DEIM Forum 2010 C7-5

異種XMLデータに対するファセット検索における多様な検索

駒水　孝裕†　　天笠　俊之†,††　　北川　博之†,††

† 筑波大学大学院システム情報工学研究科
†† 筑波大学計算科学研究センター
〒 305-8573 茨城県つくば市天王台 1-1-1
E-mail: †taka-coma@kde.cs.tsukuba.ac.jp, ††{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし　本稿では XML データにおけるファセット検索手法を射影演算，結合演算を含む多様な検索をサポートするための手法について議論する．先行研究で提案した XML に対するファセット検索手法では，同時に複数の XML 要素を検索することができる．しかしながら，XML データが複雑，膨大になるとそれに伴い検索される XML 要素の数も増えてしまう．これに対し，利用者の検索処理をサポートするための新たな演算を提案する．射影演算は，現在選択されているオブジェクトの中から，特定のクラスに属するオブジェクトのみを残し他は候補から落とすための演算である．一方，結合演算は現在検索されている XML 要素に対して付加情報を得るため，関連のある XML 要素を結合するための演算である．結合演算のうち，XML の木構造に基づいて結合を行う演算を構造結合，プロパティ値の条件で結合を行う演算を θ 結合という．提案する演算をプロトタイプに実装し，その有効性を検証する．
キーワード　XML，ファセット検索，異種 XML データ，情報検索

Supporting Projection and Join on a Faceted Navigation over Heterogeneous XML data

Takahiro KOMAMIZU†, Toshiyuki AMAGASA†,††, and Hiroyuki KITAGAWA†,††

† Graduate School of Systems and Information Engineering
†† Center for Conputational Sciences
University of Tsukuba
1-1-1 Tennodai，Tsukuba，Ibaraki 305-8573，Japan
E-mail: †taka-coma@kde.cs.tsukuba.ac.jp, ††{amagasa,kitagawa}@cs.tsukuba.ac.jp

Abstract　In this paper, we discuss projection and join operations for a faceted navigation method over heterogeneous XML data. Our previous proposal allows users to navigate over heterogeneous XML data. However, when the number of XML elements being processed increase and the structure becomes more complicated, it is difficult to find intended data. To solve this problem, we introduce projection and join operations in this paper. The projection operation allows users to eliminate objects other than expected ones. On the other hand, the join operation is to combine related XML elements to obtain additional information. We propose two kinds of join operations, structural join and θ-join. The structural join operation is used to combine XML elements which have structural relationships of the tree structure. On the other hand, to combine various XML elements which users give conditions for combining, we propose the θ-join operation. In this paper, we extend the prototype system to support the proposed operations.
Key words　XML, faceted navigation, heterogeneous XML data, information retrieval

1. はじめに

XML (Extensible Markup Language) は，標準のデータフォーマットとして広く普及し，膨大な情報が XML によって記述されつつある．そのため情報源の有効利用の観点から，XML 形式で記述されたデータから有用なデータを効率よく検索する技術が望まれている．

一般に，XML データに対して検索処理を行うためには

XPath [1] や XQuery [2] などのパス式に基づく問合せ言語が用いられる．しかし，これらの問合せ言語を利用するためには利用者があらかじめ検索しようとする XML の構造を知っていなければならない．XML データの形式は多数存在し，また，一つの XML データがきわめて複雑な構造を持つことも少なくないため，このような仮定をおくことは現実的ではなく，利用可能な場面が限られてしまう．

この問題に対し，情報検索の技術を応用した XML キーワード検索に関する研究がこの数年で活発に行われてきた [3]〜[5]. XML キーワード検索では，利用者はいくつかのキーワードを入力として与える．与えられた入力に対し，システムはキーワードと最もよくマッチする部分 XML データを計算し，結果として返却する．XML の構造を知らなくとも問合せが可能であるため，パス式に基づく問合せ言語よりも簡単に使うことができる．

しかしながら，XML データの検索を考える場合，利用者が常に明確な検索意図を持っているとは限らない．上記手法の場合，検索意図を問合せ式，あるいはキーワードとして表現できない限り，XML データから情報を得ることはできない．このような検索意図が曖昧な利用者をサポートするために，探索的検索手法 [6] が提案されている．

そこで我々は，探索的検索手法の一つであるファセット検索を XML データ検索に適用することを提案した [7]〜[9]. [7] では任意の XML に対してファセット検索を行うためのフレームワークを提案し, [8] では，異種 XML データに対してファセット検索を行うための諸定義と検索操作の定義を形式的に与えた．

これに対し，本研究では提案する XML のファセット検索インターフェースの機能を拡張することを目的として，これまでにサポートされていない射影演算と結合演算を提案する．射影演算は，ファセット検索で対象としている XML 要素に対して絞り込みを行う演算である．検索の中間結果に，雑多な XML 要素が入り混じっている状況で，興味の対象を絞り込むときに使う．一方，現在選択されている XML 要素に対して，それと関連を持つ別の XML 要素を検索対象とする際に使うのが結合演算である．両者の演算について形式的な定義を与え，プロトタイプシステムの実装について述べる．

本稿の構成は以下のようになる．まず第 2 節では準備としてファセット検索を紹介する．第 3 節では先行研究について説明する．続く第 4 節では本研究で追加した機能である，射影演算と結合演算を説明する．第 5 節ではシステムの設計とシステムで利用されるデータベーススキーマについて議論する．第 6 節では関連研究を紹介し，最後に第 7 節にてまとめと今後の課題について述べる．

2. ファセット検索

ファセット検索は探索的検索手法の一つで，利用者の検索行為をサポートする手法である．ファセット検索システムのインターフェースは，ファセット (facet) とキー (key) を表示する．ファセットは検索対象のオブジェクト集合を分類するカテゴリである．また，キーはファセットに対する値である．利用者は

```
<!ELEMENT books (book*)>
<!ELEMENT book (authors, title, date)>
<!ELEMENT authors (author+)>
<!ELEMENT author (first, middle?, last)>
<!ATTLIST author age CDATA #IMPLIED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT date (year, month)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT middle (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT month (#PCDATA)>
```

図 1　Document Type Definition

ファセットとキーのペアを選択することで検索対象を絞り込むことができる．その結果，選択されたオブジェクトに基づき，選択可能なファセットと，取りうる値（キー）が再計算される．この過程を，利用者が望むオブジェクトが選択されるまで繰り返す．逆に，現在の結果において目的のデータが見つけられない場合は，選択したファセットとキーのペアを取り除くことで，検索範囲を広げることができる．

3. 異種 XML データに対するファセット検索

本節では先行研究 [8] で提案した XML に対するファセット検索手法について説明する．

3.1 クラスとプロパティ

3.1.1 スキーマ木

XML データに対するファセット検索の説明をする前に，まず XML のスキーマから導出されるスキーマ木について説明する．スキーマ木は，DTD (Document Type Definition) や XML Schema [10] などの XML データの構造を表す情報を木構造で表現したものである．各ノードは要素を表し，ノード間をつなぐエッジは親子関係を表している．また，スキーマ木の各要素は濃度 (cardinality) の情報を保持している．スキーマ木において複数回出現する要素は * や + が濃度として付加されている．* は 0 回あるいは 1 回以上出現することを示しており，+ は 1 回以上出現することを表している．また，出現するかどうかがデータに依存する要素には濃度として ? が付いている．これは 0 回あるいは 1 回出現するということを示している．濃度に何もついていない要素は実際のデータ中に必ず一回出現することを表している．スキーマ木の定義を定義 1 に示す．図 1 の DTD から抽出されたスキーマ木は図 2 のようになる．

[定義 1]（スキーマ木） スキーマ木 ST は $ST = (V_s, E_s, r_s)$ の三つ組で表現される木構造である．ここで $V_s = \{v_{s1}, v_{s2}, ..., v_{sn}\}$ はノード集合を表し，$E_s = \{(v_{si}, v_{sj}) \mid v_{si}, v_{sj} \in V_s \wedge i \neq j\}$ は枝集合を表す．要素 r_s を $r \in V_s$ なる根ノードである．なお，各ノード $v_s(\in V_s)$ は濃度 $(cardinality)$ が与えられており，とりうる値は $\{*, +, ?, \emptyset\}$ のいずれかである．これを $v.card$ で参照する．

3.1.2 クラス

XML データは木構造を有するため，まず検索対象となる要

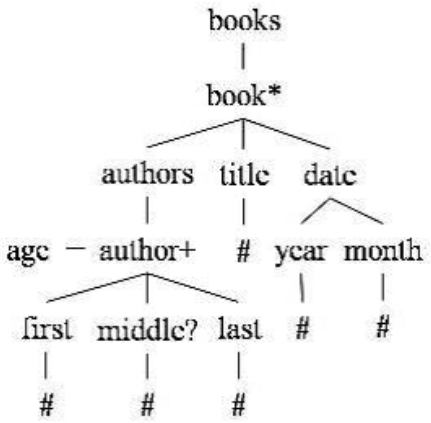

図 2　スキーマ木

素を決める必要がある．そのため，XML データのスキーマ情報の特徴を利用して検索対象となる要素を定義する．このスキーマ上における検索対象要素をクラスと呼ぶ．

XML データにおいては複数回出現する要素，すなわち，スキーマ木における * や + の付いたノードはあるまとまった情報を表していると考えることができる．このような情報の単位をクラスとして定義し，検索の対象の候補とする．定義 2 にクラスの定義を示す．

［定義 2］（クラス） スキーマ木 $ST = (V_s, E_s, r_s)$ において，クラス集合 C は以下のように定義される．

$$C = \{v \mid v \in V_s \wedge v.card \in \{*, +\}\}$$

図 2 のスキーマ木についてクラスの例を示す．定義 2 より濃度 * または + を持っている book 要素と author 要素がクラスとなる．

3.1.3　プロパティ

クラスとして指定された要素に対して，プロパティを定義する．基本的な考え方として，あるクラスのプロパティ情報は，クラスとして指定された子供あるいは子孫ノードとして存在し，そのプロパティの値としてテキスト値を含むものとする．そこで，ここではあるクラス c に対して，1) c の子要素，あるいは子孫要素であり，2) テキスト値を直接含み，3) それ自身が別のクラス要素でなく，かつ c との経路に別のクラスを含まない要素を c のプロパティとして定義する．このような要素をプロパティと呼び，その定義を定義 3 に示す．

［定義 3］（プロパティ） あるクラス $c \in C$ について，c のもつプロパティ $c.P$ は以下のように定義される．

$c.P = \{v \mid v$ は子ノードにテキスト値を持ち，c までの経路に別のクラスを含まない $\}$

ここで，$c.P$ のあるプロパティを $c.p$，$c.p$ の要素名を要素名 $c.p.name$ として参照する．なお，c からの p までのパス式を相対パス $c.p.context$ として参照する．

図 2 のスキーマ木についてとプロパティの例を示す．定義 3 により各々のクラスのプロパティは，クラス author のプロパティについては first, middle, last, age の要素がプロパティとして抽出される．また，クラス book のプロパティは title, year, month となる．このようにして図 1 で定義される XML データのクラスとプロパティが抽出される (表 1).

```
<books>
  <book>
    <authors>
      <author age="22">
        <first>Albert</first>
        <last>Holstein</last>
      </author>
    </authors>
    <title>tale</title>
    <date>
      <year>2008</year>
      <month>03</month>
    </date>
  </book>
</books>
```

図 3　XML データ例

表 1　図 1 の DTD におけるクラスとプロパティ

クラス	プロパティ
author	first, middle, last, age
book	title, year, month

3.2　オブジェクトとファセットとキー

3.2.1　オブジェクト

スキーマに対してクラスとプロパティの定義を行ったが，次にクラスに対して XML データからオブジェクトを与える．本研究ではクラスに対応する実際の部分 XML データをオブジェクトと呼び，これが検索対象となる．定義 4 にオブジェクトの定義を示す．

［定義 4］（オブジェクト） あるクラス $c \in C$ のオブジェクトとは，XML データ上でスキーマノード c で規定される部分 XML 木である．ここで，クラス C におけるオブジェクト集合を $c.O$ と定義する．

図 3 に図 2 で示したスキーマ木に対する実際の XML データの例を示した．抽出されたクラスは book と author で，これらのプロパティ集合はそれぞれ {title, year, month}, {first, middle, last, age} となっている．これらのクラスに対するオブジェクトは各々の要素以下の部分 XML データとなる．

3.2.2　ファセット

定義 4 で定義したオブジェクトの集合に対してファセット検索を適用する．これまでにオブジェクトとそのプロパティ値の定義を行ったので，ファセットおよびキーの定義は従来のものと同様である．まず，クラス集合 C に対して，そこに出現す

表 2　検索対象オブジェクト集合に対するキー

ファセット	キー
title	tale
year	2008
month	03
age	22
first	Albert
last	Holsteiin

るプロパティをファセットとして定義する．ファセットの定義は定義 5 のようになる．

［定義 5］（ファセット） 検索対象クラス C のファセット集合 F は以下のように定義される．

$$F = \bigcup_{c \in C} c.A$$

図 3 の例では，各クラスの全プロパティ集合の内 {title, year, month, first, last, age} が出現していることが分かる．これらのプロパティの集合の和集合がファセットとなる．

3.2.3　キ　　ー

ファセットに対し，オブジェクトがとる値をキーと呼ぶ．キーの定義は定義 6 のようになる．

［定義 6］（キー） ファセット $f \in F$ についてキー集合 K_f は以下のように定義される．

$$K_f = \bigcup_{o \in c.O s.t. c \in C' \wedge f \in c.A} o.f.value$$

図 3 の例のファセットに対してテキスト値 {tale, 2008, 03, Albert, Holstein, 22} がある．これらが各ファセットに対するキーとなる．図 3 のデータにおけるファセットとキーは表 2 のようになる．

3.3　検索処理 -選択演算-

次にこれらの定義を用いた検索処理について説明する．ここでの検索は通常のファセット検索と同じようにファセットとキーのペアを選択することによって行われる．ある値が選択されるとそれに応じて選択可能なファセットとキー，選択中のオブジェクト一覧が更新され，利用者はこれに対してファセット，キーを指定する．この過程を繰り返すことによって，徐々に絞り込んでいく．ここでファセットとキーの選択処理を選択演算と呼ぶ．この選択演算の定義を定義 7 に示す．

［定義 7］（選択演算） オブジェクト集合 O に対する選択演算は以下のように定義される．なお，S は入力でファセットとキーのペアの集合である．

$$\sigma_S(O) = \{o \mid (f,k) \in S \wedge o \in O \wedge o.f.value = k\}$$

先行研究では，上記の定義に基づいて効率的な検索処理を実現するためのデータベース構成についても議論している，構成の詳細については後述する．

表 3　book オブジェクトの例

id	title	year	month
12	Facet story	2009	03
16	Story of Key	2005	04
23	Faceted tale	2007	06

表 4　author オブジェクトの例

id	first	middle	last	age
25	Albert	M	Facet	56
29	John	R	Bit	33
34	Mick	J	Python	32

4.　XML データに対するファセット検索における射影演算と結合演算

先行研究で提案した枠組みでは，検索対象となるオブジェクトには複数のクラスが含まれる．例では，book と author がクラスであったので，それぞれに対応するオブジェクトが検索対象となる．一般的にある XML に含まれるクラスの数はこれより多い場合も少なくない．さらに，異なるスキーマに基づく異種 XML データを扱う場合，その数はもっと増えてしまう．この問題に対処するために，本研究では，検索対象のオブジェクトをクラスで絞り込む射影演算を提案する．

また，author について絞り込みを行ったとき，その著者に関する book 情報を得たい場合や，その逆のケースも考えられる．このときは，XML データ上で関連を持つ別のクラスを同時に提示することで，利用者の要求に答えることができると考えられる．このために結合演算を提案する．本節ではこれらの射影演算と結合演算について説明する．

4.1　射 影 演 算

射影演算はオブジェクト集合に対してクラスを選択し絞り込みを行う．まず，利用者はいくつかのクラスを選択する．これをクラス集合 C' とする．選択されたクラスのオブジェクトの和集合が結果一覧となり，それに伴いファセットとキーが更新される．利用者はこれに対し，選択演算や射影演算をさらに行うことができる．射影演算はで定義 8 のように定義される．

［定義 8］（射影演算） オブジェクト集合 O に対する射影演算は以下のように定義される．なお，C' は選択されたクラスの集合である．

$$\Pi_{C'}(O) = \{o \mid c \in C' \wedge o \in c.O\}$$

射影演算の例として， book オブジェクトと author オブジェクトを考える．それぞれが表 3，表 4，表 5 のように与えられたものとする．id はオブジェクト ID を示しており，下線の引かれた各プロパティはそれぞれのオブジェクトのファセットを意味している．この時に以下の二つの問い合わせのケースを考える．

（1） 本 (book) に関する情報が欲しいが著者 (author) は必要ない

（2） 本 (book) に関する情報とその著者 (author) の情報

を知りたい
これらの問合せがどのように表現されるかを示す．なお，検索結果を ID のみで表現している．

（1） 射影演算のパラメータとして book を指定することで book 要素のみを検索対象とすることができる．その結果は book 要素のオブジェクト集合のみが選択される．

$$\Pi_{\{book\}}(O) = \{12, 16, 23\}$$

（2） 複数のクラスを指定することもできる．その場合得られる結果は， book オブジェクトと author オブジェクトの和集合となる．

$$\Pi_{\{book,author\}}(O) = \{12, 16, 23, 25, 29, 34\}$$

4.2 結合演算

複数の XML 要素を結合する場合として以下の二つの状況が考えられる．

（1） XML データ上で構造的な関係にあることなる二つのクラスを結合する

（2） XML データ上の構造的な関連に関わらず，ある特定のプロパティの値によって異なるクラスを結合する

4.2.1 構造結合

XML データ上で，先祖子孫関係または兄弟関係にある二つのクラスを結合する．ここではこれを構造結合と呼ぶ．構造結合の定義を以下に示す．

［定義 9］（構造結合） 二つのクラス c_1, c_2 に対する構造結合は以下のように定義される．

$$O_1 \bowtie_{ST} O_2 = \{(o_1, o_2) \mid o_1 \in O_1 \wedge o_2 \in O_2$$
$$\wedge o_1 \text{と } o_2 \text{が } XML \text{ 上で先祖子孫関係または兄弟関係を持つ }\}$$

表 3 と表 4 における book 要素と author 要素の構造結合の例を示す．例にあるように book 要素は author 要素の先祖である．これらのクラスを結合する場合，以下のように記述する．

$$\{12, 16, 23\} \bowtie_{ST} \{25, 29, 34\}$$
$$= \{(12, 29), (12, 34), (16, 25), (23, 25), (23, 34)\}$$

このとき， book 要素のオブジェクトと author 要素のオブジェクトを結合すると表 6 のような結果が得られる．

4.2.2 θ 結合

構造結合では，選択されているオブジェクトに対して，構造的な関係をもつクラスを求めたが，取得したい情報は必ず先祖または子孫にあるとは限らない．例えば，author 要素に対して，別の XML データに含まれるより詳細な情報を得たい場合には，構想結合では不十分である．

そのため，ここでは θ 結合演算を定義する．これは関係代数演算と同様に，プロパティに関する条件を指定した上で，その条件を満たすオブジェクトのペアを求める演算である．このとき，結合に使うプロパティと値に関する条件は利用者が与えるものとする．

表 5 award オブジェクトの例

id	book-title	name
45	Facet story	Facet award
46	Bit Story	Bit award
57	Faceted tale	Facet award

表 6 構造結合の結果

title	year	month	first	middle	last	age
Facet story	2009	03	John	R	Bit	33
Facet story	2009	03	Mick	J	Python	32
Story of Key	2005	04	Albert	M	Facet	56
Faceted tale	2007	06	Albert	M	Facet	56
Faceted tale	2007	06	Mick	J	Python	32

表 7 等結合の結果

title	year	month	book-title	name
Facet story	2009	03	Facet story	Facet award
Faceted tale	2007	06	Faceted tale	Facet award

［定義 10］（θ 結合） 二つのオブジェクト集合の θ 結合は以下のように定義される．

$$O_1 \bowtie_{\{f_1 \theta f_2\}} O_2 = \{(o_1, o_2) \mid$$
$$\wedge o_1 \in O_1 \wedge o_2 \in O_2 \wedge o_1.f_1.key \theta o_2.f_2.key\}$$

表 3 と表 5 における book 要素と award 要素の等結合の例を示す．ここでは，章 (award) を取った事のある本を取得するために等結合を適用する．book 要素の title と award 要素の book-title が一致するデータを検索する．このような等価条件による等結合は以下のように表現される．

$$\{12, 16, 23\} \bowtie_{title=book\text{-}title\}} \{45, 46, 57\}$$
$$= \{(12, 45), (23, 57)\}$$

この結合結果は表 7 のようになる．このようにすることで複数のオブジェクトを等価条件によって結合することができる．

5. ファセット検索システムの概要

本節では先行研究で提案したシステムに本研究で提案した射影演算と結合演算を取り入れたシステムについて説明する．始めに全体の概要として検索システムの概要を各コンポーネントについて述べる，その後にシステム内で利用されるデータベースのスキーマについて説明する．

5.1 検索システムの概要

図 4 に検索システムの概要を示す．

XML Database 検索結果として必要なデータが格納されたデータベースである．XML Database には検索対象となるすべての XML データがあらかじめ格納されているものとする．

Facet Database 検索に必要な情報を格納するデータベースで関係データベースを用いる．関係データベースを用いることでキー毎の検索可能オブジェクト数の計算等の集約の効率的な実行を期待できる．Facet Database として XML データベースを用いない理由としては，XML データベースがサポートす

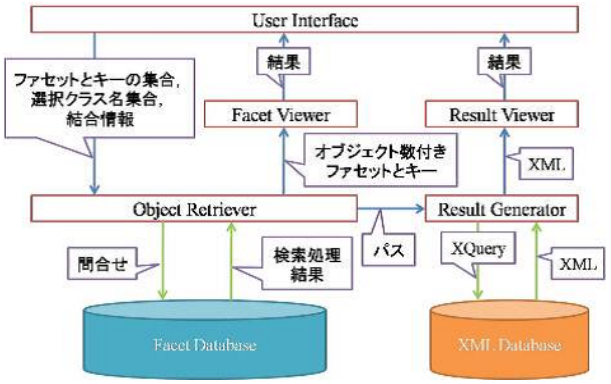

図 4　システム概要

る XQuery では，集約演算が定義されていないことと，大規模なデータに対する性能が関係データベースと比較してあまり高くない点が挙げられる．データベースのスキーマについては次節にて説明する．

User Interface 利用者とのインタラクションを行う．利用者に対し，使用可能なファセットとキーを提示し，問合せに関する情報を Object Retriever に伝える．この情報には選択されたファセットとキーのほかに射影演算で用いられるクラス選択の情報，結合演算で用いられる各クラスと θ 結合の場合は結合条件が含まれる．また，これまでに選択してきたファセットとキー，クラスを表示し，利用者の検索をサポートする．

Object Retriever 各演算を取り扱いオブジェクトを絞り込む．Object Retriever の入力はファセットとキーのペアの集合，選択されたクラス名の集合，結合に関する情報である．選択演算では選択されたファセットとキーを基にオブジェクトの絞込む．射影演算では選択されたクラスに属するオブジェクトに絞り込む．結合演算では選択されたクラスの先祖子孫関係または兄弟関係にあるクラスを提示する．この段階で利用者がクラスを選択すればこの二つのクラスのオブジェクト集合を構造結合する．もし，選択しない場合はすべてのクラスを表示し，任意のクラスを選択させる．その後この二つのクラスにおけるファセットの候補をすべて表示し，結合条件を入力させる．これに基づきこの二つのクラスのオブジェクト集合の結合を行う．これらの演算ののちにキー毎，ファセット毎の検索可能オブジェクト数の計算を行う．入力について SQL を生成し，ファセットとキーの検索可能オブジェクト数付きリストと結果オブジェクトを取得するためのパス式集合を獲得する．その各々を Facet Viewer と Result Generator に送信する．

Result Generator 送信されてきたパス式集合を基に XQuery を生成し結果を取得する．その後その結果である XML データを Result Viewer に送る．

Facet Viewer 送信されてきたファセットとキーの検索可能オブジェクト数つきリストを利用者の見やすい形に変換する．単純な例としては HTML のリストを用いてファセットの下にキーを表示することが考えられる．このようにファセット検索システムを構築する人物の表示したい形に変換できるため，データ毎に異なる利用者像に合わせたシステムを構築できる．

表 8　データベーススキーマ

概念	スキーマ
クラス	クラス（クラス ID，クラス名，パス式）
プロパティ	プロパティ（クラス ID，プロパティ名，パス式）
オブジェクト	オブジェクト（オブジェクト ID，クラス ID，パス式）
ファセット	ファセット名（キー，クラス ID，オブジェクト ID）
構造関係	構造関係（クラス ID-1，オブジェクト ID-1， クラス ID-2，オブジェクト ID-2）

ここで変換されたファセットとキーのリストを User Interface に渡して，利用者に再度表示する．

Result Viewer 基本的な機能は Facet Viewer と同じである．Facet Viewer との違いは入力が XML データで与えられるため，XSLT [11] を用いての変換ができることである．これはすなわち，利用者が取得したい形に変換できるということである．ここで変換された検索結果を User Interface に渡し，利用者に表示する．

5.2　データベーススキーマ

本節では本システム（図 4）で用いる Facet Database のデータベーススキーマを紹介する．

Facet Database への問合せとしては次の二種類がある．一つ目は検索結果 XML データ取得用のパス式を取得する問合せで，もう一つは絞り込まれたオブジェクトのファセットとキーの算出の問合せである．二つ目の問合せでは単純な検索に加えて，各ファセットとキーが検索できるオブジェクト数を算出しなければならない．そこで，これらを実現可能なデータベーススキーマを考察する．

各スキーマは表 8 のようになっている．クラス，プロパティ，オブジェクト，ファセット のスキーマは [8] にて定義されたものを紹介する．本稿ではこれらに加えて構造結合演算のための概念 構造関係 を追加した．

まずクラスのテーブル名は クラス で，各々のクラスに ID を付加する．クラス名は主キーにはなり得ない．これは異種 XML データを扱う際に，複数の DTD において同一の要素名を持つものの異なる内容モデルを持つ要素が複数存在し得るからである．このような観点から同じ名前を持ったクラスでも識別できるように クラス ID を用いる．また，同スキーマ内の パス式 は検索結果 XML データを得るために使用するものである．

プロパティテーブルはその名前をプロパティとし，所属しているクラスのクラス ID とプロパティ名を主キーとする．これは同一クラス内には同じ名前のプロパティは存在せず，他のクラスに同一名のプロパティが存在するためである．また，このスキーマの パス式 はクラスの要素からの相対パス式で表現される．

次にオブジェクトテーブルはテーブル名 オブジェクト とし，すべてのオブジェクトを管理する．これまでのテーブルはデータ構造に関する情報を保持してきたが，このオブジェクトテーブルとこの後に紹介するファセットは実データが入る．このスキーマではオブジェクトを一意に識別するために オブジェクト ID と クラス ID の組合せを主キーとしています．ここでの パ

ス式は 実データへのアクセスのためのパス式で，基本的にはクラスが持つパス式と同じだが，以下のようにデータの場所がより詳細に記されている．

$//book/author$[25]

その中の book 要素の子要素の author 要素の 25 番目の要素を指示している．

続いてファセットテーブルを説明する．このスキーマはテーブル名をファセット名としてあり，ファセット毎にテーブルを構築する．このスキーマを構成する，キー，クラス ID，オブジェクト ID の組合せが主キーとなる．これは，あるオブジェクトのファセットを見たときにそのオブジェクトが複数のキーを持つこと可能性があることによる．例えば，買い物のカテゴリを考えると商品がオブジェクトでジャンルがファセットであると考えられる．このジャンルにおいて，複数のジャンルにまたがる商品が考えられる．具体的には，ダイエットに関する本を考えると，この本は ダイエット・健康 というジャンルと本というジャンルの両方に属する．この場合ジャンルというファセットに二つのキー（ダイエット・健康，本）があることになる．このような場合が想定されるため，表 8 のようなスキーマになっている．

構造関係ではオブジェクト同士の構造的な関係を表すために二つのオブジェクトのプライマリーキーの組み合わせで表現されている．これ利用することで必要なファセットの計算等を行うことができる．なお．今回はこのような単純なやり方を採用したが，この方法では構造情報を保持するのに膨大なディスクスペースを使用してしまう．それに対し，構造的な孫関係を計算する方法としてはこれ以外にも XML データ上での前順序，後順序を利用する方法，Dewey order を利用する方法など，ディスクスペースが小さいものも存在する．これらの方法とのディスクスペースや検索速度の比較は今後行っていきたいと考えている．

6. 関 連 研 究

本節では，本研究に関連する研究を紹介する．

RDF を対象としたファセット検索 [12]
Oren らは [12] においてファセット検索を RDF データに適用しており，本稿とは扱うデータが異なる．また，Oren らはファセット検索におけるファセットの順番の重要性について述べ，ファセットの順序付けに対する指標を提案し，その有用性を検証した．ファセットの順序付けに関しては先行研究である [7] で [12] の手法を拡張したものを用いていたが，計算にかかるコストが高く速度面に問題があると結論付けていた．そのため，ファセットの順序付け手法関しては別の議論が必要であると考えられる．

大規模ファイルシステムにおけるファセット検索 [13]
Koren らは [13] にて，ペタバイト級の大規模ファイルシステムの検索にファセット検索を用いることを提案していた．近年ペタバイト級のファイルストレージを用いることが増つつあることに着目し，[13] ではファイルシステムで検索する際に利用者がはっきりとした意図で適切な問合せを表現することはできないと述べている．この点への改善策としてファセット検索を適用していた．このことから，大規模データの検索に対する手法としてファセット検索が有用であることがうかがえる．ファイルシステムにおける検索対象はすべてのファイルであるが，本稿では XML データを対象にしているために検索対象が異種のものになるという点で扱うデータが異なっている．

ファセット検索のためのフレームワークの研究 [14]
[14] では Tzitzikas らによって *FleXplorer* 提案された．構造データと非構造データに対するファセット検索のミドルウェアとしてのフレームワークが提案されており，検索の高速化を目標としている．またファセットの階層構造を扱うための手法として階層構造を記憶するためのテーブルを用いている．本研究とは階層構造を考慮するという点は共通であるが，その手法としてテーブルを用いるという点で本研究とは異なる．また，この研究ではファセットとキーの抽出については触れられておらず，この点に関しても本研究とは異なる．

7. ま と め

本研究では [8] の手法を拡張し，多様な検索ができる手法を提案した．基本的なクラス，プロパティ等の定義は [8] に従い，本研究ではこれらを基に射影演算，結合演算に形式的な定義を与えた．射影演算は様々な XML 要素が混在する中で利用者が興味の対象とする XML 要素を選択できるようにする演算である．一方，結合演算は現在選択されている結果と構造的あるいは値的な関連のある XML 要素を同時に検索できるようにする演算である．これらの演算により，ファセットとキーのペアを選択するといった単純な選択演算だけでなく，注目する検索対象への絞り込みや他の関連する XML 要素と同時検索を可能にし，多様な検索を実現した．また, [8] のシステムを拡張し，これらの演算を利用可能にするプロトタイプシステムの実装についても紹介した．

今後の課題としては，本稿で拡張した機能を含むシステムの実装を行い，実在のデータに対して検索を行い，パフォーマンスを計測し評価を行う．また，現状の Facet Database に採用するデータベースについて関係データベースだけでなく他のデータベースを用いることも視野に入れて検討する．本稿でのオブジェクトの構造的な関係を計算するための機構について，他の手法との比較，考察を行う．さらに，画面の都合上ファセットを表示できる数に制限があるため，表示するファセットを自動で選択するため，あるいはファセットの順序付けを行う手法についても検討していきたいと考えている．

謝辞 本研究の一部は科学研究費補助金特定領域研究（B）（＃21013004）と科学研究費補助金挑戦的萌芽研究（＃21650017）と科学研究費補助金若手研究（B）（＃21700093）による．ここに記して謝意を示す．

文　　献

[1] “XML Path Language”. http://www.w3.org/TR/xpath/.
[2] “XML Query Language”. http://www.w3.org/TR/xquery/.
[3] G. Li, J. Feng, J. Wang, and L. Zhou, “Effective keyword search for valuable lcas over xml documents,” CIKM,

eds. by M.J. Silva, A.H.F. Laender, R.A. Baeza-Yates, D.L. McGuinness, B. Olstad, Ø.H. Olsen, and A.O. Falcão, pp.31–40, ACM, 2007.
[4] Z. Liu and Y. Chen, “Identifying meaningful return information for xml keyword search,” SIGMOD Conference, eds. by C.Y. Chan, B.C. Ooi, and A. Zhou, pp.329–340, ACM, 2007.
[5] A. Termehchy and M. Winslett, “Effective, design-independent xml keyword search,” CIKM, eds. by D.W.-L. Cheung, I.-Y. Song, W.W. Chu, X. Hu, and J.J. Lin, pp.107–116, ACM, 2009.
[6] Ryen W., White, B. Kules, Steven M., Drucker, and M.C. Schraefel, “Supporting exploratory search,” Introduction, CommuSupporting Exploratory Senications of the ACM, vol.49, no.4, pp.36–39, 2006.
[7] 駒水孝裕，天笠俊之，北川博之，“XML データに対するファセットナビゲーションのためのフレームワーク FoX の提案,” DEIM, pp.B7–6，2009.
[8] 駒水孝裕，天笠俊之，北川博之，“異種 XML データに対するファセット検索手法の提案,” 情報処理学会研究報告「デジタルドキュメント (DD)」，vol.2009，pp.●●–●●，2009.
[9] 天笠俊之，石井理修，吉江友照，建部修見，佐藤三久，“XML データを対象としたファセット検索インターフェースの生成,” 情報処理学会研究報告「デジタルドキュメント (DD)」，vol.2008, no.53，pp.7–13，2008.
[10] “XML Schema 1.1”. http://www.w3.org/XML/Schema.
[11] “XSL Transformations”. http://www.w3.org/TR/xslt.
[12] E. Oren, R. Delbru, and S. Decker, “Extending faceted navigation for RDF data,” International Semantic Web Conference, eds. by I.F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, vol.4273, pp.559–572, Lecture Notes in Computer Science, Springer, 2006.
[13] J. Koren, A. Leung, Y. Zhang, C. Maltzahn, S. Ames, and E.L. Miller, “Searching and navigating petabyte-scale file systems based on facets,” PDSW, ed. by G.A. Gibson, pp.21–25, ACM Press, 2007.
[14] Y. Tzitzikas, N. Armenatzoglou, and P. Papadakos, “FleXplorer: A framework for providing faceted and dynamic taxonomy-based information exploration,” DEXA Workshops, pp.392–396, IEEE Computer Society, 2008.