XML データ表現を考慮した関係データベースのキーワード検索

張　麗茹† 　大森　匡† 　星　守†

†電気通信大学大学院情報システム学研究科 〒182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail: †{zhangliru, omori}@hol.is.uec.ac.jp

あらまし 近年，関係データベースのような構造化データベースや XML データベースにおいても，問い合わせ言語の代わりにキーワード入力による検索方法が提案されている．本研究は XML データの格納を許した関係データベースのキーワード検索方式を提案する．まず，XML データベースと関係データベースにおける従来研究のキーワード検索方式の機能比較を行い，XML に相当する情報も格納するとき，両者が共に不十分な解しか探せないことを示す．次に，これを解決するため，ハイブリッド型 XML-関係データベースにおいてキーワード検索を行うアルゴリズムを提案する．

キーワード ハイブリッドデータベース，キーワード検索，XML

Keyword-based Search for Hybrid XML-Relational Databases

Liru ZHANG† 　Tadashi OHMORI† 　and 　Mamoru HOSHI†

† The University of Electro-Communications, Graduate School of Information Systems, Chofugaoka 1-5-1 Chofu, Tokyo, 182-8585 Japan
E-mail: †{zhangliru, omori}@hol.is.uec.ac.jp

Abstract In recent years, keyword-based search has been suggested to be applied instead of query-language based search for Relational Databases and XML Databases. In this paper, we propose a new method of keyword-based search for a Hybrid database, the Relational Database allowing XML data format. Firstly, we analyze the ability of keyword-based search for XML Database and traditional Relational Databases and clarify that the results of keyword-based search on each system are defective. Next, we propose a method of keyword-based search for Hybrid XML-Relational Databases and explain that better results are obtained in the Hybrid system.

Key words Hybrid Database，keyword search，XML

1. はじめに

近年，キーワード入力による検索に関する研究はよく注目され，エンタープライズサーチエンジンの土台とする関係データベースにおけるキーワード検索方法も提案された（DBXplorer[1]）．一方，データ交換に優れている XML データが広く利用されつつある．XML データベースにおけるキーワード検索の手法も研究されている（XRANK[2]）．

XML データ表現を考慮して管理するシステムは既存の手法としてピュアな XML データベースで処理することか伝統的な関係データベースにマッピングして格納，管理することになる．しかし，XML データベースと関係データベースシステムの構築方法が根本的に違って，検索アルゴリズムも異なる．XML データベースと関係データベースでのキーワード検索結果は一致しているかどうかまだ知られていない．

本研究はキーワード検索の機能分析に焦点を当てて，まず，XML データベースと関係データベースでキーワード検索の機能分析する．次に，ハイブリッド XML-関係データベースが XML データ型と従来の関係データベースのデータ型両者を格納したときにキーワード検索を行うアルゴリズムを提案する．

2. 各システムのキーワード検索機能の分析

XML データの格納や検索など管理したいときには典型的に二種類手法がある．一つは XML データベースシステムである．XML データがそのままツリーの形で格納，管理される．二つ目は機能的に強い関係データベースで，実体(Entity)と関連(Relationship)に対応する XML データの情報をマッピングし，格納，管理するシステムである．

2.1. XRANK システム[2]によるキーワード検索

XML データベースにおけるキーワード検索から分析してみる．図 1(a)，(b)で示す XML 例 1 と XML 例 2 は論文システムの簡単な XML データが例である．例 1 の XML データは，Conference 単位で“会議―セッション―論文”の XML 階層表現で，例 2 の XML は Author 単位で“著者―論文”の XML 階層になる．

図 1 の(a)例 1 の XML データは表 1 の DTD の定義に従って，XML ツリー構造は図 2，インスタンスは図 3 のようになって，

同様に図 1 の(b)例 2 の XML データは表 2 の DTD の定義に従って, XML ツリー構造が図 4, インスタンスは図 5 のようになる.

Conference—XML例1のDTD定義 :

```
<!ELEMENT  Conference  (Conf * )>
<!ELEMENT  Conf  (C_title,  Session+ )>
<!ATTLIST  Conf  cid  CDATA  #REQUIRED >
<!ELEMENT  C_title  (#PCDATA )>
<!ELEMENT  Session  (S_title,  Paper+ )>
<!ATTLIST  Session  sid  CDATA  #REQUIRED >
<!ELEMENT  S_title  ( #PCDATA  )>
<!ELEMENT  Paper  ( P_title,  keywords* )>
<!ATTLIST  Paper  pid  CDATA  #REQUIRED >
<!ELEMENT  P_title  ( #PCDATA  )>
<!ELEMENT  keywords  ( #PCDATA  )>
```

表1 XML例1ーDTD

Author—XML例2のDTD定義 :

```
<!ELEMENT  Authors  (author * )>
<!ELEMENT  author  (name,  paper+ )>
<!ATTLIST  author  aid  CDATA  #REQUIRED >
<!ELEMENT  name  (#PCDATA )>
<!ELEMENT  paper  ( p_title,  keywords* )>
<!ATTLIST  paper  pid  CDATA  #REQUIRED >
<!ELEMENT  p_title  ( #PCDATA  )>
<!ELEMENT  keywords  ( #PCDATA  )>
```

表2 XML例2ーDTD

```
<Conference  cid="V04">
    <C_title>VLDB1998 </C_title>
     <Session  sid="S001">
      <S_title>Test and ・・・</S_title>
    <Paper>
        <title>・・・</title>
        <author>・・・ </author>
        <keywords>・・・</keywords>
        <cite>・・・ </cite>
      </Paper>
   …
  </Session>
     …
</Conference>
```

(a) XML 例 1 -Conference

```
<Author name="Sanjay Agrawal">
  <Paper  id="P003">
       <title >・・・</title>
       <keywords>・・・</keywords>
     <conference>・・・ </conference>
  </Paper>
  <Paper  id="P004">
       <title >・・・  </title>
       <keywords>・・・</keywords>
     <conference >・・・ </conference>
  </Paper>
    …
</Author>
```

(b) XML 例 2 –Author

図 1 XML ドキュメント(a)例 1 (b)例 2

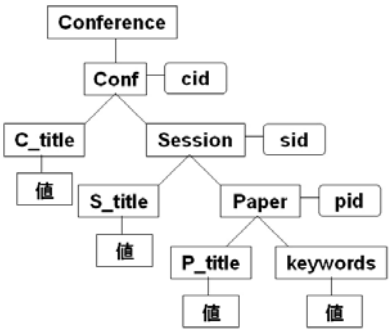


図 2 XML 例 1 DTD によるツリー構造

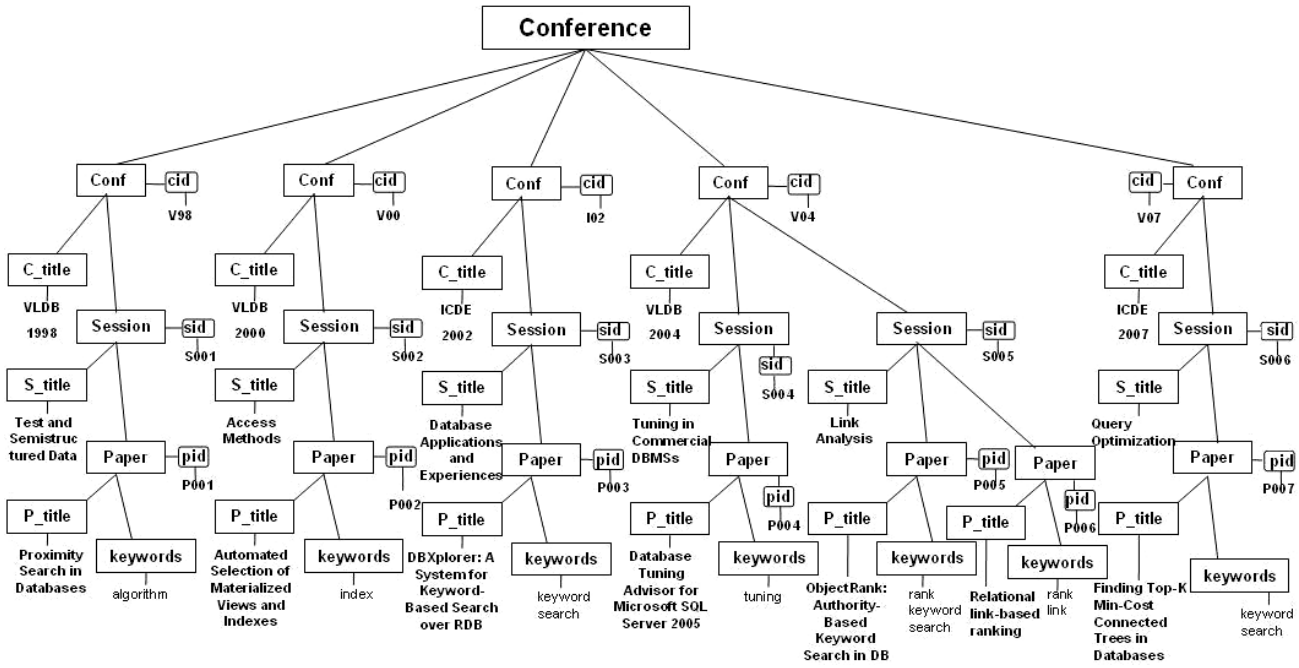


図 3 XML 例 1 の実例表現ツリー--- XML-Tree1

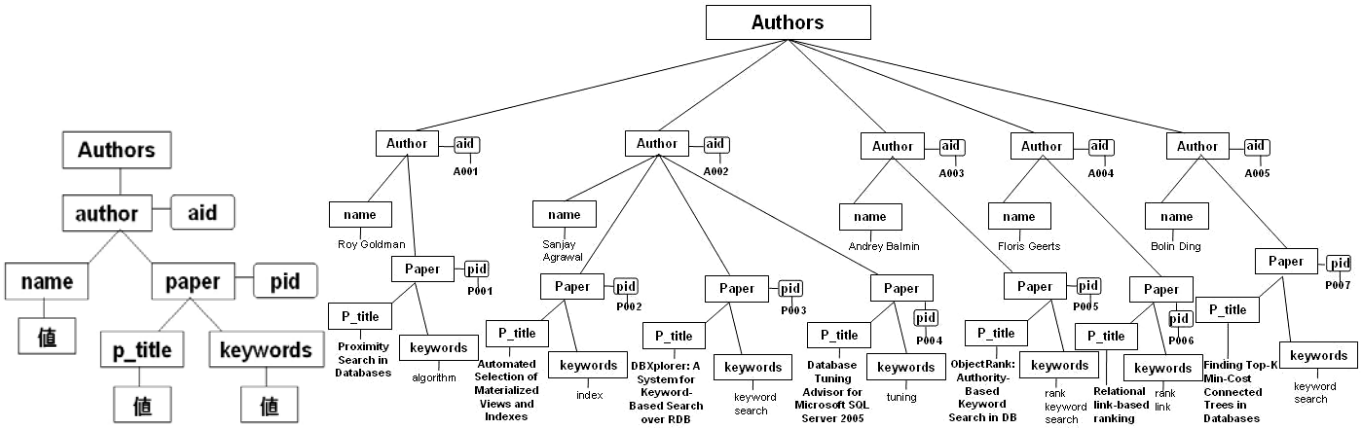
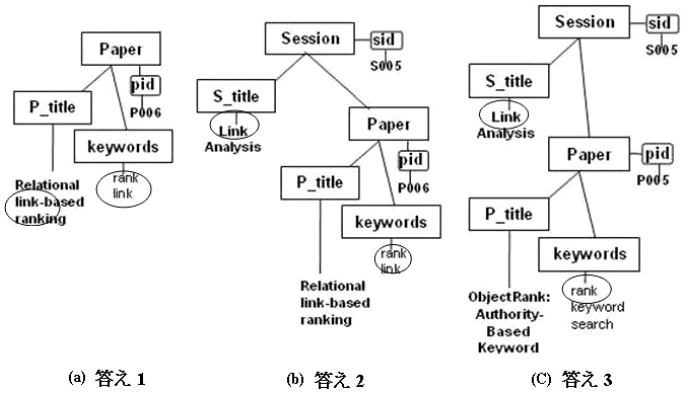


図 4 例 2 DTD によるツリー構造

図 5 XML 例 2 の実例表現ツリー---XML-Tree2

XRANK[2]システムにおいで Conference による階層を持つ XML-Tree1(図 3)に対して，$Query = \{rank, link\}$ の場合（つまりクエリが rank, link の場合）は一番目の答えが図 6(a)のような subtree の Paper(pid: P006)になる．（図 6 では○で囲んでいる所は keyword を示す. 本稿ではすべての図の○は同じ意味で用いる．）



(a) 答え 1　　(b) 答え 2　　(C) 答え 3

図 6　$Query = \{rank, link\}$ 答え

二番目の答えは図 6(b)のような subtree の Session(sid:S005)と Paper(pid:P006)になる．三番目の答えは図 6(c)のような subtree の Session(sid:S005)と Paper(pid:P005)になる. 4 番目の答えは図 7 のような subtree の Session(sid:S005)を通って Paper(pid:P006, pid:P005)になる．

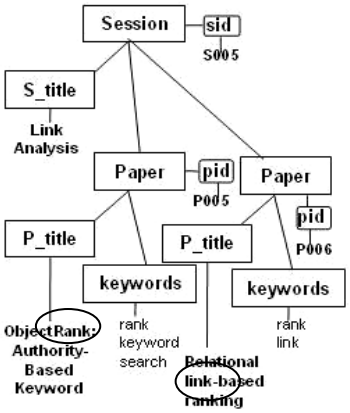


図 7　$Query = \{rank, link\}$ 答え 4

XRANK[2]システムでキーワードがそれぞれ子孫ノードのテキストでヒットしたら，これらの子孫ノードの共通先祖ノードを探す．一番近い先祖ノードを subtree の根ノードとして得る．

しかし，図 7 の答え Paper : P005 と P006 には引用関係があるなら，関連性が高いと判断されるべきだが，XRANK の手法でランキングが一番低い答えになる．

XRANK[2]システムでは, キーワードがヒットする部分がサイズ的に（Height と Width による）一番小さい subtree を答えとして優先的に出す．例えば，図 6(a)の答えは図 6(b)の答えよりランクが高くなる．図 6(b)の subtree の高さは図 6(a)より高く，○印のノード間の距離は遠いからである．同様に図 7 の subtree は横幅があって，図 6(b)(c)より○印のノード間の距離は遠いである．

ここで，検索結果の妥当性に注目したい．例えば，図 3 の XML-Tree1 の実例表現に対し，$Query = \{system, analysis\}$（二つのキーワード“system”と“analysis”の AND 検索）を与える．“system”は PaperP003 にヒットし，“analysis”は SessionS005 にヒットする．XRANK の手法では，P003 と S005 の関係による検索結果が subtree にならないので, 答えにはならない. しかし，S005 の下 P005 が P003 と引用関係があるなら，図 8 のように subtree の結果を出すべきである．

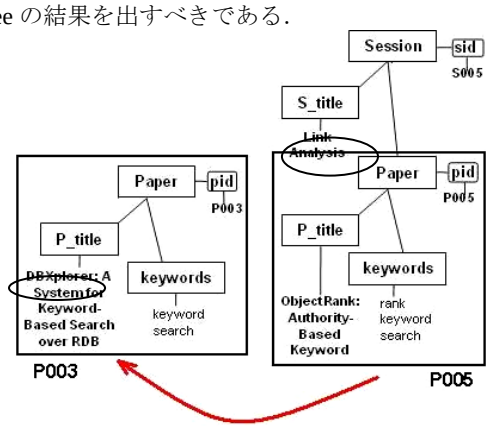


図 8　$Query = \{system, analysis\}$ の結果

XRANK システム自体は Citation のような関係性は XLink で書くが XRANK の答えとしては考えない．この原因で XRANK のキーワードによる検索結果は不完全である．

2.2. DBXplorer[1]システムによるキーワード検索

DBXplorer[1]システムの手法に従って, 関係データベースにおけるキーワード検索を分析する．

図 9 は関係データベースのスキーマは前例図 3，図 5 と同じ XML データ情報に基づいて設計してある．前例の XML ドキュメントは関係データベースにマッピングし格納した形である．

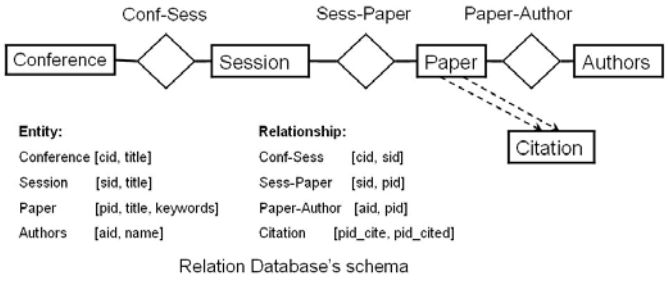


図 9　RDB schema

図 9 のデータモデルで，Conference の XML 階層情報は実体の Conference，Session，Paper と関連の Conf-Sess，Sess-Paper に分けて，格納された．

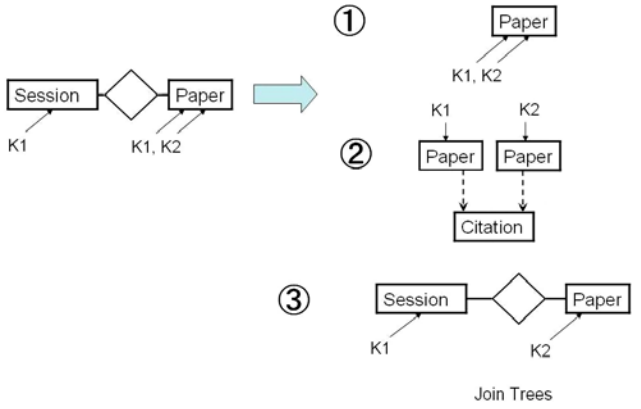


図 10　$Query = \{K1, K2\}$　Join trees

図 10 はキーワード K1,K2 を検索するとき，得られる答えを JoinTrees（問合せの手順式）①②③で表現してある．例えば，K1，K2 が Session と Paper でヒットする場合は，3 種類結果が出る．ここで，Citation の関係性は直接引用関係とする．

Join tree①の場合は次の SQL 文になる：

SELECT * FROM Paper
WHERE title= '%K1%' & '%K2%'

これは，キーワード K1 と K2 が同じペーパーでヒットすることを意味する．（ここの SQL 文は意味を表しているだけで，厳密な言語表現ではない．）

Join tree②の場合は次の SQL 文になる：

SELECT * FROM Paper P1, Paper P2, Citation C
WHERE P1.pid =C. pid_cite &
P2.pid =C. pid_cited &
P1.title= '%K1%'& P2.title= '%K2%'

つまり，キーワード K1 と K2 がそれぞれのペーパーでヒットし，この二つのペーパーの引用関係があることを意味する．

Join tree③の場合は次の SQL 文になる：

SELECT * FROM Paper, Session, Sess-Paper
WHERE Paper.pid =Sess-Paper.pid &
Session.sid =Sess-Paper.sid &
Session.title= '%K1%' & Paper.title= '%K2%'

つまり，キーワード K1 と K2 がそれぞれペーパーとセッションにヒットし，この二つのエンティティの関連関係があると意味する．

JoinTree を用いる DBXplorer[1]システムは XRANK[2]システムよりよい結果が得られる．例えば，Citation の関係性を展開すると図 11 のような結果を得ることが可能である．図 11 は Citation の引用関係で P1 を通って，S1 と P2 がつながった結果である．

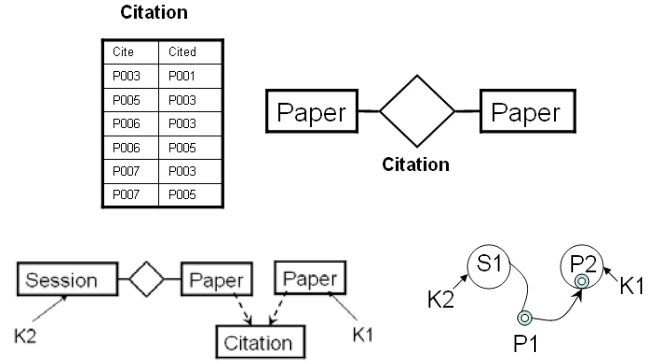


図 11　$Query = \{K1, K2\}$　K1 in P2,　K2 in S1

XRANK[2]システムは Citation の関係性を subtree で表せないので，この答えは得られなかった．DBXplorer[1]システムでは，図 11 に示した Paper 間の引用関係があれば，キーワード K1,K2 の関係性は求められるはずである，従って，DBXplorer[1]システムで答えが得られる．

以上は関係データベースで検索可能な場合のキーワード検索機能を分析した．次は検索不能の場合を説明する．

関係データベース DBXplorer[1]では必要な階層関係が得られないことがある．図 12 は $Query = \{Tuning, Analysis\}$ の場合である．キーワードがそれぞれのタプル S004 と S005 でヒットするので，セッション S004 と S005 は関係がないと判断され，結果は出ない．

SID	Title
S001	Text and Semistructured Data
S002	Access Methods
S003	Database Applications and Experiences
S004	Tuning in Commercial DBMSs
S005	Link Analysis
S006	Query Optimization, Database Applications and Experiences

図 12　$Query = \{Tuning, Analysis\}$ in Entity "Session"

しかし，SessionS004 と SessionS005 は同じ大会である．そのため，図 13 は XRANK[2]で検索結果の Subtree になるが，DBXplorer[1]システムでは検索不能である．

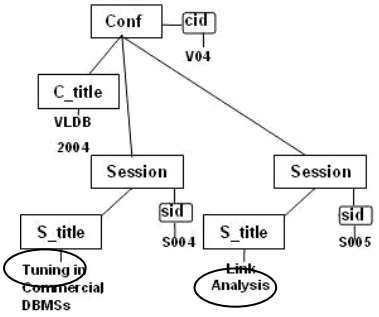


図 13　$Query = \{Tuning, Analysis\}$ の答え

3. ハイブリッド型の XML-RDB システムの提案

3.1. 目的

前述した分析で，XML データベースでは，リンケージ情報が欠けているために，キーワード検索の結果は不完全になる．一方，関係データベースはエンティティ（実体）とリレーションシップ（関連）の両方を管理するため，外部キーを用いて格納したデータ間の関連関係すべてを求められる．しかし，関係データベースは XML データの特有な階層関係を自動的にたどらないため，下位の階層でヒットした情報について上位の階層関係を求めない．その影響で検索結果の情報は不完全になってしまう．

この問題を解決するため, 本研究は XML データの格納を許した関係データベースに基づいて, ハイブリッド型の XML-関係データベース(XML-RDB)システムのデータモデルとハイブリッド型 XML-RDB のキーワード検索機能の設計を提案した．

3.2. ハイブリッド型の XML-RDB システム

Hybrid 型の XML-関係データベースとは，従来のデータ型と XML データ型両方を扱い(図 15 に示すように), Xpath, XQuery, SQL/XML と従来の SQL 問合せ言語を用いて管理するシステムである．

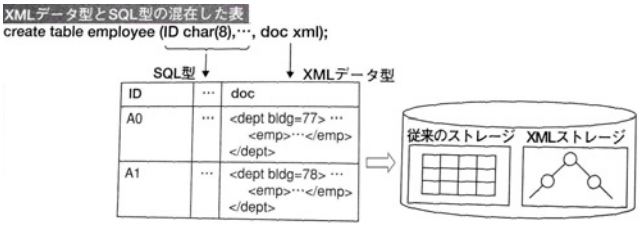


図 15　出典：「XQuery＋XML データベース入門」[4]

3.3. ハイブリッド型の XML-RDB システムのデータモデル

XML 情報を考慮した効率なキーワード検索法を含むハイブリッド型の RDB システムは，まず RDB でのスキーマ設計し，次に XML 表現によってハイブリッド XML-RDB システムへ直し，最後 XML 情報を格納した RDB を実体化する．

前例の図 3，図 5 の場合は，混合型のデータベースシステムのスキーマが決まる前に，図 16 の RDB の ER 図を設計する．

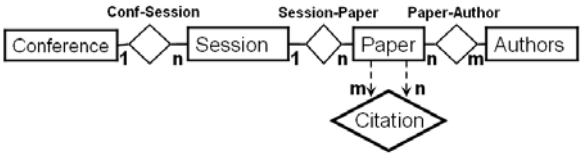


図 16　ER 図

図 16 の ER 図では大会，セッション，論文，著者などエンティティと，引用関係を含め，所属関係によってデータモデルを構成する．

次に, RDB の ER 図によって, 適切な XML 表現を定義し，（この例に対して，Conference-Session-Paper の階層関係情報は XML データの扱いになって，同様に Authors-Paper の階層関係情報も XML データの扱いになる）ハイブリッド型 XML-RDB システムのスキーマへ直す．XML データ表現を考慮して，図 17 のようなハイブリッド型のエンティティ XML 1 と XML2 を設計する．図 18 はハイブリッド関係データベースシステムのデータモデルを示す．（part-of は「一部の」を意味する．）図 19 はそのインスタンスになる．

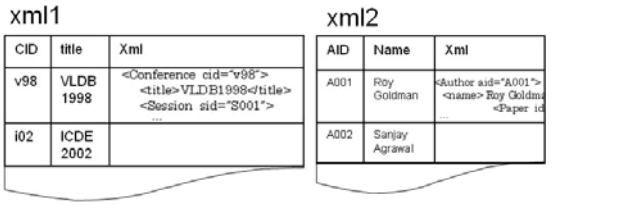


図 17　XML-Data in Hybrid RDB Entity

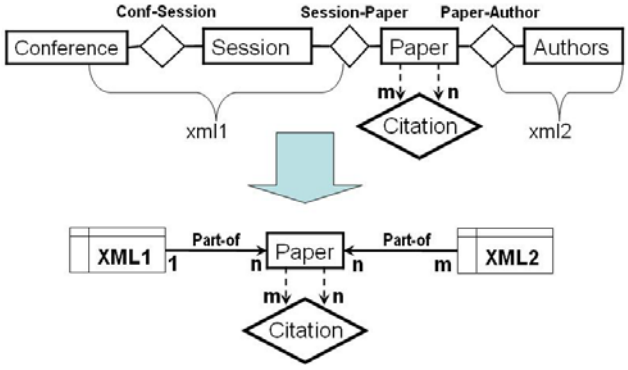


図 18　XML-RDB Data Model

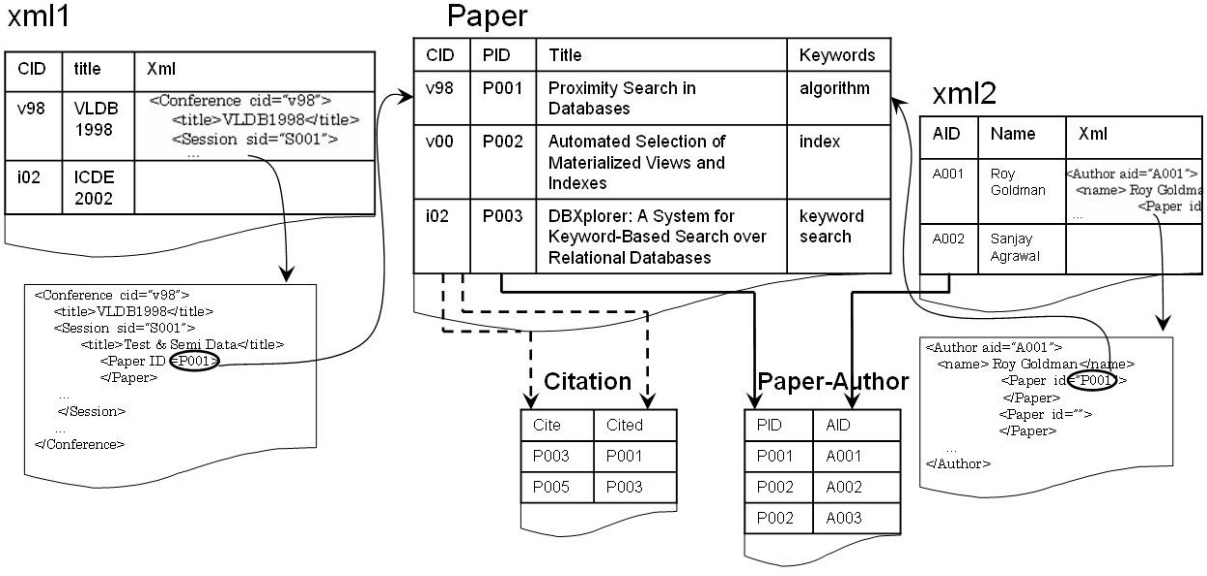


図 19　XML-RDB Instantiation

4. ハイブリッド型の XML-関係データベースシステムにおける検索

ハイブリッド型の XML-RDB システムにおいてキーワードによる検索は関係データベースに基づいて外部キーを用いて Join tree を分析したうえで，検索結果を求める．そのために，XML のエンティティと RDB のエンティティを XRjoin しなければならない．

4.1. XRjoin の定義

XRjoin とは RDB におけるリレーションのエンティティ $\boldsymbol{R}$ と混合型のエンティティ $\boldsymbol{X}$ を結合することである．まず,制約条件がない場合の XRjoin を図 20 に示す．

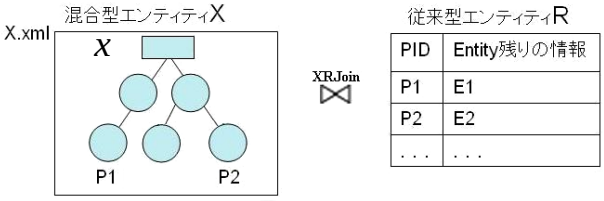


図 20　　XRjoin のオペランド

これを $XRjoin(x_{//pid}, R)$ と記し（$\boldsymbol{X}$ の XML データ $\boldsymbol{X}$ のうち $R.pid$ を満たす部分木を $x_{//R.pid}$ とする），ハイブリッドのリレーション $[x_{//pid}, R._{pid}, R.E]$ を XRjoin の結果として定義する($R.E$ は Entity の残り情報と示す)．結合した結果は図 21 で示す．

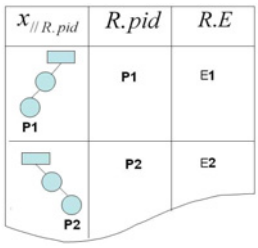


図 21　XRjoin の結果

次に，外部からキーワード入力がある場合（条件を特定したとき）の XRjoin を与える．K1，K2 がヒットしたハイブリッド型エンティティ X と，K3 をみたす RDB のエンティティ R を結合する（図 22 に示す）．図 23 に XRjoin のオペランドを示す．

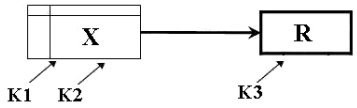


図 22　　$Query = \{K1, K2, K3\}$

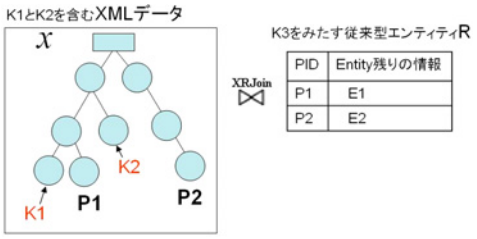


図 23　　$XRjoin(x_{//k1\wedge//k2}, R_{//k3})$ のオペランド

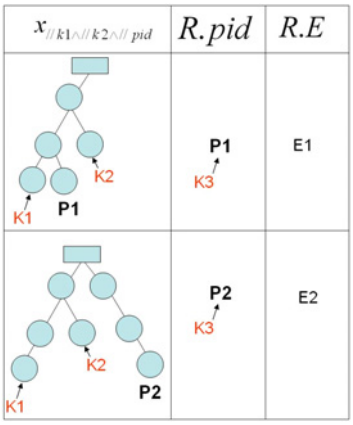


図 24　$XRjoin(x_{//k1\wedge//k2}, R_{//k3})$ の結果

図 24 は三つのキーワード入力による XRjoin の検索結果になる．検索結果はハイブリッド型のエンティティとして出力する．検索結果が必要な情報を含む XML 型の subtree と関連するエンティティ R の情報で構成される．

XML の情報は K1, K2 を含む $\boldsymbol{X}$ のノード部分からみたパスと，K3 をみたす $\boldsymbol{R}$ のタプルに対応する XML のパスから構成した最小部分木である．結果の subtree は XRANK で先祖ノードの抽出手法に従うが，subtree にルートノードまでのパスを加えている．

4.2. 検索結果の例

検索結果の例は前例図 18 の論文ハイブリッドシステムのデータモデルに沿って分析する．

・タイプ 1

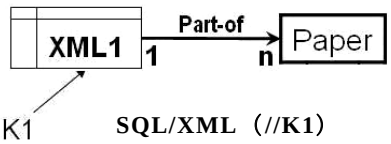


図 25　　$Query = \{K1\}$ in XML1

まず，図 25 のスキーマで，ハイブリッド型のエンティティ XML1 と RDB のエンティティ Paper を関係しているとき，キーワード K1 が XML1 でヒットした場合，XRjoin を分析する（図 25 の XML 1 は図 18，図 19 の XML1 と同じである）．

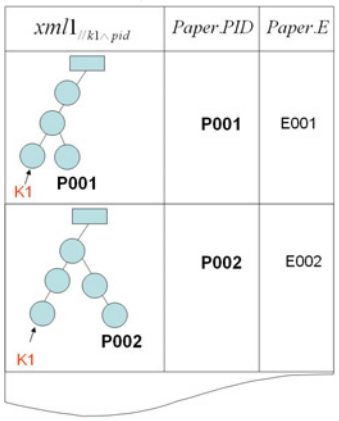
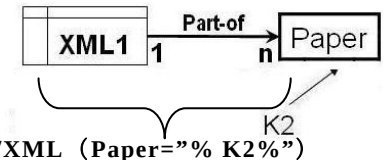


図 26　　$Query = \{K1\}$ in XML1 の答え

図 26 は図 25 のスキーマに沿って実行した検索結果である．Paper 側は条件なしで，XML1 側で条件をつけている．図 26 では SQL/XML 問合せ言語を用いて，XML1 と関連がある Paper のパスと，XML1 上でキーワード K1 を含むノードのパスから構成した subtree を検索結果の XML 情報として出力している．

・タイプ 2



SQL/XML（Paper=”% K2%”）

図 27　*Query* = {*K*2} in Paper

キーワード K2 が RDB のエンティティ Paper でヒットした場合は図 27 のようなスキーマで表される．関連のあるハイブリッド型のエンティティ XML1 の情報も取り出す．

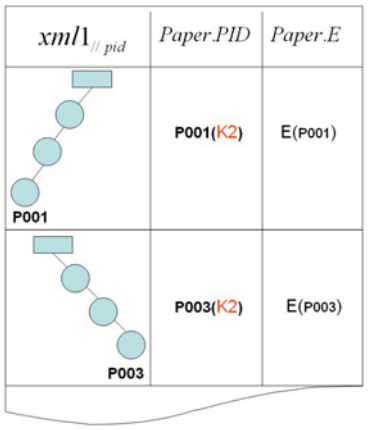
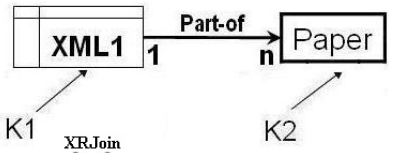


図 28　*Query* = {*K*2} in Paper の答え

図 28 は K2 をみたす Paper 側のタプル情報を検索して，XML1 側でも関連情報と XRjoin した結果である．ハイブリッド型のエンティティで条件なし，従来型のエンティティ側では条件つける例である．

・タイプ 3



SQL/XML(//K1) ⋈ **SQL/XML (Paper=”% K2%”)**

図 29　*Query* = {*K*1, *K*2} in XML-RDB

図 29 ではキーワード多数ある場合，Paper 側と XML1 側を条件つけ，XRjoin を用いて検索を行うスキーマを表す．図 30 は図 29 の場合の XRjoin 結果である．

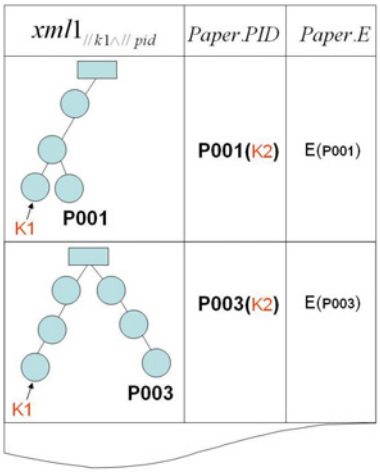


図 30　*Query* = {*K*1, *K*2} in XML-RDB の答え

4.3. ハイブリッド型システムだけできる検索

ハイブリッド型の XML-RDB システム上で，DBXplorer の考えに沿って XRjoin を用いた join tree をつくって実行するシステムを DB2 V9 上に試作した．ここではシンプルな実行例だけ示す．

例を説明するため，実験用のデータとして前例図 19 の論文データを用いる．図 31 はキーワード K1，K2 がそれぞれ XML1 と XML2 でヒットする場合のスキーマ情報である（図 32 がインスタンスである）．

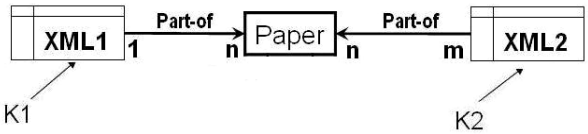


図 31　*Query* = {*K*1, *K*2} in XML-RDB

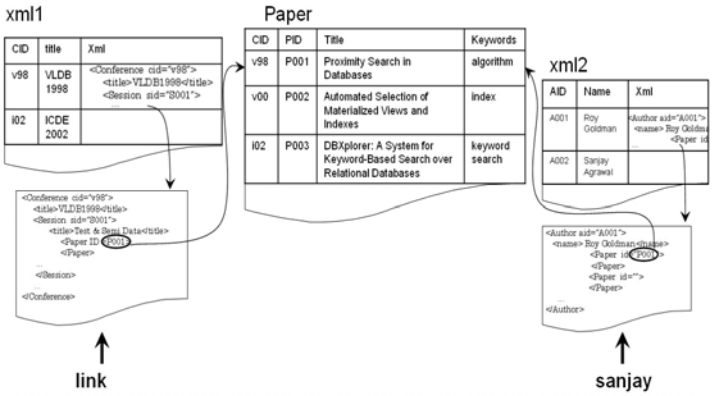


図 32　$K1 = link, K2 = sanjay$ in XML-RDB

キーワード K1，K2 は“link”，“sanjay”とする．図 32 で K1 と K2 がヒットするエンティティを示す．この問い合わせに対応する join tree は，Conference の XML1 と Paper を XRjoin した結果，と Authors の XML2 と Paper を XRjoin した結果の間で，自然結合することを指示する．図 33 は DB2 のハイブリッド型の XML-RDB システム上のこの実行結果になる．

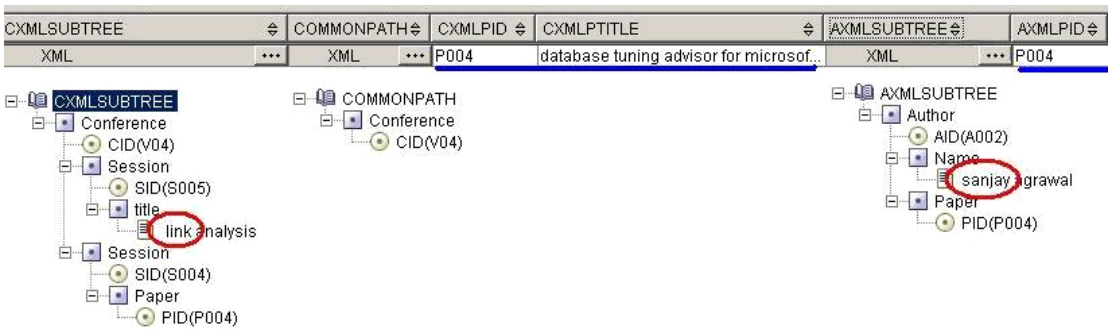


図 33　DB2 上の実行結果

図 33 は Paper の id が一致する場合である．この結果は「Sanjay 氏が書いた Paper004 は“link”に関する Session を含む Conference に出ている」という意味を表わしている．Paper 間に Citation 関係を入れた場合の Query も同様に処理できる．

図 34 はピュアな XML データベースで図 33 の検索結果を示してある．XML データベースではこの検索結果が得られない．同様に，関係データベースでも得られない結果である（“link”と P004 は同じ Session ではないから）．

ハイブリッド型の XML-関係データベースでキーワードによる検索は XML データベースと RDB の検索よりよい結果を得た．ただし，XRjoin を使ったときの join tree を正確につくるためには DBXplorer[1]システムの join tree の生成方法を修正する必要がある．

5. おわりに

本研究は XML データベースと関係データベースにおける従来のキーワード検索方式の機能比較を行い，XML データに相当する情報を格納するとき，両者が共に不十分な解しか探せないことを示した．これを解決するため，ハイブリッド型 XML-関係データベースにおいてキーワード検索を行うアルゴリズムを提案した．

ハイブリッド型の XML-関係データベースで XML データ表現を考慮したデータモデルと XRjoin 結合方式を用いた join tree 生成による検索方法を提案した．XML データベースと従来の関係データベースにおけるキーワード検索より合理的な答えが得られることを示した．XRjoin を使う場合の join tree の自動生成法については稿を改めて述べたい．

文　献

[1] Sanjay Agrawal, Surajit Chaudhuri, Gautam Das, “DBXplorer: A System for Keyword-Based Search over Relational Databases”, Proceedings of the 18th International Conference on Data Engineering, pp05-17, SanJose, California, USA, Mar., 2002.

[2] Lin Guo, Feng Shao, Chavdar Botev, Jayavel Shanmugasundaram, “XRANK: Ranked Keyword Search over XML Documents”, Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp.16-27, SanDiego, California, USA, Jun. 2003.

[3] Bolin Ding, Jeffrey Xu Yu, Shan Wang, Lu Qin, Xiao Zhang, Xuemin Lin, “Finding Top-k Min-Cost Connected Trees in Databases”, Proceedings of the 23th International Conference on Data Engineering, pp.836-845, Istanbul, Turkey, Apr. 2007.

[4] 菅原 香代子，米持 幸寿，XQuery+XML データベース入門，日経 BP 社，東京，2006.

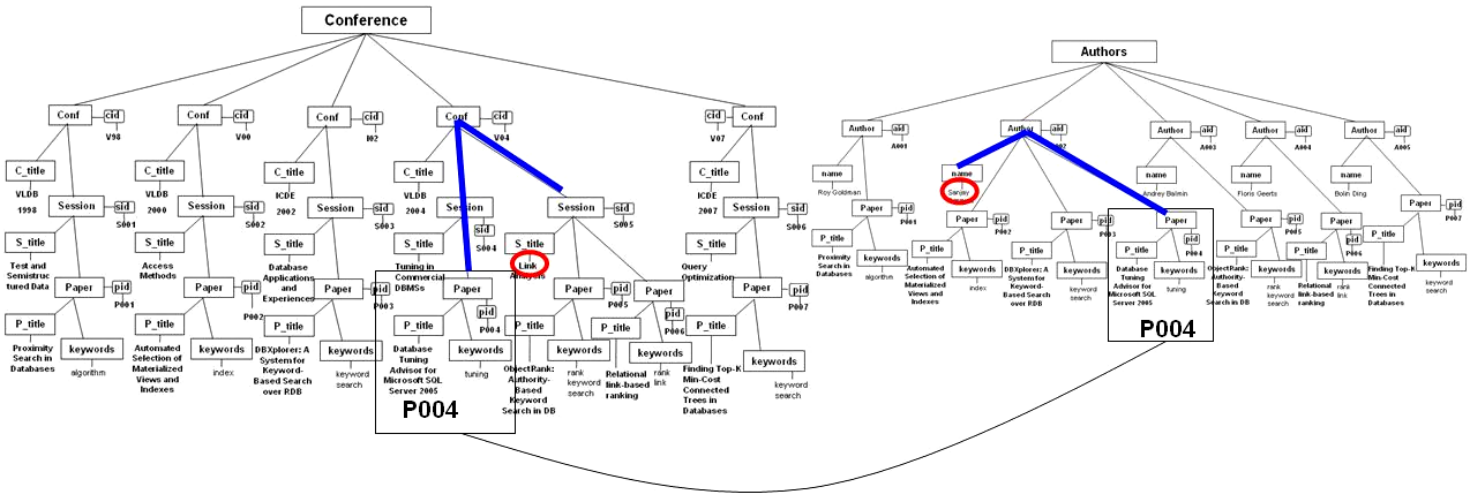


図 34　XML データベースにおいて図 33 の検索結果