を⟨ 記事分類 ⟩に従ってグルーピングした結果の⟨ 分類 ⟩の列が辞書順に生成される。その属性として⟨ 分類名 ⟩を付加している。また、出力の⟨ 記事 ⟩には⟨ 記事 ⟩のメソッド abstract() によって生成された⟨ 要約 ⟩が付加され、⟨ 記事分類 ⟩が除去される。

問合せ記述例 2 では、⟨ 記事分類 ⟩が「経済」である記事に対して、類似度計算関数 sim_cosine() を適用する事でランキングおよび順位による絞り込みを行なっている。これは rank-by〜top 節によって実現されている。ここでは入力文書中での出現順序が保存される。

4 外部関数の評価モデル

X²QL は XML-QL の拡張であるため、問合せの基本的な評価モデルは XML-QL のそれに従う。本節では、X²QL の XML-QL に対する拡張部分である外部関数に関する評価モデルについて説明する。特に、4.1 節で問合せ処理における型制約の解消方法を、4.2 節で外部関数の対応付けと処理方法を述べる。

4.1 問合せ処理時の型制約の解消

本節では、外部関数の導入に伴う型 element, content, string, number, boolean 及び文書要素型の扱いについて説明する。特に、前者の五つの型は、問合せ処理時に扱われる値を特徴付け、内部的な処理において考慮される。また、文書要素型は、内部処理においてはその上位型である element 型として扱われるが、呼び出される外部関数を特定する際には個別に識別される。問合せ処理時には、大きく分けて、変数を束縛する値の型と参照される値の型について考慮する必要がある。

問合せ中では、element_as と content_as は element, content 型として変数を束縛し、タグ名や属性名、属性値等は string 型として変数を束縛する。これら以外に明示的な代入文は存在しないため、変数を束縛する値の型はこれら三つに限られる[1]。

参照される値としては、変数に束縛された値や関数の返値があり得る。特に後者では、任意の型が返され得る。値の参照は問合せ構文が許す任意の箇所で行なわれるが、その際、参照される値は構文上期待される型に変換されて展開される。変換不可能である場合、その問合せは処理されない。

値の参照箇所としては次が考えられる。

1. where 節中のパターン式において同名の変数が複数回出現する問合せ中での、その参照箇所
2. where 節中の述語
3. rank-by 及び order-by 節の基準式
4. top 節の選択件数
5. construct 節中の参照箇所
6. 外部関数引数

このうち、1, 5 で参照される値は、記述位置に応じて element, content, string 型である事が期待される。また、2 において、述語の評価結果は boolean 型である。特に、大小関係比較 <, <=, =>, > の両辺は number 型、等値関係比較 =, != の両辺は string 型として扱われる。3 では算術式であれば number 型として、そうでなければ string 型として扱われる。4 では number 型が、6 では関数定義に従った型が期待

[1] 本稿では説明を省いた添字変数の値である出現順序は number 型であり、実際にはこれを加えて四つの型を考える。

される。

ここで、期待される型と参照される値の型が一致しない場合、表 1 の変換が行なわれる。表中の $s(x)$ は値 x の string 型への変換を、$n(x)$ は number 型への変換を、$b(x)$ は boolean 型への変換を、— は変換が不要もしくは未定義である事を示す。また、$c(x)$ は x を単一の要素とする列を構成する事を示し、$e(x)$ は x の最初の要素を取り出す事を示す。各文書要素型が型変換の対象となる時には、element 型への変換が始めに適用されるものとする。各文書要素型以外の五つの型について以下で説明する。

element 型, content 型から string 型への変換結果は、タグを取り除いて得られるテキストである。number 型は、数値を示す文字列である事から、単純に string 型として扱われる。boolean 型は、その真偽によって string への変換では true, false となり、number への変換では 1, 0 となる。number 型, boolean 型への他の型からの変換時には、それに先立って string 型へ変換される。尚、string 型から number 型, boolean 型への変換は、その値が数値および真偽値を表現している時にのみ可能である。

element 型, string 型から content 型への変換は、それらの値のみからなる列を構成する事で与えられる。content 型から element 型への変換は、その値である列が一つの文書要素のみから構成されている時にのみ行なわれ、その時、唯一の内容である文書要素が変換結果となる。

4.2 外部関数の対応付けと処理モデル

3.3 節において、外部関数は一般関数とメソッドとがある事を述べた。特にメソッドにおいては、外部関数定義と文書要素を対応付ける必要がある。また、外部関数呼出しにおいては処理に必要な情報を実装へ渡さねばならない。本節では、これらの処理モデルを説明する。ここでは、特に Java による処理系と外部関数実装を考える。

文書要素固有メソッドの外部関数定義によって、外部関数実装を与える複数の Java クラスのメソッドが文書要素型に対応付けられる。ここで、文書要素型と Java の実装クラスの対応は、一般には一対多となり得る。処理系は問合せ処理に先立って、各文書要素型について、この対応付けを保持した**メソッドテーブル**と呼ぶオブジェクトを生成する (図 1)。尚、現在は Java のリフレクション機能を用いて Method オブジェクトを取得し、これを保持する連想配列を利用して実現している。

一方、問合せ処理時には必要に応じて、個々の文書要素に対応する**文書要素オブジェクト**と呼ぶオブジェクトを生成する。これは、文書要素に対する最初のアクセス時から問合せ処理の終了時まで生存し、同一の文書要素については最初に生成された文書要素オブジェクトが対応付けられ、利用される。

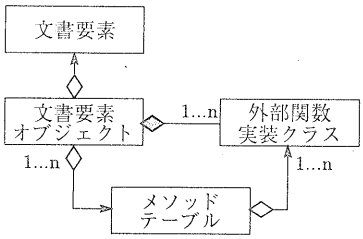

図 1: 文書要素オブジェクトクラス図

文書要素オブジェクトは、W3C DOM の Element インタフェースを拡張したインタフェースを持ち、これを通じて対応する文書要素へのアクセス機能を提供する。尚、文書要素オブジェクトは、標準の Element インタフェースが提供するメソッドを通じた他の文書要素へのアクセスによっても生成される。文書要素オブジェクト固有の拡張点は以下である。

- 文書要素固有メソッドの起動機能を提供
  - 適切なメソッドテーブルへの参照を保持
- 実装クラスに対して状態保持機能を提供
  - 実装クラスのインスタンスを生成、保持
  - 文書要素オブジェクトが連想配列を提供

文書要素固有メソッドの起動は、拡張インタフェースのメソッド起動機能によって行なわれる。外部関数呼出しにおいては、対応するメソッドテーブルから、同名の関数名で引数型が厳密に一致するものを検索する。ここで、関数名が一致するものの引数型が厳密には一致しない場合、4.1 節で説明した型変換が行なわれる。但し、複数の同名の関数が存在し、型変換が一意に定められない場合には関数呼出しは失敗する。対応する外部関数が決定されると、内部的にメソッドテーブルを通じて実装クラスのインスタンスのメソッドが起動される。

文書要素オブジェクトの状態を保持する機能は次の二点によって実現される。実装クラスのインスタンスを保持し続ける事でそれに固有の状態を保持する。また、文書要素オブジェクト所有の連想配列を実装クラスへ提供する事でそれらにまたがって文書要素オブジェクトの状態を保持、共有可能とする。文書要素オブジェクトが問合せ処理が終了するまで生存しているため、上記の状態保持機能によって、異なる文書要素固有メソッドの起動に渡って、文書要素オブジェクトの内部状態が保持される。この特性を用いて、ベクトル空間モデルにおける tf-idf 法のように大局的な情報が必要な場合には、そのメソッドが最初に起動された時点で必要な情報をあらかじめ調べておき、二度目以

降はそれを利用するといった処理の実装も与える事が可能である[2]。

実装クラスのメソッド起動時には、その引数リストの第一引数としてその文書要素オブジェクトへの参照が渡される。実装クラスからは、この参照を通じて文書要素オブジェクトの提供する機能を利用可能であるため、文書要素へのアクセス機能、メソッド起動機能、状態保持機能の全てを利用可能である。これにより、必要に応じて他の文書要素オブジェクトの文書要素固有メソッドの起動が可能である。

以上が、文書要素固有メソッドに関する処理モデルである。一般関数は、それら全てをまとめた無名の文書要素オブジェクトに対応付けられ、それへの参照が実装メソッドの第一引数として渡される点を除いて同様に処理される。

現在、我々は X^{2}QL 問合せを XSLT スタイルシート [19] へ変換するというアプローチで、XSLT 処理系を用いた処理系を実装している [17]-[18]。この処理系では、変換の際に、文書要素型に対応するスタブモジュールと呼ぶクラスを自動生成して利用する。本稿の処理モデルではメソッドテーブルを用いて間接的に文書要素固有メソッドを提供するが、スタブはこれを Java クラスのメソッドとして直接提供する。外部関数呼出しは、スタブのメソッドを起動するような XSLT の拡張関数呼出しに変換され、スタブのメソッドは文書要素オブジェクトのメソッド起動機能を用いて文書要素固有メソッドを起動する事で、XSLT 処理系と X^{2}QL 外部関数処理系の連携を行なっている。

5 関連研究

現在、XML に対する様々な問合せ言語が提案されている。

W3C へ提案された XML-QL [3] は XML に対する問合せ言語である。これは前述の通り、X^{2}QL における問合せ記述の基本となっている。しかし、3.2 節で述べた方法でユーザが定義する関数は、既存の枠組みに基づく問合せとして記述可能なもののみであり、機能拡張性を補うものではない。また、今後の拡張としてユーザ定義述語の導入について言及されているが、現在のところその詳細については明確に述べられていない。高度な文書処理においては、述語に限らない幅広い処理機能の拡張性が要求される。X^{2}QL の外部関数は、引数や返値の型が適正であれば任意の位置に記述する事ができ、必要に応じて文書要素や記述内容そのものを生成する事も可能である。

XSLT [19] は W3C により策定された XML 用変換言語であり、そのパス記述言語 XPath [21] の拡張として XQL [4] が提案されている。XQL と XPath は再構成能力は持っていない。また、XLST は記述能力は高いものの、その記述は低水準で手続き的なものとなる。また、ユーザ定義の拡張関数を利用可能であるが、そのサポートの有無および利用方法は、その処理系の実装依存である。4.2 節述べたように、我々は XSLT 処理系を用いた X^{2}QL 処理系のプロトタイプシステムを構築中である。

YATL は木構造データモデル上のシステムである YAT [8] 用の言語である。これはパターン間の変換に基づくルール指向の言語であり、やはり高い文書構造操作の能力を持つ。文字列処理等のための拡張関数が利用可能であるが、高度な文書処理に必要とされる機能の実現には言及されていない。

Lorel [5] はラベル付き有向グラフデータモデルである OEM に対する問合せ言語であり、OQL [22] の拡張として捉える事ができる。これは簡潔さを重視しており、再構成の能力をほとんど持たない。一方、UnQL [6] もまた、OEM と同様の有向グラフデータモデルに対する問合せ言語であるが、高い再構成の能力を持つ。これらにおいても、構造の位置関係や値の単純な関係しか扱う事ができず、処理機能の拡張性を持っていない。

また、XML におけるデータモデルに関する議論も行なわれている。

XML Information Set [23] では XML 文書において扱われる基本的な構成要素について規定している。XPath や以下の二つを含む、W3C で標準化されるデータモデルの多くはこれに準拠しつつある。

XML Schema [20] では、DTD の記述力不足を補うため、リテラルとして書かれるデータの型 (単純型) や文書要素によって記述されるデータの型 (複合型) について述べている。

XML Query Data Model [24] では、XML 問合せ言語における W3C XML 問合せ代数の策定に向けて、その基礎となるデータモデルを提案している。これは、木構造データモデルであり、XML Schama におけるデータ型や、コレクション型や参照の表現方法などをサポートする。

本稿では、XML Schema における一部の単純型に相当する型と、`element` 型の下位型として文書要素型を導入した。今後、本稿で提案した X^{2}QL も上記を代表とする標準化動向と整合するよう、検討を行なうべきであると考えられる。

6 まとめ

従来の構造化文書を対象として問合せ言語では、文書構造に対する比較的強力な操作体系を有するものの、多様化、複雑化する文書処理要求に対して、必ずしも十分な機能を提供しているとは言えない状況がある。一方で、それらの機能を網羅的に提供する事は非現実的である。この問題を解決するには、問合せ言語

[2] 集約関数のような、問合せのコンテクストに応じた処理を出力生成前に行なう事はできないため、何らかの拡張によりサポートする必要があると思われる。

に柔軟な拡張性を持たせ、利用者がその目的に合わせて必要な処理機能を導入可能にする必要がある。

本稿では、XML-QL をベースとし、必要に応じた外部関数による機能拡張性を持つ、拡張可能 XML 問合せ言語 X^2QL について説明した。また、XML-QL に対する拡張部分である外部関数について、その型の扱いや、文書要素との関連付け方法、処理モデルを示した。

今後、提案問合せ言語と DTD 等のスキーマ情報との関連について、特に文書要素間の継承関係等について考察を行なっていく予定である。また、型システム等のデータモデルについて、関連技術の標準化動向との整合性を保つための検討や、集約関数として用いられる外部関数に対するサポートの導入が必要であると思われる。X^2QL の記述能力の検証や、大規模な文書データベースにおける処理の効率化も重要な課題である。

謝辞

本研究の一部は、文部省科学研究費基盤研究 (B) (12480067) 及び奨励研究 (A) (12780183) の助成による。

参考文献

[1] R. Sacks-Davis, T. Arnold-Moore and J. Zobel. Database Systems for Structured Documents, *International Symposium on ADTI '94*, pp.272-283, Nara, 1994.

[2] World Wide Web Consortium, http://www.w3.org/.

[3] A. Deutsch, M. Fernandez, D. Florescu, A. Levy and D. Suciu. A Query Language for XML, *Proceedings of the Eighth International World Wide Web Conference (WWW8)*, Computer Networks, Vol. 31, No. 11-16, pp. 1155-1169, 1999.

[4] J. Robie, J. Lapp and D. Schach. XML Query Language (XQL). *The Query Languages Workshop (QL'98)*, http://www12.w3.org/TandS/QL/QL98/pp/xql.html, 1998.

[5] S. Abiteboul, D. Quass, J. McHugh. J. Widom and J. Wiener. The Lorel Query Language for Semistructured Data, *International Journal on Digital Libraries*, Vol. 1, No. 1, pp. 68-88, 1997.

[6] P. Buneman, S. B. Davidson, G. G. Hillebrand and D. Suciu. A Query Language and Optimization Techniques for Unstructured Data, *Proceedings of ACM-SIGMOD '96*, pp. 506-516, Motreal, 1996.

[7] M. F. Fernandez, D. Florescu, J. Kang, A. Y. Levy and D. Sucie. Catching the Boat with Strudel: Experiences with a Web-site Management System, *Proceedings of ACM-SIGMOD '98*, pp. 414-425, Seattle, 1998.

[8] S. Cluet, C. Delobel, J. Simeon and K. Smaga. Your Mediators Need Data Convention!, *Proceedings of ACM-SIGMOD '98*, pp. 414-425, Seattle, 1998.

[9] D. Konopnicki and O. Shinueli. W3QL: Query System for the World Wide Web, *Proceedings of Twenty-First Conference on VLDB*, pp. 54-65, Zurich, 1995.

[10] A. Mendelzon, G. Mihaia and T. Milo. Querying the World Wide Web, *International Journal on Digital Libraries*, Vol. 1, No. 1, pp. 54-67, 1997.

[11] 田島 敬史. "半構造データのためのデータモデルと操作言語", 情報処理学会論文誌データベース, Vol. 40, No. SIG 3 (TOD 1), pp. 152-170, 1999.

[12] I. Mani and M. T. Maybury (eds.). *Advances in Automatic Text Summarization*, MIT Press, 1999.

[13] J. Zobel and A. Moffat. Exploring the Similarity Space, *ACM SIGIR Forum*, Vol. 32 No. 1, pp. 18-34, 1998.

[14] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Reading, MA, 1989.

[15] 奥村 学, 難波 秀嗣. "テキスト自動要約に関する研究動向", 自然言語処理,「テキスト要約のための言語処理」特集号, Vol. 6, No. 6, 1999.

[16] H. A. Hearst. Subtopic Structuring for Full-Length Document Access, *Proceedings of ACM-SIGIR '93*, pp. 59-68, Pittsburg, 1993.

[17] 品川 徳秀, 北川 博之, 石川佳治. "拡張可能 XML 問合せ言語 X^2QL とその処理系", 第 11 回データ工学ワークショップ (DEWS2000), 2000.

[18] N. Shinagawa, H. Kitagawa and Y. Ishikawa. X^2QL: An eXtensible XML Query Language Supporting User-Defined Foreign Functions, *2000 ADBIS-DASFAA*, Praha, 2000. (to appear)

[19] J. Clark (ed.). *XSL Transformations (XSLT)*, http://www.w3.org/TR/xslt, 1999.

[20] D. C. Fallside; H. S. Thompson, D. Beech, M. Maloney and N. Mendelsohn; P. V. Biron and A. Malhotra (eds.). *XML Schema Part 0: Primer, Part 1: Structures, Part 2: Datatypes*, http://www.w3.org/TR/xmlschema-[0-2]/, 2000.

[21] J. Clark and S. DeRose. *XML Path Language (XPath) Version 1.0 (working draft)*, http://www.w3.org/TR/WD-xpath, 1999.

[22] R. G. G. Cattell, D. K. Barry, M. Berler, J. Eastman, D. Jordan, C. Russell, O. Schadow, T. Stanienda and F. Velez (eds.). *The Object Data Standard: ODMG 3.0*, Morgan Kaufmann Publishers, 2000.

[23] J. Cowan and D. Megginson (eds.). *XML Information Set (XML Infoset)*, http://www.w3.org/TR/xml-infoset, 1999.

[24] M. Fernandez and J. Robie (eds.). *XML Query Data Model*, http://www.w3.org/TR/query-datamodel, 2000.