

## 拡張可能 XML 問合せ言語 X<sup>2</sup>QL を用いた WWW アプリケーション統合

倉垣 公一、品川 徳秀、北川 博之

〒 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

電話番号：0298(53)5385

E-mail：{kuragaki,siena,kitagawa}@dblab.is.tsukuba.ac.jp

近年、各種 WWW アプリケーションが充実してきており、これらの統合利用要求が高まっている。また、インターネット上での情報交換のベースとして XML が注目されている。これらより、本研究では出力データに XML 文書を用いる WWW アプリケーションを対象に、その統合利用を容易にする環境を提供する。具体的に、XML に対する問合せ言語を用いることで、高水準な統合記述を可能とすることを目的とする。WWW アプリケーション統合では、従来の XML 問合せ言語の記述能力に加え、(1) 各アプリケーションへの要求や WWW ページ内のナビゲーションの記述、(2) 一連の処理手順の記述、が必要である。本研究では、我々が研究開発中の X<sup>2</sup>QL の外部関数を用いることで (1) を、新たな処理記述言語 X<sup>2</sup>PL を導入することで (2) を可能とする。

キーワード WWW アプリケーション統合, XML, X<sup>2</sup>QL, X<sup>2</sup>PL

## WWW Application Integration with an eXtensible XML Query Language X<sup>2</sup>QL

KURAGAKI Kouichi, SHINAGAWA Norihide, KITAGAWA Hiroyuki

〒 305-8573 1-1-1 Temnodai Tsukuba Ibaraki Japan  
Institute of Information Sciences and Electronics, University of  
Tsukuba.

Tel：0298(53)5385

E-mail：{kuragaki,siena,kitagawa}@dblab.is.tsukuba.ac.jp

### Abstract

With the recent and rapid advance of the internet technology, WWW application integration has become more and more important. XML is expected to be widely used for information exchange on the internet. In this paper, we propose an approach for WWW application integration based on an XML query language. It enables higher level integration specification than approaches based on programming languages. In our approach, we use our extensible XML query language, X<sup>2</sup>QL. Foreign functions in X<sup>2</sup>QL enable us to invoke WWW applications and to navigate through WWW pages. We also introduce X<sup>2</sup>PL to specify processing procedures.

key words WWW Application Integration, XML, X<sup>2</sup>QL, X<sup>2</sup>PL

## 1 はじめに

近年のWWWの普及に伴い、WWW上で利用可能なアプリケーションが増加している。例えば、WWWページの検索や電子商取引等のサービスが提供されている。今後、これらWWWアプリケーションが一層充実していくに従い、これらを統合して一つのWWWアプリケーションとして利用したいという要求が高まってくるのが予想される[4][3][2][6]。統合利用の簡単な例として、amazon.com等のサイトで書籍情報を調べた後、図書館の蔵書検索システムを用いて当該書籍の有無を調べたいというような利用が考えられる。しかし、それらのサービス内容や利用方法は様々であり、個々のWWWアプリケーションを統合して新たなWWWアプリケーションを構築するには、著しいコストが生じる。更に、個々のWWWアプリケーションに変更が生じた場合には、それに対応するための修正コストも問題になる。

ところで、近年、インターネットにおける情報交換においてXML[11]が注目されており、WWWページや各種アプリケーションが保持するデータの記述形式にXMLを用いることが一般的になりつつあるという流れがある。このことから、WWWアプリケーションへの入力データやその結果出力の記述にXMLが利用されることが一般的となると思われる。

本研究では、以上のような背景に基づき、XML文書に対する操作記述を用いた複数のWWWアプリケーションの統合の実現を目的とする。具体的には、各WWWアプリケーションをXML文書を入力出力インタフェースとする抽象マシンと捉え、XML文書に対する問合せ言語による高水準な記述をベースとした、WWWアプリケーション統合記述を可能とする。個々のWWWアプリケーションへの要求とその処理内容の受渡しの記述には、我々の研究グループで研究開発中のXML問合せ言語X<sup>2</sup>QL[9][8]を用いる。X<sup>2</sup>QLは、外部関数を導入することで必要に応じて種々の処理機能を拡張し、利用することが可能である。本研究ではこの特徴を利用して、XMLで記述されたデータをWWWアプリケーションへの入力として渡す機能や、WWWページのナビゲーション機能を外部関数として導入することで、これらを利用した処理を記述可能とする。また、WWWアプリケーションを統合するために必要とされる一連の処理や制御の流れを記述する言語として、X<sup>2</sup>PLを導入する。

以下、2節では、XML問合せ言語をベースとしたWWWアプリケーション統合の実現における問題点と本研究におけるアプローチを説明する。3節では、本研究が目的としているWWWアプリケーション統合利用の例を示す。4節では、我々の研究グループで研究開発中のX<sup>2</sup>QLについて説明する。5節では、本研究においてWWWアプリケーションの起動やWWWページのナビゲーション処理を行なう際に必要となるX<sup>2</sup>QLの外部関数について説明する。6節では、本研究で導入するX<sup>2</sup>PLについて説明する。7節では、本研究での

手法を用いた統合例を示す。8節では、システムの概略を説明する。最後に、9節で、まとめと今後の課題を述べる。

## 2 WWWアプリケーション統合へのアプローチ

### 2.1 WWWアプリケーション統合

前節で述べた通り、インターネット上で多種多様なWWWアプリケーションが利用可能になるにつれ、それらを統合利用することへの要求が高まりつつある。統合利用の利点は、新規にアプリケーションを構築することなくより充実したサービスをユーザに提供可能であることである。つまり、個々のWWWアプリケーションではユーザが要求するサービスを提供することが困難な場合でも、既存のWWWアプリケーションから得られるデータを用いて上位の統合アプリケーションがデータを再構成することにより、ユーザが必要とするサービスをより適切に提供することが可能になる。具体的には、前節で述べたように、あるキーワードに関連した出版書籍のうち、図書館に存在する書籍を検索するような統合利用が考えられる。このようなWWWアプリケーション統合利用例を図1に示す。

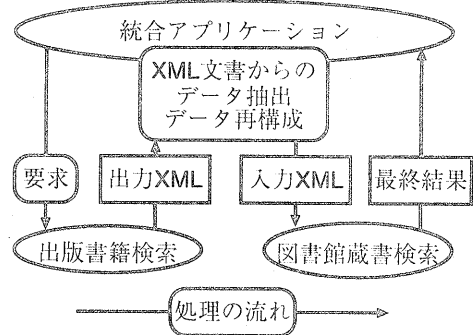


図1: 統合利用例

図1に示すように、WWWアプリケーション統合を行なうには、データの入力を伴うWWWアプリケーションへの要求、出力XML文書からの必要なデータの抽出、別のWWWアプリケーションへの入力データとなるXML文書の再構成等が必要である。また、これら処理の流れをユーザの意図通りに制御する機能も必要である。

### 2.2 統合へのアプローチ

WWWアプリケーションの統合を記述するための手法としては、大きく三つの手法が考えられる。それらは、(1)Javaのような汎用のプログラミング言語を用いる手法、(2)WWWアプリケーションの処理に特化

した WebL[10] のようなプログラミング言語を用いる手法、(3) XML 文書に対する問合せ言語に基づく記述を利用する手法である。

(1)、(2)の手法は両者ともプログラミング言語を用いるため、柔軟で自由な統合処理を行なうことが可能である。しかし、新規開発や保守にかかるコストが大きいが予想される。加えて、プログラミングにおける専門知識が要求されるため、対象となるユーザが限定される。(3)の手法は、問合せ記述のための知識は必要であるが、プログラム記述に比べ高水準であるためプログラミングの専門知識を必要としない。このため、(1)、(2)に比べ、より一般的なユーザによる WWW アプリケーション統合が可能である。高水準な記述が可能で問合せ言語を用いて、文書操作に着目して WWW アプリケーション統合を行なうため、保守などに伴う負荷は(1)、(2)によるプログラムの変更に伴う負荷に比べて軽減される。反面、統合に用いる問合せ言語の記述能力によっては、WWW アプリケーション統合の自由度が制限される。しかし、上述したような利点があるため、WWW アプリケーション統合の一つの有力なアプローチであると考えられる。また、本稿で提案する X<sup>2</sup>PL のような処理の流れの制御を記述するための言語を併用することで、記述能力の制約を緩和することが可能である。これらの理由により、本研究では(3)の手法を用いた WWW アプリケーション統合について検討する。

### 2.3 問合せ言語を用いた統合における課題

前述の通り、本研究では、XML に対する問合せ言語を用いた文書操作記述レベルでの WWW アプリケーション統合を行なう。

しかし、WWW アプリケーションを統合する上で既存の問合せ言語をそのまま用いることにはいくつかの問題がある。一般に、WWW アプリケーションを統合するためには、適切な入力データを与え、条件分岐や繰り返し処理を含むユーザが意図する手順に従って各 WWW アプリケーションを利用し、得られたデータの処理を行なうことが可能でなければならない。ところが、一般的な問合せ言語では、XML 文書から必要な部分を取り出したり、それらを再構成したりすることは可能であるが、動的に WWW アプリケーションへ要求を行ったり、データを受け渡したりすることができない。また、ユーザが意図する処理手順を記述する手段も提供されていない。

本研究では、前者の問題に対応するため、我々の研究グループで研究開発中である X<sup>2</sup>QL を利用する。X<sup>2</sup>QL は、XML-QL をベースとした XML 問合せ言語であり、外部関数の導入による処理機能の拡張を可能としている。この拡張性により、問合せとして宣言的に記述することが困難な WWW アプリケーションへの要求等の処理機能を外部関数として与え、問合せ記述中で利用することが可能である。また、後者の問題に対応するため、本稿で新たに X<sup>2</sup>PL を導入する。こ

れにより、ユーザが意図する処理手順の記述を可能とする。

### 3 統合例

本節では、本研究で扱う WWW アプリケーション統合の具体例を示す。まず、統合対象の WWW アプリケーションをどのような XML 文書を入出力とするアプリケーションとしてモデル化するかを示し、次に、アプリケーションの統合利用例を示す。4節及び7節では、本節の例を用いて X<sup>2</sup>QL 及び X<sup>2</sup>PL 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。

**WWW アプリケーションのモデル化** 統合対象の WWW アプリケーションとして、出版書籍検索サイトと図書館蔵書検索サイトを想定する。これらが提供するページは XML (XHTML) で記述されていると仮定する。

現在の多くの WWW アプリケーションは、データ入力に HTML のフォームを用い、その処理結果も HTML で記述されている。しかし、今後はこれらの記述に XML が利用されるようになると予想される。また、適切なラッパーを用意することで、現状でもこれを XML で記述されたページを入出力のベースとするアプリケーションとして扱うことが可能である。

出版書籍検索サイトの検索条件の入力ページは、分類とキーワードの入力を必要とする。検索結果は N 件ずつ提供される。その出力ページは、高々 N 件の書籍情報と残りの書籍情報リストである他の出力ページの URL を含み、1冊の書籍情報は題目、著者、価格から構成される。図書館蔵書検索サイトの検索条件の入力ページは題目、著者、キーワードのうち少なくとも一つの入力を必要とする。検索結果の出力ページは全件の書籍情報を含み、1冊の書籍情報は題目、著者、所在から構成される。XML におけるフォームやハイパーリンクについて W3C で検討が行なわれているが [7][5]、現在のところ十分に利用可能とはなっていないため、入力ページは XHTML を用いて記述されるものとし、そのインスタンスの例を次に示す。また、出力ページは XML を用いて記述されるものとし、その DTD を次に示す。

```
<!-- 出版書籍検索条件入力ページ -->
<HTML>
<BODY>
<FORM>
<INPUT TYPE="TEXT" NAME="PRODUCT"/>
<INPUT TYPE="TEXT" NAME="KEYS"/>
</FORM>
</BODY>
</HTML>
```

```
<!-- 出版書籍検索結果出力ページ -->
<!ELEMENT RESULT (BOOK+,ANCHOR?)>
```

```

<!ELEMENT BOOK (TITLE,AUTHOR,PRICE)>
<!ELEMENT TITLE #PCDATA>
<!ELEMENT AUTHOR #PCDATA>
<!ELEMENT PRICE #PCDATA>
<!ELEMENT ANCHOR #PCDATA>

```

```

<!-- 図書館蔵書検索条件入力ページ -->
<HTML>
<BODY>
<FORM>
<INPUT TYPE="TEXT" NAME="TITLE"/>
<INPUT TYPE="TEXT" NAME="AUTHOR"/>
<INPUT TYPE="TEXT" NAME="KEYS"/>
</FORM>
</BODY>
</HTML>

```

```

<!-- 図書館蔵書検索結果出力ページ -->
<!ELEMENT RESULT BOOK+>
<!ELEMENT BOOK (TITLE,AUTHOR,PLACE)>
<!ELEMENT TITLE #PCDATA>
<!ELEMENT AUTHOR #PCDATA>
<!ELEMENT PLACE #PCDATA>

```

**統合処理例** ユーザが、現在出版されている書籍からあるキーワードに関連するものを検索し、それらが図書館の蔵書の中に存在するかどうかを知りたいとする。これを行なうためには、出版書籍検索サイトの出力ページの情報を用いて図書館蔵書検索サイトで検索を行なうという統合処理が必要となる。

## 4 拡張可能 XML 問合せ言語 X<sup>2</sup>QL

本研究で利用する X<sup>2</sup>QL は、XML に対する代表的な問合せ言語の一つである XML-QL[1] をベースとした構文によって記述される。また、問合せ中にはユーザが外部プログラムとして与え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を記述することが可能である。以下では 4.1 節で XML-QL を簡単に説明し、4.2 節で X<sup>2</sup>QL に導入されている外部関数について述べる。

### 4.1 XML-QL

#### 4.1.1 基本構文

XML-QL は XML 文書に対する問合せ言語であり、基本構文は次で与えられる。

```

WHERE  パターン式及び変数束縛 [IN 対象]
      [, パターン式及び変数束縛 [IN 対象]]*
      [, 述語]*
CONSTRUCT 出力生成式

```

3節の図書館蔵書検索出力ページの DTD に従う XML 文書から (TITLE) と (AUTHOR) を抽出する文書操作は、次のように記述される。

```

WHERE  <RESULT></> CONTENT_AS $x
      IN "output.xml"
CONSTRUCT <BOOKLIST>
      WHERE  <BOOK>
            <TITLE></> ELEMENT_AS $t
            <AUTHOR></> ELEMENT_AS $a
            </> IN $x
      CONSTRUCT <BOOK>
            $t $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。CONSTRUCT 節中において WHERE ~ CONSTRUCT 節を入れ子にする事が可能であり、一番外側にある WHERE 節は省略可能である。変数は文書要素、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 \$e と CONTENT\_AS \$c 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それぞれの変数 \$e と \$c 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。

この記述例では、1 行目の WHERE 節で \$x を (RESULT) の要素内容で束縛し、4 行目の WHERE 節で \$t と \$a をそれぞれ (TITLE) と (AUTHOR) の要素で束縛する。

**CONSTRUCT 節** WHERE 節のパターン式中で束縛された変数の各組み合わせから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。

この記述例では、副問合せの処理結果を内容とする、ただ一つの (BOOKLIST) が生成される。また、副問合せでは (TITLE) と (AUTHOR) を内容とする (BOOK) の列が生成される。

#### 4.1.2 関数定義

次の構文を用いて、ユーザは必要に応じて関数を定義する事が可能である。即ち、問合せとして記述可能な関数のみが定義可能である。

```

FUNCTION 関数名 ( 引数リスト )
WHERE ~ CONSTRUCT 問合せ
END

```

### 4.2 外部関数

XML-QL で定義可能な関数は問合せとして記述可能なもののみであるが、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。また、高度なデータ処理に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も不可

能である。そこで、X<sup>2</sup>QL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。

外部関数の定義構文は次の通りである。メソッドは、問合せ中で“文書要素名.メソッド名(引数リスト)”という形式で利用される。

```
FUNCTION データ型 関数名(引数リスト)
DEFINED-BY "関数実装を含む URI"
```

```
FUNCTION データ型
      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 ::= データ型 引数名
            [, データ型 引数名]*
```

外部関数の定義では型を明示的に与える。関数の引数および返値として有効な型は、NUMBER(数値)、STRING(文字列)、ELEMENT(文書要素)、CONTENT(要素内容)、BOOLEAN(真偽値)の五つと、任意の文書要素型である。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する。個々の文書要素型は、ELEMENT型の下位型として扱われ、文書要素名をその型名とする。

## 5 外部関数とその利用

XML-QLはXML文書に対する問合せ言語であるが、問合せの対象となるXML文書をWWWアプリケーションとの対話的な処理によって動的に得ることはできない。そこで、本研究ではX<sup>2</sup>QLを用い、外部関数によってWWWアプリケーションとの対話的な処理機能を導入することで、問合せ対象となるXML文書を動的に取得することを可能にする。

多くのWWWアプリケーションからの情報取得は、フォームへの入力をもって行なわれる。このため、フォームへの入力を行ない、出力結果を受け取る機能が必要である。また、結果が多数存在する場合には、結果が複数のWWWページに分割されて提供されることが多い。この時、これらの分割提供される個々のページには、他のページより詳細な情報を示すページのURLがハイパーリンクとして含まれる。そのため、これらのリンクされたページを取得する機能も必要である。そこで、本研究では前者をsubmit()、後者をfollow()という外部関数として導入する。

### 5.1 submit()

フォームを用いたWWWアプリケーションへ入力を行なうためには、フォームを構成するコントロールと呼ばれる文書要素に適切な入力を与え、それらを引数としてフォームの中で指定されたURLへ送信する必要がある。〈INPUT〉や〈TEXTAREA〉等の各コントロールは〈NAME〉属性による値を持っている。この処理を行なうため、外部関数submit()を(FORM)のメソッドとして導入する。submit()は次で定義される。

```
FUNCTION CONTENT FORM.submit(CONTENT $args)
DEFINED-BY "http://..."
```

ここで、\$argsは1回の検索に必要なフォームへの入力データを記述した下記の構造を持つ(INPUTDATA)の列であるものとする。

```
<INPUTDATA>
  <PRODUCT>...</>
  <KEYS>...</>
</>
```

submit()は、各(INPUTDATA)からWWWアプリケーションへ送信する引数を生成し、それらを指定されたURLへ送信することで得られたページの列を結果として返す。〈INPUTDATA〉からのURLの生成は次のように行なう。〈INPUTDATA〉の子文書要素の文書要素名と(FORM)の子文書要素の文書要素名が一致するものに対して、その文書要素名と(INPUTDATA)内のその文書要素名に対応する要素内容との組を作成し、この組の列を引数としてURLを生成する。

### 5.2 follow()

WWWにおいては、ハイパーリンクをたどることによって指定されたページへと移動することが可能である。WWWアプリケーションでもこの機能が利用されている。例えば、WWWアプリケーションへの要求結果が複数ページに分割されて提供される場合である。各ページは、一定件数ずつの結果と他のページへのハイパーリンクを持つか、リンク先のページがリンク元のページの詳細なデータを提供している。そこで、リンクされたページを取得する処理を外部関数follow()が行なうことにする。follow()は、ハイパーリンクを記述した文書要素〈ANCHOR〉のメソッドとして次で定義される。

```
FUNCTION CONTENT ANCHOR.follow(STRING $path)
DEFINED-BY "http://..."
```

\$pathは、正規パス表現を値とする文字列である。この正規パス表現は、〈ANCHOR〉の要素内容であるURLのページを起点としたナビゲート条件を表す。follow()は、この正規パス表現に一致するURLのページ全てを要素とする列を返す。引数が指定されない場

合は、(ANCHOR)の要素内容である URL のページを返す。

```
<KEY>database</>
</>
```

変数 \$cont に (INPUTDATA) 要素を追加する。

## 6 XML 処理記述言語 X<sup>2</sup>PL

前節では、必要な機能を X<sup>2</sup>QL における外部関数として導入することで問題の一つを解決した。しかし、X<sup>2</sup>QL は逐次処理や条件分岐、繰り返し等を含む手順の記述をサポートしていない。そこで、本研究ではこの記述を行なうための制御記述言語 X<sup>2</sup>PL を新たに導入する。これによって、より柔軟な WWW アプリケーション統合が可能になる。X<sup>2</sup>PL の構文を図 2 に示す。

```
X2PL ::= ブロック
ブロック ::= (代入文 | 追加文 | 条件文 |
              繰り返し文 | 返り文)*
代入文 ::= 変数'←-' 式
追加文 ::= 変数'<<-' 式
条件文 ::= 'if(' 述語'){' ブロック'}'
          ('else{' ブロック'}')?
繰り返し文 ::= 'while(' 述語'){' ブロック'}'
返り文 ::= 'return' 式?
式 ::= X2QL | 数値 | 文字列 | 真偽値
```

図 2: X<sup>2</sup>PL 構文

また、X<sup>2</sup>PL では、記述上のシーケンスが処理順序を意味する。以下では各構文の説明を行なう。

### 代入文

変数に要素、要素内容、数値、文字列または真偽値を代入するために使用する。要素や要素内容は X<sup>2</sup>QL を用いて記述する。記述例を次に示す。

```
$num ← 0
```

```
$elem ←
  CONSTRUCT <INPUTDATA>
    <PRODUCT>books</>
    <KEY>database</>
  </>
```

一つ目は変数 \$num に 数値 0 を代入する。二つ目は変数 \$elem に (INPUTDATA) 要素を代入する。

### 追加文

変数に要素、要素内容、数値、文字列または真偽値を追加するために使用する。要素や要素内容は X<sup>2</sup>QL を用いて記述する。記述例を次に示す。

```
$cont <<-
  CONSTRUCT <INPUTDATA>
    <PRODUCT>books</>
```

### 条件文

述語を用いて記述された条件に基づいて、次に行なうべき処理を決定するために使用する。述語は、表 1 の比較演算子を用いて記述される。

表 1: 使用可能演算子

|         |              |
|---------|--------------|
| <(=,.)= | : 数値の大小比較    |
| =.!=    | : 数値、文字の同値比較 |
| &&      | : AND 演算子    |
|         | : OR 演算子     |

else 節は省略可能である。条件分岐後の処理内容はブロックを使用して記述される。記述例を次に示す。

```
if($num > 0){
  $result ←
    CONSTRUCT <RESULT>success</>
}
else{
  $result ←
    CONSTRUCT <RESULT>error</>
}
```

上の例では、\$num の値によって、\$result を束縛する要素内容を変更する。

### 繰り返し文

述語を用いて記述された条件を満たす限り、特定の処理を繰り返すために使用する。述語は条件分岐の述語と同一のものが記述可能である。記述例を次に示す。

```
$num ← 0
while($num < 10){
  $count <<-
    CONSTRUCT <RESULT>$num</>
  $num ← $num + 1
}
```

0 から 9 までの数値を要素内容とする (RESULT) を \$count に追加する。

### 返り文

最終的に結果として返される文書を送すために使用する。記述例を次に示す。

```
return $result
```

ここで、\$result は結果として返される文書を示す。

## 7 統合記述例

前節までで説明した外部関数と X<sup>2</sup>PL を用いると、3節の統合処理例は以下のように記述される。大まかな流れは、以下の通りである。

1～5行目 出版書籍検索フォームへの入力データを作成する。

6～12行目 出版書籍検索サイトを利用し、検索結果を得る。

13～28行目 出版書籍検索サイトでの検索結果を元に、図書館蔵書検索フォームへの入力データを最高50件作成する。

29～48行目 図書館蔵書検索フォームへの入力データが存在するなら、図書館蔵書検索サイトを利用し、各入力データを条件とした全ての検索結果を得る。

55行目 最終結果を返す。

また、詳細については、次の通りである。

1～5行目 CONSTRUCT 節で作成された XML 文書を \$a\_input に代入する。

7～9行目 \$a\_form を出版書籍検索サイトページの入力フォーム (FORM) で束縛する。

10～11行目 \$a\_form のメソッド submit() を \$a\_input を引数として呼び出し、出版書籍検索サイトでの検索結果で \$r を束縛する。

13～14行目 繰り返しの初期値と条件であり、\$num が50未満かつ \$a\_result を束縛した要素内容がある間、15～27行の処理を繰り返す。

15～19行目 \$a\_result 内の個々の <BOOK> の要素内容を <INPUTDATA> の要素内容として再構成することで図書館蔵書検索サイトへの入力データを作成し、\$t\_input に追加する。

20行目 \$t\_input を束縛している要素数 (図書館蔵書検索サイトへの入力データ数) を \$num に代入する。numberIn() は引数として渡された CONTENT 型の値に含まれる文書要素の数を返す外部関数である。

21～27行目 \$a\_result を束縛している XML 文書内の <ANCHOR> の要素内容である URL が示すページで \$a\_result を束縛しなおす。

29行目 15行目で得られた \$t\_input が空でないかどうかを調べる。

30～47行目 \$t\_input から得られる書籍を図書館蔵書検索サイトで検索し、蔵書の中にあればその書籍の情報を \$result へ代入する。

49～53行目 \$t\_input が空であった場合の最終結果を CONSTRUCT 節で生成し、\$result へ代入する。

55行目 \$result を最終結果として返す。

```

1:$a_input <-
2:  CONSTRUCT <INPUTDATA>
3:          <PRODUCT>books</>
4:          <KEYS>database</>
5:          </>

6:$a_result <-
7:  WHERE <HTML>
8:          <FORM></> ELEMENT_AS $a_form
9:          </> IN "出版書籍検索サイト URL",
10:         <RESULT></> ELEMENT_AS $r
11:         IN $a_form.submit($a_input)
12:  CONSTRUCT $r

13:$num <- 0
14:while($num < 50 && $a_result != ""){
15:  $t_input <<-
16:    WHERE <RESULT>
17:          <BOOK></> CONTENT_AS $b
18:          </> IN $a_result
19:    CONSTRUCT <INPUTDATA> $b </>
20:  $num <- CONSTRUCT numberIn($t_input)
21:  $a_result <-
22:    WHERE <RESULT>
23:          <ANCHOR></> ELEMENT_AS $link
24:          </> IN $a_result,
25:          <RESULT></> ELEMENT_AS $n
26:          IN $link.follow()
27:    CONSTRUCT $n
28:}

29:if($t_input != ""){
30:  $result <-
31:    CONSTRUCT <BOOKLIST>
32:      WHERE <HTML>
33:          <FORM></> ELEMENT_AS $t_form
34:          </> IN "図書館蔵書検索サイト URL",
35:          <RESULT>
36:            <BOOK>
37:              <TITLE></> ELEMENT_AS $title
38:              <AUTHOR></> ELEMENT_AS $author
39:              <PLACE></> ELEMENT_AS $place
40:            </>
41:          </> IN $t_form.submit($t_input)
42:    CONSTRUCT <BOOK>
43:      $title
44:      $author
45:      $place
46:    </>
47:    </>
48:}
49:else{
50:  $result <-
51:    CONSTRUCT <BOOKLIST>
52:      No Published Book

```

```
53:      </>
54:}
```

```
55: return $result
```

## 8 システムモデル

図3に本研究で提案するシステムの概略と処理の流れを示す。

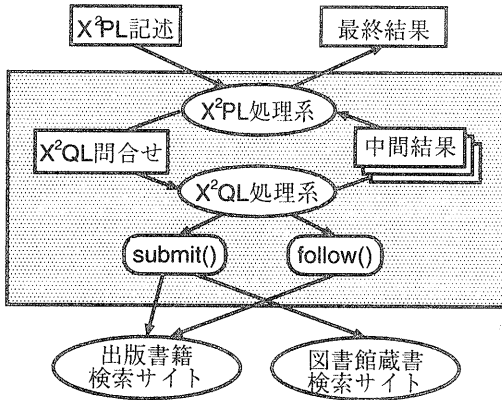


図3: システム概略

はじめに、ユーザが X<sup>2</sup>PL 記述を X<sup>2</sup>PL 処理系に与える。X<sup>2</sup>PL 処理系は、X<sup>2</sup>PL 記述から X<sup>2</sup>QL 処理系で処理を行なう部分を抽出し、X<sup>2</sup>QL 問合せを行なう。X<sup>2</sup>QL 処理系は、受け取った X<sup>2</sup>QL 問合せの処理を行なう。この処理内容に外部関数 submit(), follow() が含まれている場合、X<sup>2</sup>QL 処理系はその外部関数実装を呼び出す。これらの外部関数は、5節の説明に従って WWW アプリケーションと通信し、得られた結果を返す。X<sup>2</sup>QL 処理系は問合せ処理結果を X<sup>2</sup>PL 処理系に返し、X<sup>2</sup>PL 処理系はこれを中間結果として受け取る。この後、まだ処理が続くのであれば、同様にして X<sup>2</sup>PL 処理系が X<sup>2</sup>QL 処理系を呼ぶ。X<sup>2</sup>PL 記述の RETURN 文によって文書が返されると、それが最終結果としてユーザへ渡される。一般に、一つの統合処理を記述した X<sup>2</sup>PL 記述において、X<sup>2</sup>PL 処理系と X<sup>2</sup>QL 処理系との間での処理は数回行なわれる。

## 9 まとめと今後の課題

近年の多種多様な WWW アプリケーションに対する統合利用の要求に対して、本稿では XML 問合せ言語を用いた WWW アプリケーション統合を提案した。これを実現するため、我々の研究グループで研究開発中である X<sup>2</sup>QL と、本稿において新たに導入した X<sup>2</sup>PL を用いた。

今後の課題としては、本システムの設計と実装、実際の統合処理への適用と評価、処理記述をより容易に

するユーザインタフェースの提案等が挙げられる。

## 10 謝辞

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

## 参考文献

- [1] 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 31(11-16), 1155-1169, 1999.
- [2] D. Konopnicki, and O. Shmueli. A Comprehensive Framework for Querying and Integrating WWW Data and Services. CoopIS, 172-183, 1999.
- [3] H. Kitagawa, A. Morishima, and H. Mizuguchi. Integration of Heterogeneous Information Sources in InfoWeaver. Advances in Databases and Multimedia for the New Century - A Swiss/Japanese Perspective -, World Scientific Publishing(to appear), 2000.
- [4] H. Davulcu, J. Freire, M. Kifer, and I. V. Ramakrishnan. A Layered Architecture for Querying Dynamic Web Content. SIGMOD Conf., 491-502, 1999.
- [5] J. Clark, and S. DeRose. XML Path Language (XPath) Version1.0 (recommendation), <http://www.w3.org/TR/xpath>, 1999.
- [6] 倉垣公一, 石川佳治, 北川博之. WWW 環境におけるメタアプリケーション構築支援. 情報処理学会第60回全国大会講演論文集 (3), 83-84, 2000.
- [7] M. Dubinko, S. Schnitzenbaumer, M. Wedel, and D. Raggett. XForms Requirements (working draft), <http://www.w3.org/TR/xhtml-forms-req>, 2000.
- [8] N. Shinagawa, H. Kitagawa, and Y. Ishikawa. X<sup>2</sup>QL: An eXtensible XML Query Language Supporting User-defined Foreign Functions. Proc. 2000 ADBIS-DASFAA Symposium on Advances in Databases and Information Systems, Prague, Czech. September 2000. (to appear)
- [9] 品川徳秀, 北川博之, 石川佳治. 拡張可能 XML 問合せ言語 X<sup>2</sup>QL とその処理系, DEWS2000, 2000.
- [10] T. Kistler, and H. Marais. WebL - A Programming Language for the Web. WWW7/Computer Networks 30(1-7), 259-270, 1998.
- [11] World Wide Web Consortium. <http://www.w3.org/>.