キーワードを利用したXML文書検索

絹谷弘子[†,☆] 波多野賢治[††]
吉川正俊[†††] 植村俊亮[††]

1998年にW3CがXMLを勧告してから6年が経過し，次世代のWeb言語としてXMLはその応用範囲を急速に拡大している．また1990年代にインターネットの普及にともなって開発されたWebサーチエンジンは，情報検索技術を基盤として発展し，今やインターネット検索には不可欠なものとなっている．Webサーチエンジンが，検索キーワードに関連するHTMLで記述されたWebページに順位を付けて提示するように，検索キーワードから利用者が必要とするXML文書を順位付けして提示するXMLサーチエンジンへの要求は大きい．さらに，XML文書が持つ柔軟性から，文書単位の検索に加えて，文書内の論理構造で分割された部分文書単位の検索が可能であり，Webサーチエンジン同様の前文検索技術をそのままXMLサーチエンジンに適用するだけでは不十分である．そのために，XMLサーチエンジンには新しい基盤技術，すなわち検索言語と検索方法，解答候補の部分文書決定方法や性能評価方法，などの研究が必要である．本論文では，構造化文書に対するキーワードを利用した検索に関する研究を振り返り，最近の研究動向について紹介し，XMLサーチエンジンに必要な基盤技術と今後の展望について述べる．

A Survey of Keyword-based XML Document Retrieval

Hiroko Kinutani,[†,☆] Kenji Hatano,[††] Masatoshi Yoshikawa[†††]
and Shunsuke Uemura[††]

XML, which has been recommended by W3C since 1998, is recognized as the next generation Web language. And also, Web search engines have been developed with popularization in the Internet in the 1990s, which were developed based on information retrieval techniques. Now, they are indispensable for the Internet search. Web search engines enable to search for HTML documents with their ranks according to input keywords. In a similar way to the development of Web search engines, XML search engines will become very important tools for users wishing to search for XML documents. However, the techniques of Web search engines cannot be applied to XML search engines directly. This is because the retrieval targets of XML search engines are not only XML documents themselves, but also their fragments, so their structures must also be taken into account. Therefore, XML search engines require new techniques, such as search language, search techniques, automatic determination of granularity of retrieval results and their evaluation. In this paper, we report on recent research into keyword-based structured document retrieval systems. Moreover, we describe fundamental issues and the future direction of XML search engines.

1. はじめに

1998年にWorld Wide Web Consortium（W3C）[☆☆]がXMLを勧告してから6年が経過し，次世代のWeb言語としてXML[1)~5)]はその応用範囲を急速に拡大している．またXMLの普及にともなってXML文書検索に関する研究に注目が集まっている．

すでにWebサーチエンジンは，インターネット上でアクセス可能なWebページを巨大なデータベースとして扱い，利用者の必要な情報を提供する手段を全文検索技術を基盤として開発してきた．ここでは，利

† 科学技術振興機構戦略的基礎研究推進事業
Core Research for Evolutional Science and Technology（CREST）Program, Japan Science and Technology Agency（JST）
☆ 現在，お茶の水女子大学総合情報処理センター
Presently with Information, Media and Education Square, Ochanomizu University
†† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology（NAIST）
††† 名古屋大学情報基盤センター
Information Technology Center, Nagoya University

☆☆ http://www.w3.org/

用者がいくつかのキーワードを入力すると，関連するHTMLで記述されたWebページに順位を付けた検索結果が提示される．

XML文書を対象とした検索エンジン（以下「**XMLサーチエンジン**」と呼ぶ）に対してもWebサーチエンジンと同様に，検索キーワードから利用者が必要とするXML文書の順位付けられた検索結果を求める要求は大きい．HTMLは表示に関するマークアップ言語であり，HTML文書の内容は人間が読むことが目的である．一方，XML文書の内容には，人間が読むことを目的とした文書としての性質（文書系XML）とプログラムが解釈して処理を行うデータとしての性質（データ系XML）をあわせ持つ場合がある[6),7)]．さらにXML文書内の論理構造で分割された部分文書を単位として管理することが可能であり，Webサーチエンジンの技術を適用するだけではXML文書の柔軟性を活かしきれない．またXMLサーチエンジンには，Webサーチエンジン同様の全文検索技術に加え，文書構造単位でのデータベース的なフィールド検索技術も必要となる．さらにWebサーチエンジンが行っているWebページのリンク構造解析，すなわち文書の引用関係のネットワーク情報を利用して関連する文書を検索結果として抽出することや，引用度のスコアを組み合わせることにより検索結果の重要度順を決定することをXMLサーチエンジンに応用したり，XML文書内のデータやメタデータを効率的に管理したりする技術が必要である．そのために，XMLサーチエンジンの新しい基盤技術，たとえば検索方法，解答候補の部分文書決定方法や性能評価方法など，に対する研究が必要である．

本論文では，次世代のWebサーチエンジンとして嘱望されるXMLサーチエンジンに関する研究動向を紹介する．まず構造化文書に対するキーワードを利用した検索に関する研究を振り返り，XMLサーチエンジンに関する最近の研究動向について紹介し，XMLサーチエンジンに必要な基盤技術と今後の展望について述べる．

以下2章ではXML文書の基本事項とXMLサーチエンジンにおけるキーワードを利用した検索の問題点を明らかにし，3章ではXMLの前身であるSGML文書に対する検索について，4章ではデータベース研究者の立場から，XML問合せ言語の土台にキーワード検索機能を追加した検索言語の提案について，5章では情報検索研究者の立場から，XMLサーチエンジンに関する提案について，6章では検索結果の評価方法に関する最近の取組みについて述べ，最後に7章では今後の展望と研究課題について述べる．

2. XMLサーチエンジン

まずXML文書の基本事項を明らかにする．次に従来のWebサーチエンジンとXMLサーチエンジンの違いを明らかにし，XMLサーチエンジンにおけるキーワードを利用した検索の例をあげてその問題点を明らかにする．

2.1 基本事項

2.1.1 XML文書

XMLは文書やデータを表現するテキストをタグによってマークアップした言語である．XML文書は1つ以上の「**要素（Element）**」を含む．XML文書には「**文書要素**」が必ず1つあり，その他の要素は開始タグおよび終了タグがその「**要素の内容**」を囲む．開始タグに**「属性」**を指定することができる．要素は入れ子にすることができ，要素が入れ子構造をなす場合を「**整形式のXML文書（well-formed）**」と呼ぶ．「**妥当なXML文書（valid）**」はさらに，**文書型定義（DTD）**や**XMLスキーマ定義**で別に定義した文書構造を満たすXML文書のことである．

XML文書は唯一の文書要素を持つため，文書要素を根ノードとし，他の要素は要素名をラベルに持つ中間ノード（要素ノード），要素の内容であるテキストはテキストをラベルに持つ葉ノード（テキストノード）として木構造でモデル化することができる☆．

図1，**図2**はXML文書のDTDとそのDTDを満たすXML文書の一例である．**図3**は図2のXML文書が持つ論理構造の木構造表現である．図2の左端の番号はノード番号を表し，その右側の要素と図3の要素ノードが対応する．

次に，XML文書を文書構造を反映して分割した部分文書について紹介する．

2.1.2 XML部分文書

構造化文書検索モデルには，重複のないリストモデル（*non–overlapping lists*）と近接ノードモデル（*proximal nodes*）がある[8)]．リストモデルでは文書を前から順番に章や節を単位として文書を重複のないように分割し，分割した部分を部分文書として索引付けを行う．検索キーワードの出現箇所から分割に利用した構造を識別することはできるが，入れ子関係にあるその他の構造については識別できない[9)]．一方近接ノードモデルでは構造化文書の論理構造を保持した分

☆ XML文書のデータモデルはリンク構造を考慮してグラフとする場合もあるが，本論文では木構造とする．また，単純化のため属性やコメントなどのノードは省略する．

```
<!ELEMENT article (hdr,abst,bdy)>
<!ELEMENT hdr  (ti, au+)>
<!ELEMENT abst (#PCDATA)>
<!ELEMENT bdy  (sec+)>
<!ELEMENT sec  (ti*, (ss1* | p*))>
<!ELEMENT ss1  (st*, (ss2* | p*))>
<!ELEMENT ss2  (st*, p*)>
<!ELEMENT ti   (#PCDATA)>
<!ELEMENT au   (#PCDATA)>
<!ELEMENT st   (#PCDATA)>
<!ELEMENT p    (#PCDATA)>
```

図 1 DTD 例
Fig. 1 A sample DTD.

```
ノード
番号
1 <article>
2 <hdr>
3  <ti> COMPUTER GRAPHICS </ti>
4  <au> Ronald Azusa </au>
  </hdr>
5 <abst> Azusa published a survey on augmented
       reality. </abst>
6 <bdy>
7  <sec>
8   <p> What is augmented reality? An AR system
      supplements the real world with virtual
      objects that appear to coexist in the same
      space as the real world. </p>
9   <p> AR's growth and progress have been
      remarkable.</p>
   </sec>
10 <sec>
11  <ti> ENABLING TECHNOLOGIES </ti>
12  <ss1>
13   <st> Displays </st>
14   <p> We can classify displays into the following
        categories: head worn, handheld,
        and projective. </p>
15   <ss2>
16    <st> Head-worn displays </st>
17    <p> Users mount this type of display
        on their heads. </p>
18    <p> Head-worn AR displays would be
        no larger than a pair of sunglasses. </p>
     </ss2>
19   <ss2>
20    <st> Problem areas in AR displays. </st>
21    <p> See-through displays don't have sufficient
        brightness. </p>
     </ss2>
    </ss1>
    </sec>
  </bdy>
 </article>
```

図 2 XML 文書例
Fig. 2 A sample XML document.

割方法で文書を分割し，分割した部分を部分文書とする[10]．この方法では文書構造を保持した索引を利用することで，文書構造とキーワードを指定した検索指定ができる．また構造化文書中の語の位置について，フラットな文字列の並びとして文書の前からの位置と，文書の階層構造上の位置の両者が利用できる．その結果，現在の XML 文書検索研究では，この近接ノードモデルを利用することが多い．

本論文では XML 文書中に出現するすべての要素について，開始タグと終了タグで囲まれた部分を XML 部分文書と呼ぶ．すなわち，要素ノードを根とする木全体を XML 部分文書と呼ぶ．いい換えれば，各部分文書は必ず 1 つの最上位ノードを持つ部分木の範囲であり，部分文書をその最上位要素ノードの番号 n を用いて指

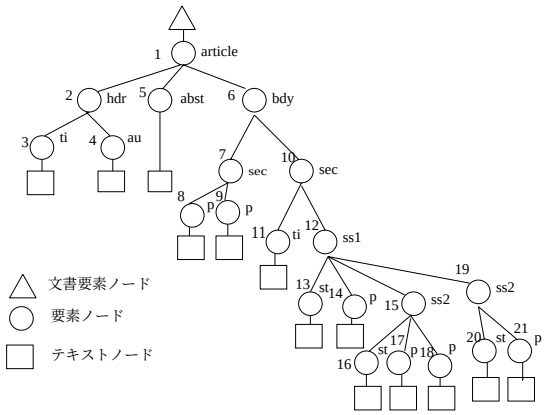

図 3 XML 文書の木構造表現
Fig. 3 Tree representation of the sample XML document.

定することができる．図 3 においてはノード番号#1 から#21 までの 21 個の部分文書を考えることができる．

2.1.3 XML 文書検索研究の 2 つの流れ

従来から構造化文書を対象とした研究者の研究分野が 2 つある．一方がデータベース研究であり，他方が情報検索研究である．両者の研究分野の違いから研究の着眼点が異なっていた．

前者は主にデータ系 XML を扱い，文書中の特定の項目（構造）について高速なデータ処理を行う観点からデータベース管理システムに構造化文書をどう適応させるかに主眼がある．したがって利用者の問合せは，データベースへの問合せ同様，指定した項目に含まれる値を探すことである．問合せ処理は，問合せ言語によって記述された条件を満たす値を完全照合で求める正確な手続きである．問合せを行うために利用者は文書構造と問合せ言語の知識をあらかじめ持つ必要がある．

一方後者は主に文書系 XML を扱い，利用者の検索要求に対して適切な文書や文書の一部を検索結果として特定するためのアルゴリズムと検索行動に主眼がある．検索システムの目的は従来の情報検索と同様で，必要としている情報を探すための手段を提供することである．そのため，検索キーワードの同意語による拡張や検索結果の順位付けなども考慮されるあいまい性のある手続きである．検索を行うための検索言語は，一般的には検索キーワードを列挙したものであり，その指定方法は前者に比べれば簡単である☆．**図 4** は

☆ 本論文では完全照合結果を求める場合を**「問合せ（query）」**，順位付けのある結果を求める場合を**「検索（search）」**と呼ぶことにする．「検索式」に比べ「検索言語」といういい方は一般的ではないが，本論文では，問合せ言語と対比するために，検索要求を検索システムに入力する方法を「検索言語」と呼ぶ．実際，Cohen らは検索言語を定義している[11]．

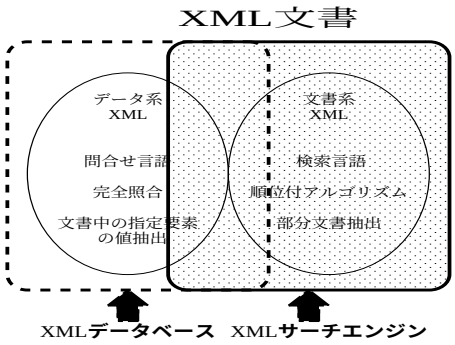

図 4 XML 文書の分類
Fig. 4 A classification of XML documents.

XML 文書を対象とした検索システムが持つ性質をおおまかに分類したものである．実際の XML 文書中にはデータ系 XML と文書系 XML が併存する場合があり，XML サーチエンジンは，データベースへの問合せ機能と情報検索機能を保持することが望まれる．XML サーチエンジンは，従来の構造化文書検索や Web サーチエンジンでは想定していなかった部分文書を単位として，データ系 XML と文書系 XML の両方を処理する必要がある．

2.2 XML サーチエンジンと Web サーチエンジン

XML サーチエンジンは，主に文書系 XML を対象とし，検索キーワードを利用して利用者が必要とする XML 文書や XML 部分文書を順位付けして提示するシステムである．**表 1** は XML サーチエンジンと Web サーチエンジンを比較した表である．XML サーチエンジンの目的は入力キーワードから関連する情報を入手することであり，検索結果は利用者にもたらす新たな情報の有無や満足度によって評価される点は Web サーチエンジンと同じである．大きな違いは，多様な構造の XML 文書を対象とする場合や表示と検索の最小単位が部分文書であることである．Web サーチエンジンでは Web ページ中に複数の話題があったり，出現キーワード間に文脈上まったく関連がない場合や必要な情報が Web ページの一部分である場合を識別できなかった．しかし，XML サーチエンジンでは，XML 文書を細分化することでこれらを識別することが可能になる．

さらに XML サーチエンジンでは，検索結果の順位付けも文書単位，部分文書単位あるいは特定の要素単位に行うことが可能であり，文書構造を文脈と見なすことで，出力する検索結果に文脈上最適な部分文書を特定することも可能である．Web サーチエンジンの検索結果がつねに HTML 文書単位であったのに対し，XML サーチエンジンでは細分化された部分文書単位の検索結果を求める必要がある．

表 1 XML サーチエンジンと Web サーチエンジン
Table 1 XML search engines and Web search engines.

	XML サーチエンジン	Web サーチエンジン
対象文書	XML 文書	HTML 文書
文書構造 （DTD など）	すべて同じ場合（妥当 XML） 異なる場合（整形式 XML）	すべて同じ
表示	スタイルシートでの編集が必要	Web ブラウザ
表示と検索の最小単位	XML 部分文書	HTML 文書
索引付け	XML 文書単位 特定の要素単位 XML 部分文書単位	HTML 文書単位 特定の要素単位（TITLE, META など）
検索指定方法	キーワード AND，OR，NOT 検索 要素名とキーワード	キーワード AND，OR，NOT 検索
リンク構造	文書内（ID/IDREF） XML 文書間の関係（XPointer）	HTML 文書間の関係
順位付け	XML 文書単位 XML 部分文書単位 特定の要素単位	HTML 文書単位
検索結果	XML 文書（XML 文書へのポインタ） XML 部分文書（XML 部分文書へのポインタ）	HTML 文書（HTML 文書へのポインタ）

XML サーチエンジンが新たに必要としている技術は，Web サーチエンジンでは不可能な文書構造によって細分化された部分文書単位での管理である．特に，検索対象や検索結果となる部分文書の条件を指定できる検索言語，解答候補となる部分文書の決定方法や検索結果の順位付け方法，評価方法が重要である．

2.3 キーワードを利用した XML 文書検索の例

XML サーチエンジンが新たに必要としている研究課題には，（1）検索指定方法，（2）解答候補となる部分文書の決定方法，（3）検索結果の順位付けと評価方法などがある．本節では，現状での XML サーチエンジン研究における問題点について例を示して説明する．

2.3.1 検索指定方法

Web サーチエンジンのように，XML サーチエンジンの検索指定も簡単な方法が望まれる．しかも部分文書単位の高精度の検索を行うための指定方法が必要である．XML 文書に対するキーワードを利用した検索指定方法は 2 種類に分類される．

(1) 利用者が従来の Web サーチエンジン同様にキーワードのリストを指定する場合（Content Only 検

索：CO 検索）

> *“AR Augmented Reality display”*

この検索指定では *“Augumented Reality”* の表示に関する部分文書を検索する．これらのキーワードを文書構造の要素名と文書内容の両者に適用する場合と，文書内容のみに適用する場合が考えられる．

(2) 明示的に文書構造と文書内容に関するキーワードを区別して指定する場合（Content and Structure 検索：CAS 検索）

> `sec` *contains “AR Augmented Reality display”*

この検索指定では要素 `sec` の内容が *“Augumented Reality”* の表示に関連する部分文書を検索する．この場合関連度による順位付けは文書中の要素 `sec` が根ノードとなっている部分文書に対して行う．さらに要素 `sec` に関して同等の要素，たとえば `subsec` や `ss1`，`ss2` にも対象を広げる場合も考えられる．

Web サーチエンジンは，HTML 文書単位の検索であり，検索対象も検索結果も HTML 文書集合であった．XML サーチエンジンは，XML 部分文書単位の検索であり，検索対象も検索結果も部分文書集合である．そのため，CO 検索では，検索対象，解答候補となる部分文書をシステムが決定する必要がある．また CAS 検索では，検索対象，解答候補の部分文書の条件を指定する必要がある．したがって，高精度な検索のためには，新たな検索条件の指定方法や機能的な検索言語が必要である．

2.3.2 解答候補となる部分文書の決定方法

検索の解答候補となる部分文書の決定方法は利用者の検索目的による．たとえば利用者が Augmented Reality の表示に関してキーワード *“AR Augmented Reality display”* で検索を行う場合，**図 5** 中のノード番号`#1` から`#21` までの部分文書を検索対象とすることができる．この文書全体では，検索キーワードとの関連性が低くても，部分文書として関連性の強い部分を取り出せる．この文書では，キーワードを含むテキストノードを持つ部分文書は，図 5 のようにノード番号`#1`，`#6-10`，`#12`，`#13`，`#15-21` の部分文書であり，これらについて検索キーワードとの関連性による順位付けがあれば利用者は便利である．

検索結果に利用者が期待するものが「なるべく余計な情報を含まず検索キーワードと関連の強い小さな部分文書を探すこと」である場合と「検索キーワードに関連する内容を網羅するためには周辺情報や抽象的な情報をも含んだ適当な大きさの部分文書を探すこと」である場合によって順位付けの方針も異なると考えられる[12]．前者の場合，ノード番号`#12`，`#15` や`#19` の部分文書が期待され，後者の場合，ノード番号`#1`，`#6`，`#7` や`#10` の部分文書が期待されるだろう．さらにノード番号`#8`，`#9`，`#13`，`#16-18`，`#20`，`#21` の部分文書も解答候補になりうる．

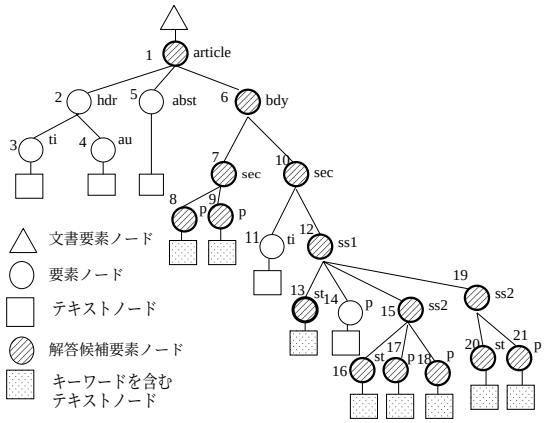

図 5 検索キーワードを含むテキストノードと解答候補の部分文書
Fig. 5 Text nodes containing query keywords and partial documents of result candidate.

2.3.3 検索結果の順位付けと評価方法

検索キーワードと解答候補の XML 部分文書との類似度計算には，テキスト内容と検索キーワードの類似度計算に加え，検索キーワードが出現する文書内の位置やノードの関係，さらに文書の階層構造やリンク構造を考慮した計算方法も考えられる．これらを総合して順位付けすることで検索性能が向上する可能性がある．

検索結果は，検索キーワードとの類似度によって順序付けられた XML 部分文書のリストである．そのため部分文書間に入れ子関係がある場合も考えられる．解答候補の部分文書間に重複がある場合，検索結果に重複を認めた順位付けと重複を排除した順位付けの 2 通りが考えられる．また，検索結果が利用者の検索要求に適合するか否かの判定も重複を認める場合と，高順位に出現した検索結果との重複を排除する場合で異なる．

3. 構造化文書に対する検索

XML が出現する以前から構造化文書を記述する言語として利用されていた Standard Generalized Markup Language（SGML）[13)~15)] は，デバイスやシステムに

依存せずにテキストを電子的に表現するための方法を定義する国際規格である．SGML はマニュアルや電子図書館における電子出版で文書の構造化をすすめるためのマークアップ言語として 1986 年に制定され，その後構造化文書データベースと検索に関する研究がなされてきた[10),16)~21)]．従来の情報検索では，プレーンテキストを対象として，文書内に書誌情報やタイトル，章，節などを含んでいても計算機が文書内容を解析して文書構造を抽出することが困難であった．構造化文書である SGML 文書を対象とした検索システムでは，各文書の構造が DTD によってあらかじめ定義されているので，各要素が文書中で持つ役割があらかじめ分析できる．その結果検索対象となる文書中の構造（たとえばタイトルや前書き）をあらかじめ絞り込むことで検索性能を改善することが可能となった．

3.1 構造化文書が必要とする問合せ

1994 年，Sacks-Davis らは SGML 文書集合が必要としている問合せを 8 つのクラスに分類した[18)]．そのほとんどは問合せ結果の真偽値を求める完全照合検索であるが，文書内容に関する順位付き検索について，文書全体の順位付けだけではなく，指定した文書構造内の内容に関して順位付けを行う必要性を述べた．次の問合せでは，指定した要素内容について順位付けられた結果を求める．

> *Find <section>s with (<para>s similar to "parallel processing" with similarity measure > 0.2)*

この場合の問合せと文書との類似度の計算には，よく使われている問合せベクトルと文書ベクトルのコサイン相関値が利用されることを見込んでいた．すなわち，W_d を文書 d の長さ，$w_{x,t}$ を文書（あるいは問合せ）x 中のキーワード t の重みとすると，問合せ q と文書 d の類似度 $C_{q,d}$ は次の式で表される．

$$C_{q,d} = \frac{\sum_t (w_{q,t} \cdot w_{d,t})}{W_d}$$

しかし，構造化文書集合に対する問合せ言語は具体的には提案されなかった．

3.2 検索結果の順位付け

1993 年，Fuller らはハイパーテキストや SGML 文書について各ノードを独立した文書と考え，構造化されたノードをハイパーテキストデータベースに格納し，内容に基づく抽出と表示方法を考察した[16)]．文書が持つ構造情報は検索に役立つと考え，文書を小さな部分文書に分割し，分割した部分文書を検索単位，索引生成の単位，評価単位，解答候補の表示単位として考察した．検索結果のノードの順位付けと表示方法として，(1) 該当ノードの類似度とそのノードの子ノードの類似度を子供の数で割ったものの組合せで再帰的に計算する方法，すなわち w_{ij}^k を文書 D_i の根ノードの k 番目の子中の j 番目の語の重みとし，s を文書 D_i の根ノードの子ノードの数とすると問合せ Q と文書 D_i との類似度 C_{Q,D_i} は次の式で表される方法，

$$C_{Q,D_i} = \frac{\sum_{k=1}^{s} \sum_{j=1}^{t} q_j w_{ij}^k}{s}$$

で求めて類似度順に表示する方法と，(2) 問合せ Q と文書 D_i との類似度 C_{Q,D_i} は一般的なコサイン相関値で求めるが，利用者が表示を必要とする構造を前もって指定し，検索結果を表示する場合に類似度が低くても指定したノードを高類似度のノードより優先するかどうかを決定する方法を提案した．

1994 年，Wilkinson は構造化文書検索において特定の構造，たとえば section を検索する場合には，文書全体を検索対象とした順位付けではなく section を単位とした部分文書を検索対象とした順位付けの方が精度が上がることを実験で示した[19)]．そのため文書全体の索引付けとは別に文書中の section を単位とした部分文書に分割して索引付けを行っている．さらに同じ section でも内容によって目的，要旨，タイトルなどに分類して重みを変化させて順位付けを行うことで精度が向上することを示した．この研究により構造化文書検索には，文書構造を利用した索引を利用することが重要であることを示した．

一方 1993 年に Kilpeläinen らは問合せを構造化文書の論理木構造と問合せ木構造の tree inclusion 問題としてとらえた[17)]．利用者は文書構造の正確な知識は必要とせず，検索対象として必要な部分の文書構造と文書内容だけを指定する．問合せパターンの文書中でのコストを定義し，検索システムは最小のコストのパターンを問合せ結果として求める．順序木の木構造マッチング問題の計算量は NP 完全であるが，ノード数 m の問合せ木とノード数 n のパターン木の場合は $O(mn)$ であり，ラベルが再帰的に出現しない場合は $O(n)$ であることを示した．

1998 年，Myaeng らは内容と構造に基づく要求を満たす情報検索システムの構築を目的として，推論ネットワークに基づく検索手法を提案した[21)]．このシステムでは文書構造と文書内容に関する検索条件と，検索結果として取り出す文書構造を指定し，推論ネットワークを利用して解答候補のスコアを計算する．検索処理は，(1) 検索キーワードを少なくとも 1 つ含む

“retrieval” 要素や葉ノードとその間の経路を候補としてマークをつける，(2) “retrieval” 要素ごとに確率を計算する，(3) 検索キーワードを持つ葉ノード全部の確率を計算する，(4) 検索式の演算子に基づいて各検索キーワードの確率を集計して “retrieval” 要素の確率を計算する，の 4 段階で構成される．

このシステムの実験結果では，構造のない文書に比べ構造と内容に基づく検索の精度が高いこと，文書全体を取り出す場合に，すべての要素を利用した場合より特定の要素を利用した方が精度が高く，さらに重み付けすることでその効果があがることを示した．

以上の研究では SGML 文書集合は，単一の DTD に従った共通の文書構造を持っていることを前提として，従来の文書単位の情報検索では不可能であった特定の要素を対象とした順位付き検索や文書構造を考慮した順位付けを可能とした．

3.3 検索結果の評価尺度

SGML 文書集合に対する高精度検索の手法の提案のかたわら，構造化文書が持っている文書の論理階層構造に注目して解答候補として最適な文書の粒度とその評価尺度について検討した研究を Lalmas は行っている．

1997 年，Lalmas は構造化文書中の階層構造の中間要素が検索キーワードとの関連性が一番強い場合を考慮に入れて，文書の検索単位を文書中の部分要素（オブジェクト）とした．上位の要素の意味的内容は最下位の要素オブジェクトの意味的内容の集約とし，この集約をあいまい性を表現できる証拠理論（Dempster-Shafer 理論）によってモデル化した[22]．検索キーワードを含む要素オブジェクト 1 つに対し，そのオブジェクトをはさむ階層構造の最上位から最下位までの間の経路に存在するすべての要素オブジェクトを解答候補としている．システムは利用者がその解答を見るエントリと階層を上下に移動して表示する手段を提供することで，構造化文書の解答候補を利用者に分かりやすく表示できると述べている．このような解答候補に関する利用者の満足の度合いの前提として，利用者の不正確な知識やあいまい性を考慮することが重要であると主張した．ここでは利用者が解答候補に満足する指標として解答候補がどれだけ検索キーワードに関連した特有な内容であるかを表す *specificity* と解答候補がどれだけ検索キーワードに関する内容を網羅しているかを表す *exhaustivity* の 2 つの指標の利用を提案している．Lalmas の証拠理論は，それまでの確率モデルになかった利用者の検索行動のあいまい性を考慮したことが特徴である．しかし評価尺度の提案は単なる提案にとどまっていた．

表 2 SGML 文書検索と XML サーチエンジン
Table 2 SGML docment retrieval and XML search engines.

	SGML 文書検索	XML サーチエンジン
対象文書	SGML 文書	XML 文書
文書構造 DTD など	すべて同じ 必須	すべて同じ（妥当 XML） 必須ではない 異なる（整形式 XML）
ネットワーク	非前提	前提
仕様	厳密で複雑	柔軟で単純
利用範囲	電子マニュアル，電子出版，公文書	Web 上の文書，オフィス関係書類，アプリケーション言語による業界の語彙利用など
ツール	専用ツール	オープンソース，公開モジュールなど多数
関連仕様	2〜3	多数
普及	限定的	広範囲

3.4 SGML 文書検索と XML サーチエンジン

XML と SGML の違いは，XML が SGML に比べより柔軟性および拡張性のあることである．**表 2** は，SGML 文書検索と XML サーチエンジンを比較した表である．SGML 文書は必ず DTD によって文書構造を定義しなければならず，文書構造を改変することは容易ではない．XML 文書では妥当な XML 文書に加え文書構造の入れ子関係さえ成立していれば DTD による文書構造の定義を必要としない整形式 XML 文書もあり，文書の作成や改変が比較的容易である．さらに多種多様な複数の XML 文書に対する検索の必要性が高い．

SGML は電子マニュアル，電子出版あるいは公文書などで利用され，厳密な文書管理を必要とする分野で利用されてきた．その結果 SGML 文書検索を行う利用者は，対象文書と文書検索システムについて精通していた．XML サーチエンジンの利用者は，Web サーチエンジン同様，対象文書やサーチエンジンについての事前知識を持たない．その結果 XML サーチエンジンに不慣れな利用者でも検索可能な操作が重要である．また，XML の利用範囲は，SGML に比べはるかに広いため，利用用途に応じた XML サーチエンジンの応用範囲も広い．

4. XML 問合せ言語におけるキーワード検索機能

XML サーチエンジンの検索言語には，データベースへの問合せ機能と情報検索機能を保持することが望まれる．そのため XML サーチエンジンは，従来の構

造化文書検索や Web サーチエンジンでは想定していなかったデータ系 XML と文書系 XML の両方を処理する必要がある．本章ではデータベース研究者を中心とした XML 問合せ言語におけるキーワード検索機能拡張に関する研究を紹介する．

XML 文書からのデータ抽出や再構成のために標準的な XML 問合せ言語が必要であることは XML が W3C の勧告になった 1998 年から認識されていた．同年 W3C による XML の問合せ言語に関するワークショップ QL'98 が開催され，多数の XML 問合せ言語が提案された☆1．

特に XQL[23]，XML–QL[24),25]，Quilt[26]は問合せ言語として広く認知されていたが，それらの良いところを採用し，さらに問合せ代数によって形式化された言語として XQuery[27]が登場した．これらの XML 問合せ言語はデータベース研究者によって提案されたため，データ系 XML に主眼があり，XML 文書集合を一括管理することを目的としている．

4.1 XQuery

W3C が検討している XML 問合せ言語 XQuery についての最初の提案は 2001 年に行われた．XQuery の目標は「単一の文書や複数の文書集合に対しての問合せで，文書全体や文書内容や文書構造に関しての問合せ条件に合った文書の部分木を選び，選択したものについての新しい文書を構築できること」である[28]．XQuery は半構造データベースとしての XML 文書集合に対して必要な文書構造だけを抽出し，その結果を利用者が必要とする文書構造に変換して出力することを目的とした言語であり，高速で効率の良い処理を目指している．XQuery は FLWR（`For, Let, Where, Return`）式で構成され，文書構造，文書内容に関する問合せを文書中のノード集合に対する操作として記述する．特に `Where` 節の部分の述語やフィルタ条件を指定した場合，条件を満たすか否かの真偽値によって処理を行う．条件式には文書内容の文字列に関する指定ができる．

文字列の比較，大文字小文字の変換，文字列の結合など文字列に関するデータ操作を行う関数が定義されている．たとえば文字列に関しての問合せは次の XQuery で指定できる．次の例では<`news_item`>中のテキストに文字列 “Gorilla Corporation” を含む場合，<`item_summary`>という構造を新たに作成する．

```
For $item in doc(‘‘string.xml’’)//news_item
Where contains(string($item/text()),
                    ‘‘Gorilla Corporation’’)
Return
    <item_summary>
         $item/title/text() .
         $item/date/text() .
         string(($item//par)[1])
    </item_summary>
```

XQuery を実装したプロトタイプシステム，QEXO☆2，Quip☆3，RainbowCore☆4なども出現している．

4.2 XQuery の全文検索機能拡張提案

W3C では，さらにデータ系 XML の問合せに加え，文書系 XML の検索も XQuery で扱える必要性を認め，XQuery に全文検索機能の追加に関する提案を行っている[29]．全文検索機能としての最小限必要な機能として，単語検索，フレーズ検索，不要語処理のサポート，接頭語指定，接尾語指定，単語を単位とする近接検索，順序を指定した近接検索，論理積 AND，論理和 OR，論理否定 NOT，語の正規化，語尾処理や語の区切り処理，順位付け，関連性をあげている．この提案における XQuery の主要な拡張は，全文検索の述語と関連度を表すスコア関数であるが，スコア関数については各システムの実装にまかされている．

さらに，全文検索機能について例を示して説明している[30]．この中にはキーワード検索として辞書，シソーラス，分類表の利用を前提とした例もある．これらは情報検索研究者 Myaeng[21]や後述する Schlieder[31]，Fuhr[32]などの研究を参考として作成されている．また XQuery の全文検索拡張を実装した TeXQuery[33),34]も公開されている．

この提案では XML 問合せ言語が必要としている全文検索機能を想定しているだけで，キーワードを利用した XML 文書検索が必要としている機能のごく一部について言及したにすぎない．すなわち，文書構造の要素と文書内容のテキスト両者にキーワードを適用したり，解答候補となる要素や部分文書を指定しない検索は想定されていない．このように XQuery は構文が難しいにもかかわらず，XML サーチエンジンのための検索言語としては機能が不足しているため，もっと

☆1 http://www.w3.org/TandS/QL/QL98/

☆2 http://www.gnu.org/software/qexo/

☆3 http://developer.softwareag.com/tamino/quip/

☆4 http://davis.wpi.edu/~dsrt/raibow/

簡単な検索方法でありながら必要な機能を満載した検索言語への要求が強い．

4.3 XML 問合せ言語とキーワード検索の統合利用

本節では，XQuery 以外の XML 問合せ言語に対してのキーワード検索機能拡張に関する研究を紹介する．これらの研究では検索キーワードと文書内のテキストの照合にあいまい性を加味する手法を提案している．

2000 年，Florescu らは XML 問合せ処理とキーワード検索を統合する手法を提案した[35]．この研究では XML 問合せ言語 XML-QL[24),25)]を拡張し，XML データ問合せとキーワード検索との統合利用を目指した．文字列に関してあいまい性を許すキーワード検索を想定しているが，検索結果として取り出す部分の要素を利用者が指定する必要がある．

この提案での問合せは次のように表すことができる．ここで導入した`like`はパターンマッチによる文字列比較を行うものである．

“Dingle によって書かれた 1999 年の Web に関するすべての論文を取り出す．”

```
WHERE <article>
        <author><name>$N</name></author>
        <title>$T</title><year> 1999 </year>
      </article>
      ELEMENT_AS $E IN ‘‘bib.xml’’,
      $N like *Dingle*, $T like *web*
CONSTRUCT $E
```

この研究によりデータ系 XML 文書に対する問合せ処理だけではなく，指定キーワードと文書内容のテキストの照合にあいまい性を認める文書系 XML 文書の検索に関する研究が注目されるようになった．しかしここで想定されているのは完全一致検索であり，解答候補の順位付けは想定されていない．

4.4 テキスト類似演算子の導入

2000 年，Theobald らが提案した flexible XML search language（XXL）は SQL 形式の検索言語だが，類似条件による拡張を行い，解答候補の順位付けを行う[36)~38)]．この言語では類似演算子 “~” を導入し，経路式の指定には正規表現を導入することで問合せ中の構造条件のあいまい性を表す．検索条件として指定された要素名，属性名，テキストに対してオントロジやシソーラスを利用して，検索キーワードの文書中のテキスト内に出現した語の確率を求めて順位付けを行う．

XXL の問合せ例を以下に示す．

誰が Art ensemble of Chicago 演奏の CD でバリトンサックスを演奏しているのか？

```
Select M, T
From http://my.cdcollection.edu/allcds.xml
Where ~cd As C
  And C.#.artist As A
  And A=‘‘Art ensemble of Chicago’’
  And C.#.(track)? As T
  And T.~musician As M
  And M.# ~ ‘‘bass saxophone’’
```

XXL では，XML 文書中の要素名が表す意味を，オントロジを利用して構造の類似性に利用している点が特徴である．

2001 年に Chinenyanga らが提案した ELIXIR[39),40)]は，XML-QL[24)]にテキスト類似演算子 “~” を導入して機能拡張した XML 情報検索のための検索言語である．解答候補は検索キーワードとの関連性の高い順に並べられた XML 文書のノードである．文書内容の類似性はベクトル空間モデルを適用して求め，解答候補数は検索時に指定する．

ELIXIR の検索言語例を以下に示す．

1) 本と CD のタイトルデータベースから “Ukrainian cookery” に似たフレーズの `item` *を探す*

```
CONSTRUCT <item>$t</>
WHERE <items.(book|cd)>$t</> in ‘‘db.xml’’,
      $t ~ ‘‘Ukrainian cookery’’.
```

2) 本と CD の似たタイトルの `item` *を探す*

```
CONSTRUCT <item>$b</>
WHERE <items.book>$b</> in ‘‘db.xml’’,
      <items.cd>$c</> in ‘‘db.xml’’,
      $b ~ $c.
```

ELIXIR では上記の 2) のような similarity join が可能であることが特徴である．検索処理は，ELIXIR 構文を XML-QL に変換し，さらに順位付け検索が可能なデータベース問合せ言語 WHIRL[41)]に変換し処理する．さらに効率的な検索アルゴリズムを提案している．

本節で紹介した XML 問合せ処理とキーワード検索を統合する処理は，半構造データベース中のテキストや XML の要素に対する処理を可能にするが，いずれも利用者は前もって検索対象 XML 文書の文書構造に対する知識と複雑な問合せ言語に対する知識を必要とする．しかしデータベース技術を利用していること

で，大量の XML 文書に対して入力キーワードと一致あるいは類似したテキストを高速に探し出すことができる．XQuery はデータ系 XML 文書の問合せにおいて果たす役割が大きく，今後は XQuery を基本としつつ，XQuery では不足なキーワード検索機能を追加したアプリケーションとして検索言語が出現することが予想される．

5. XML サーチエンジンに関する提案

本章では，情報検索研究者の研究を中心として，XML サーチエンジンとその検索言語の提案について紹介する．

5.1 情報検索分野における XML 文書検索の動向

2000 年以降，情報検索の分野では多くの国際会議などで XML 文書を対象とした情報検索が議論されるようになっている．2000 年夏に行われた情報検索に関する国際会議 The 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval（SIGIR）では XML と情報検索についてのワークショップが併催され，情報検索の研究の観点から XML 文書に対する検索，索引付けなどについて議論された☆．そこでは XML 検索パラダイムとは何か，XML 検索の有効な解答とは何かが議題となり，文書内容と構造に基づく順位付け方法や効率的なデータ構造などが研究課題であることが確認された[42]．

5.2 順位付き XML 全文検索

2000 年の SIGIR の XML と情報検索についてのワークショップで，構造化文書をランキング可能な全文検索システムを富田らが提案した[43),44)]．XML 文書の検索においては，文書構造と文書内のテキストに記述された内容に基づく検索結果の順位付け機能を実現することが重要であり，全文検索システムの機能として考慮すべき点は，（1）部分構造も含めた文字列検索，（2）順位付け機能，（3）テキスト以外の型（日付や数値など）の情報への対応，（4）単語抽出処理，の 4 点であると述べた．

このシステムでは，索引作成者があらかじめ検索対象とする XML 文書の部分構造を指定し，索引ファイル形式，単語抽出処理形式をフォーマットファイルに格納し，最終的にインデクサがフォーマットファイルに基づき複数の索引ファイルを ID 付きで作成する．検索言語では検索範囲を指定した重み付きキーワード指定と文書全体に対するキーワード指定を組み合わせて指定できる．また，このシステムでは，検索範囲ごとに関連度のスコアを計算することができる．さらに OR 結合はそのスコアの和を AND 結合はそのスコアの最大値をとることで，文書全体の関連度を計算して順位付き結果を出力できる．

提案システムの検索言語の例を以下に示す．

文書全体に重み *0.8* で *“UNIX”* を含み，しかも要素 `title` には重み *0.3* で *“network”*，重み *0.8* で *“TCP/IP”* を，要素 `topic` に *“IT”* か *“computer systems”* を含む文書を求める

```
( UNIX~0.8
  and title=( network~0.3 or TCP/IP~0.8 )
  and topic=(IT or ‘computer systems’ ))
```

このシステムは単一の DTD に従った XML 文書集合で検索対象部分構造の種類が少ないときに有効である．

5.3 XML 情報検索言語：XIRQL

2000 年，Fuhr らは情報検索に関係する重み付け，順位付け，関連性指向検索や構造類似性などを取り入れた XML 情報検索言語 XIRQL[32),45)]を提案した．XIRQL は XQL[23)]に述語 contains word（`cw`）を導入し，指定した要素の文書内容に指定した語が含まれるか否かを求める．

検索方法として，（1）文書構造を指定せず，内容に基づき文書内の論理構造から関連する部分文書を抽出する検索（CO 検索）と，（2）取り出す要素を指定する内容と構造に基づく検索（CAS 検索）を想定している．さらに（3）特定のデータ型として区別される部分についての類似検索も想定している．検索言語の例を以下に示す．

1）要素 `heading` の文書内容に *“XML”* を含むかあるいは要素 `section` の下位にある要素の文書内容に *“XML”* を含む `document` を求める

```
/document[.//heading cw ‘‘XML’’
          or .//section//* cw ‘‘XML’’]
```

2）要素 `section` の下位の要素の文書内容に *“XML”* を *0.6* の重みで *“syntax”* を *0.4* の重みで含む `section` を求める

```
//section[0.6・. //* cw  ‘‘XML’’
         + 0.4・. //* cw ‘‘syntax’’]
```

XIRQL は pDatalog に基づいており，各文書は EDB 述語集合に XIRQL 問合せは IDB 述語と pDat-

☆ http://www.haifa.il.ibm.com/sigir00-xml/

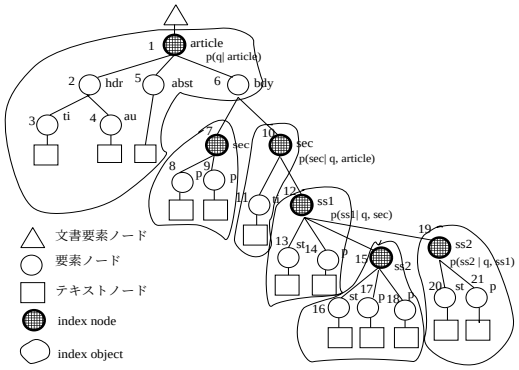

図 6 XML 文書を index object へ分割
Fig. 6 The division into index objects of the XML document.

alog 問合せに変換される．

この研究で初めて CO 検索と CO 検索に関する検索手法についての提案が行われた．CO 検索の検索対象となる部分文書の選定のため，XML 文書の論理木構造において index object を定義した．テキストは木構造の葉ノードにあり，それ自体で検索対象部分文書になりうるが，すべての葉ノードを検索単位とするには，文書としての粒度が小さい．検索対象として，適当な粒度の部分文書を選定する目的で，index object となるノードを選ぶ．また語の重み付けの観点から index object は互いに重複すべきではないが，入れ子関係を認めている．index object は論理木構造の部分木でその根ノードを index node と呼ぶ．システム管理者が一番典型的な index node を指定して索引付けを行い，さらに木構造を上にたどって他の index object になっていないテキストだけをまとめて索引付けする．各 index node を検索対象部分文書と考え，各 index node が持つ文脈中での語の重みをその部分木中の文書内容から求める．その方法は一般的な $tf \cdot idf$ を利用している．語の重みを語がその文脈上に持つ確率事象と考え，問合せ全体の確率を関連性の指標としている．**図 6** は図 3 を index object に分割した例である．

さらに XIRQL を実装した検索エンジン Hypermedia Retrieval Engine for XML（HyREX）を公開している[46]．

XIRQL では，CO 検索の必要性を提示したことが特徴である．しかし検索対象となる XML 文書の文書構造をあらかじめ分析して index object を指定する必要がある．

これらの議論を土台として 2002 年の春には，XML と情報検索に関する特集が米国の情報科学技術に関する学会の論文誌，Journal of the American Society for Information Science and Technology（JASIST）[47] に掲載された．さらに，2002 年夏の SIGIR のワークショップにおいても，再び XML 文書に対する情報検索について議論された．このワークショップでのパネル討論では「XML 検索は意味をなすのか？」，「XML 検索で取り出すものは何か？」という主題に対して議論された[48]．

5.4 構造類似性の導入：ApproXQL

次に構造と内容を指定した問合せ木と XML 文書の木構造の類似性を測定して解答候補の順位付けを行う手法を紹介する．

2001 年，Schlieder らは Kilpeläinen の tree inclusion と従来のベクトル空間モデルとを問合せベクトルの重み調整で結び付ける手法を提案し，検索言語 ApproXQL を提案した[31),49]．ApproXQL は XQL[23] を拡張した検索言語である．次に示すのが検索言語の例である．

タイトルにピアノ，コンチェルトがあり作曲が *“Rachmaninov”* の *CD* を取り出す．

```
cd[ title[‘‘piano’’ $and$ ‘‘concerto’’] $and$
    composer[‘‘Rachmaninov’’]]
```

XML 文書の各部分木に対応する部分文書を論理文書と考え検索言語の木構造表現と文書を比較して類似度を計算するため，従来の文書内容の語の $tf \cdot idf$ に加え，各論理文書中の要素名を structural term としてその出現回数と structural term ごとの出現論理文書数による $tf \cdot idf$ を利用する．木構造マッチの条件を緩め論理文書中に少なくとも 1 つの部分木で問合せ木と部分一致している場合を認め，その部分一致している部分木の数も類似度計算に利用する．またすべての tf 値を索引作成時に計算するのではなく問合せ実行時に必要な計算のみを行う．T を structural term，D を論理文書，$freq_T(D)$ を D 中での T の出現回数，$\max freq(D)$ を D 中の structural term の最大出現回数として T の tf 値 $tf_{T,D}$ を次のように定義する．

$$tf_{T,D} = \frac{freq_T(D)}{\max freq(D)}$$

またある要素（type）t について，$|D^t|$ を t の論理文書数，n_T を T で一致した D^t 中での文書数として T の idf 値 idf_T^t を次のように定義する．

$$idf_T^t = \log \frac{|D^t|}{n^T} + 1$$

その結果，文書の重みは $tf_{T,D}$ と idf_T^t の積として

求める．

このシステムは構造の類似性を従来の $tf \cdot idf$ を用いて利用していることが特徴である．

5.5 キーワード近接性の導入：XRANK

2003年，Guoらはハイパーリンク構造を持つXML文書に対するキーワード検索を効率的に行うXRANKシステムを提案した[50]．このシステムではキーワード検索の結果は文書全体ではなくキーワードを含む入れ子構造の深い部分に存在し，しかもその検索単位の粒度は要素ノードとしている．近接性の測定は，文書中の出現キーワード間の距離とキーワードと結果要素の間の距離の両者を利用する．さらに入れ子構造である文書構造の意味的なリンクも考慮した順位付けの手法を提案した．

XRANKでは検索言語をキーワードのAND結合で与え，解答候補としてすべての検索指定キーワードについて少なくとも1回出現している要素ノード集合を抽出する．

たとえば，検索指定 $Q = (\text{“}XML\text{”}, \text{“}workshop\text{”})$ では，“XML”と“workshop”を直接あるいは間接的に含んでいる要素集合が解答候補である．したがって，この要素ノード集合中のノードのすべての祖先ノードも検索対象になり，解答候補になりうる．さらにその子孫ノードがすべてのキーワードを含んでいる場合には子孫ノードがよりspecificな結果であると考える．そこで検索キーワードが出現しているテキストを子孫ノードに含む要素について出現キーワードに関する順位付け関数 $ElemRank$ を定義する．この関数はgoogleのPageRank[51]に似た方針でキーワードを含むテキストに近いノードの値が高く，その親，祖先ノードと階層構造を上がるほど値が減少する．さらにキーワードを含むテキストを子孫ノードに含むノードについてキーワード近接性関数を定義し，文書内のハイパーリンクに関してそのノードにランダムに訪れる確率，他の文書からのリンクで訪れる確率，親子関係にある経路の確率の和として全体の順位を求める．また効率的な索引構造についても論じている．しかし，検索結果の性能評価は行っていない．

XRANKの特徴は，検索言語でキーワードを要素名とテキストで区別しないこと，さらに検索結果はなるべくspecificな要素が上位になり出現キーワードが密集していてより多く他から参照されている場合に順位が高くなるように設計されていること，である．

5.6 キーワード出現位置に応じた関連性の導入：XSEarch

2003年，Cohenらが提案したXSEarch[52]では初心者向きの検索言語を開発している．指定キーワードを含むテキストノードを直接子に持つ要素ノード間に，解答候補として文脈上意味的に関連がある（meaningfully related）だけでなく，キーワードへの関連度が高い部分文書を解答候補として抽出し，その解答候補に順位付けを導入した．検索言語はキーワードリストで表現するが，各々のキーワード指定は文書内容に関するキーワード，文書構造に関する要素名，要素名とキーワードの組のいずれかで指定する．次に示すのが検索言語の例である．

> *Vianu*が書いた論理データベースに関する論文を探す．
>
> `logical +database inproceedings:author:vianu`

検索キーワードを含むテキストノードを子に持つ要素ノードに関して要素ノードのラベルを手がかりに意味的な関連を定義する．Cohenらは，XML文書中の2つのノードの出現位置によって意味的な関連の強さを計測する．2つのノードラベルが同じ場合は共通祖先があるとき，また異なる場合は最小共通祖先と各ノード間に同じ名前をラベルに持つ異なったノードを含まないときを *interconnected*な関係にあると定義した．そして，Cohenらは検索条件を満たす文書中のノード間の関連を調べて解答候補となりうる部分を特定するアルゴリズムを提案した．XSEarchでは解答候補は文書の部分木である．キーワードは文書の葉ノードのテキストと比較するので，キーワードの重みは葉ノードで計算する．その計算は $tf \cdot idf$ の変形の $tf \cdot ilf$（inverse leaf frequency）を利用する．k をキーワード，n_l を葉ノード，$occ(k, n_l)$ を n_l 中に出現する k の頻度とし，N をコーパス中のすべての葉ノード集合とすると，$tf(k, n_l)$ と $ilf(k)$ は，

$$tf(k, n_l) = \frac{occ(k, n_l)}{\max\{occ(k', n_l) | k' \in words(n_l)\}}$$

$$ilf(k) = \log\left(1 + \frac{|N|}{|\{n' \in N | k \in words(n')\}|}\right)$$

と表される．

検索言語と解答候補の部分文書の類似度は両者のベクトルのコサイン相関値と解答部分文書中のノード数と祖先子孫関係にあるノード組数を組み合わせて求める．

このシステムの特徴は検索キーワードの出現位置からノード間の関連性を調べることと，解答候補の文書の大きさをノード数で計測していることである．

5.7 意味的にまとまりのある XML 部分文書：MPD

キーワードを利用した XML 文書検索，特に CO 検索では，利用者にとって意味的にまとまりのある XML 部分文書を解答候補として取り出すことが重要である．我々はこのような部分文書を Meaningful Partial Document（MPD）と呼び，MPD を検索対象として，入力キーワードに対して利用者が結果として想定している XML 部分文書（Retrieved Partial Document: RPD）を取り出すシステムを開発している．XML 文書中には検索対象となる部分文書は文書中の要素ノードに対応する数だけ存在する．検索対象 XML 文書の文書構造が多様な場合，システム管理者が MPD を識別することが困難である．そこで，MPD と MPD 以外を識別することによって検索対象部分文書数を削減する手法を提案してきた．

我々は 2001 年に XML の文書構造を利用して入力キーワードを含む最小の部分文書 Coherent Partial Document（CPD）を決定する手法を提案した[6),7)]．この手法では CPD は XML 文書の論理構造中に存在する同名の兄弟要素が文書内容の境界として機能しているという経験則を用いて決定していた．しかし，XML 文書内には文書の論理構造だけではなく語の強調やリンクのアンカなどに用いられる要素ノードも多数存在するため，RPD として抽出されるべき XML 部分文書や MPD を抽出しきれないなどの課題があった．この問題点を改善するため，XML 文書の持つ統計量，すなわち文書内に出現する語数，異なり語数や検索キーワードを含む部分文書数を用いて MPD を決定する手法を 2003 年に提案した[53),54)]．

さらに我々は統計値の持つ意味を計量情報学を利用して判別し，検索時間短縮の効果および検索精度向上について実証実験を行った[55)]．我々の研究では，多様な文書構造の XML 文書集合を対象とし，文書構造のスキーマ定義を必要としない処理方法を提案していることが特徴である．

5.8 Focussed Retrieval and Best Entry Points

次に紹介する研究では，XML サーチエンジンにおける利用者の検索行動として，検索結果の絞り込みや解答候補の表示方法も考慮にいれて考察している．Lalmas の研究[22)]に基づき 2002 年，Kazai らは問合せに関連する部分文書を取り出す構造化文書検索を “Focussed Structured Document Retrieval” と定義し，利用者が問合せの解答を見るエントリノードを “Best Entry Point”（BEP）と呼び，その条件を分析した[56)]．

この研究では利用者はより小さくより具体的な部分を抽出することを好み，さらに関連する文書中の文脈を見ることを好むという分析に基づいている．BEP の条件は，（1）問合せに対し多数の下位ノードが関連している場合は，下位ノードを個別に抽出する代わりに上位ノードを抽出する，（2）お互いに関連するノードが順番に並んでいる場合は（1）で抽出したノードのうちの最初のノードを結果とする，である．上位ノードのスコアはそのノード自身と下位ノードのスコアから導出する．

この研究では，利用者の検索目的によって解答候補も異なり，性能評価は従来の構造のない文書の検索結果の評価尺度である精度と再現率だけでは不十分であることを前提としている．この研究を行っている Kazai らは，6.2 節で紹介する INEX プロジェクトの中心メンバであり，自らの研究のためにもこのプロジェクトにおいて，新たな評価尺度を提案している．

6. 検索結果の評価方法に関する取組み

前章で述べたように，XML サーチエンジンに関する新たな手法の提案はこの 1〜2 年に多数出現している．さらにプロトタイプシステムを実装し，テストデータを利用してシステムの性能を検証しようと努力をしている．しかし，文書系の XML 文書で現在利用可能なテストデータは，文書の種類，文書の容量とも限られているため，検索システムの有効性を確認する手段がないのが現状である．そのため，XML サーチエンジンの性能評価には何が必要かをまず確認し，誰もが利用できる XML サーチエンジンのためのテストコレクションを作成することが急務となっている．研究の障壁となっているのは，文書構造が複雑で多様な構造の XML 文書集合がないこと，検索システムが出力する検索結果が利用者の検索意図と合致するかどうかを確認する指標がないこと，従来の情報検索システムの尺度である精度，再現率を直接適用することができないこと，である．

そのためにすべての XML サーチエンジン研究者が検索結果の評価方法に強い関心を持っている．

6.1 精度と再現率

従来の情報検索システムの検索性能を示す基本的な尺度は，「精度（precision）」と「再現率（recall）」である．精度は検索された文書集合中で利用者の検索要求に適合している文書の比率を，再現率はデータベース中のすべての適合文書の中で，実際に検索されたものの比率を示す[57)]．従来の情報検索では，利用者の検

```
CO トピック：ウェアラブルコンピュータに関する装置を記述している部分文書を探す.
<inex_topic topic_id="125" query_type="CO" ct_no="127">
<title>+wearable ubiquitous mobile computing devices </title>
<description>Wearable computing devices.</description>
<narrative> To be relevant, a document or component must contain information about wearable computers,
and it may contain information about mobile devices or ubiquitous computers, such as head mounted display,
Cellular phone, RFID and IrDA. </narrative>
<keywords>ubiquitous, device, wearable, mobile, equipment, GPS, augment, reality</keywords>
</inex_topic>

CAS トピック：1998 年以降に執筆された画像検索に関する論文を探す.
<inex_topic topic_id="65" query_type="CAS" ct_no="23">
<title>//article[./fm//yr > '1998' AND about(., '"image retrieval"')]</title>
<description>Find articles about image retrieval that are written after 1998.</description>
<narrative>Relevant documents are research articles about image retrieval written after 1998.
Editorials, news or volume indices are not relevant.  </narrative>
<keywords>image retrieval, retrieve images</keywords>
</inex_topic>
```

図 7 INEX2003 トピック例
Fig. 7 INEX2003 sample topics.

索要求に対し検索結果が適合しているか否かの真偽値で各検索結果の文書内容と検索要求を照合していた．これは検索結果がつねに文書を単位としていることが前提となっている．

XML 文書検索では，利用者の検索要求が従来の構造のない文書に対する要求より複雑で多岐にわたる．また検索要求も検索結果も部分文書を単位に照合することができるため，新たな評価方法を提案する必要がある．

6.2 INEX プロジェクト

XML 文書についての検索システムと検索手法の評価を目的とした Initiative for the Evaluation of XML retrieval（INEX）プロジェクト☆が 2002 年 4 月から始まった．XML 文書，検索質問と各質問に対する解答部分文書集合から構成されるテストコレクションを作成している．対象 XML 文書は，IEEE Computer Society's Journal publications（1995 年から 2002 年）の論文 12,107 件を XML 化したものである．容量約 494 メガバイトで，各論文には平均 1,532 要素ノード，6.9 階層を構成している．一昨年行われた INEX2002 では，我々を含め 49 グループが参加し，2 種類のトピックについて各々30 の質問とその適合文書集合を求める作業を行った．また昨年の INEX2003 でも 47 グループが参加しており，テストコレクションの改良が進められている☆☆．

トピックは文書内容を指定する 'content only'（CO）と文書構造と文書内容を指定する 'content and structure'（CAS）に分けられ，文書構造は XPath 1.0 の一部を利用し，文書内容については従来の情報検索の類似度計算アルゴリズムと同等の意味を持つ関数 `about` を導入する．CO トピックについては，文書構造を指定しないため `about` 関数を省略しているが，指定キーワードは `about` 関数の引数として与えられる．**図 7** は INEX2003 におけるトピックの一例である．`title` 要素内容が検索言語の記述である．CO トピックの検索目的は，関連性の高い部分文書を探すことであり，問合せに関連する部分文書は検索システムが自動的に特定する．CAS トピックでは XPath で指定された文書構造条件を満たし，問合せキーワードと関連する部分文書を根ノードの XPath として取り出す．ただし，`about` 関数内の指定キーワードの文書中での出現は必須ではない．

解答候補の評価は，文書単位ではなく部分文書単位で行う．INEX2002 の評価作業では，トピックとの関連性を not，marginally，fairly，highly の 4 段階で，

☆ http://www.is.informatik.uni-duisburg.de/projects/inex/index.html.en

☆☆ http://inex.is.informatik.uni-duisbrg.de:2003/

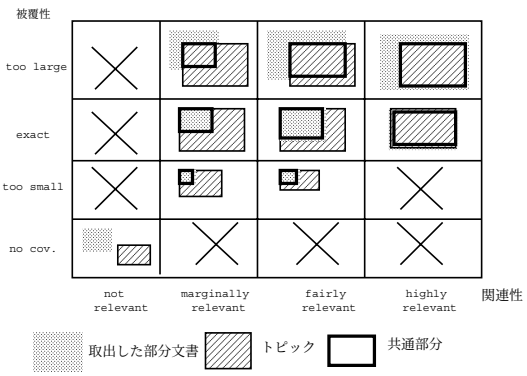

図 8 INEX2002 における関連性と被覆性の関係
Fig. 8 Relationship between relevance and coverage of INEX2002.

表 3 関連性評価結果の集計
Table 3 A summary of relevance assessments.

	CAS トピック		CO トピック	
関連性/被覆性	article	その他	article	その他
highly/exact	187	2,304	307	1,087
fairly/exact	59	1,128	165	1,107
marginally/exact	82	1,770	114	827
highly/large	173	424	394	1,145
fairly/large	137	507	599	2,295
marginally/large	236	719	854	2,708
fairly/small	21	846	118	3,825
marginally/small	54	1,119	116	3,156
合計	949	8,817	2,667	16,150

部分文書の被覆性を not，small，exact，large の 4 段階で与えた．次に示すのが関連性，被覆性の概念である．

$$\text{関連性}=\frac{|\text{トピックが要求する情報}\cap\text{部分文書内の情報}|}{|\text{トピックが要求する情報}|}$$

$$\text{被覆性}=\frac{|\text{トピックが要求する情報}\cap\text{部分文書内の情報}|}{|\text{部分文書内の情報}|}$$

図 8 は，検索結果として取り出した部分文書内容とトピック内容との関係を表している．

表 3 は 2002 年の評価作業の結果である[58]．対象 XML 文書が論文であるため，各論文を表す要素 `article` を検索単位と見なすことは，従来の情報検索と同様の検索結果をもたらす．しかし，関連性が高く被覆性も過不足ない，すなわち関連性（highly）と被覆性（exact）と判定される割合は CAS トピックでは高いが，CO トピックでは低い．これは，CAS トピックでは問合せ中に検索対象文書構造を指定していることによって検索対象が絞られていることを示している．一方 CO トピックでは `article` 部分文書の被覆性が large と判定された割合が高く `article` より粒度の小さな部分文書を正解とする場合が多いことを示している．各検索システムの検索結果の精度，再現率の計算に strict な判定としては関連性（highly）と被覆性（exact）と判定された部分文書だけを正解（表 3 の下線の値）とし，generalized な判定としてはその他の関連性，被覆性の値を持つ部分文書について，半正解として扱うことにした．検索結果の部分文書間に入れ子関係がある場合でも，すべての部分文書を独立なものとして扱った．

7. 今後の展望と研究課題

本論文では，XML サーチエンジンが必要としているキーワードを利用した XML 文書検索について，情報検索研究の観点から研究動向について調査した．**図 9** は XML サーチエンジンに関する研究動向の一覧である．この図から本論文で紹介した研究者の多くが INEX プロジェクトへ参加していることが分かる．XML サーチエンジンが必要としている主な研究課題，検索方法と検索結果の評価方法については INEX プロジェクトの活動が果たす役割が大きい．また，検索結果抽出のための順位付け手法に関する研究は，各研究者の研究課題としてすでに多くの提案がなされている．しかし CO 検索における検索対象となる部分文書の粒度決定方法や解答候補指定，利用者が期待する解答候補文書の特定やその表示手法に関する研究は，多くない．さらに，XML サーチエンジン向けの検索言語も試行段階であり，今後さらに様々な提案が出現するとみられる．

Kamps らは，XML 文書検索の検索単位は文書中に出現するすべての要素であること，固定ではないことがもたらす困難さを 2003 年の SIGIR で述べている[59]．

2003 年 12 月 15 日から 3 日間 INEX のワークショップがドイツで開催された．本論文で紹介した研究者らも参加し，41 名によって現在の XML サーチエンジンの問題点について議論を行った．議論の主な焦点は，CAS トピックに関する検索に簡潔に文書構造を制約条件として指定する検索言語と，順位付けを行う文書範囲の限定方法である．また解答候補中に多数存在する入れ子関係にある部分文書の扱い方であった．解答候補の評価尺度については，INEX2002 では本論文で紹介した関連性と被覆性を用い，INEX2003 では Lalmas[22] が提案していた exhaustivness と specificity を用いた．しかし，これらの評価尺度が二次元であることから，この二次元の値を解答候補の総合的な性能評価（精度と再現率）にどのように反映させるかに関し疑問を持つ意見もあり，試行錯誤が続く模様である．

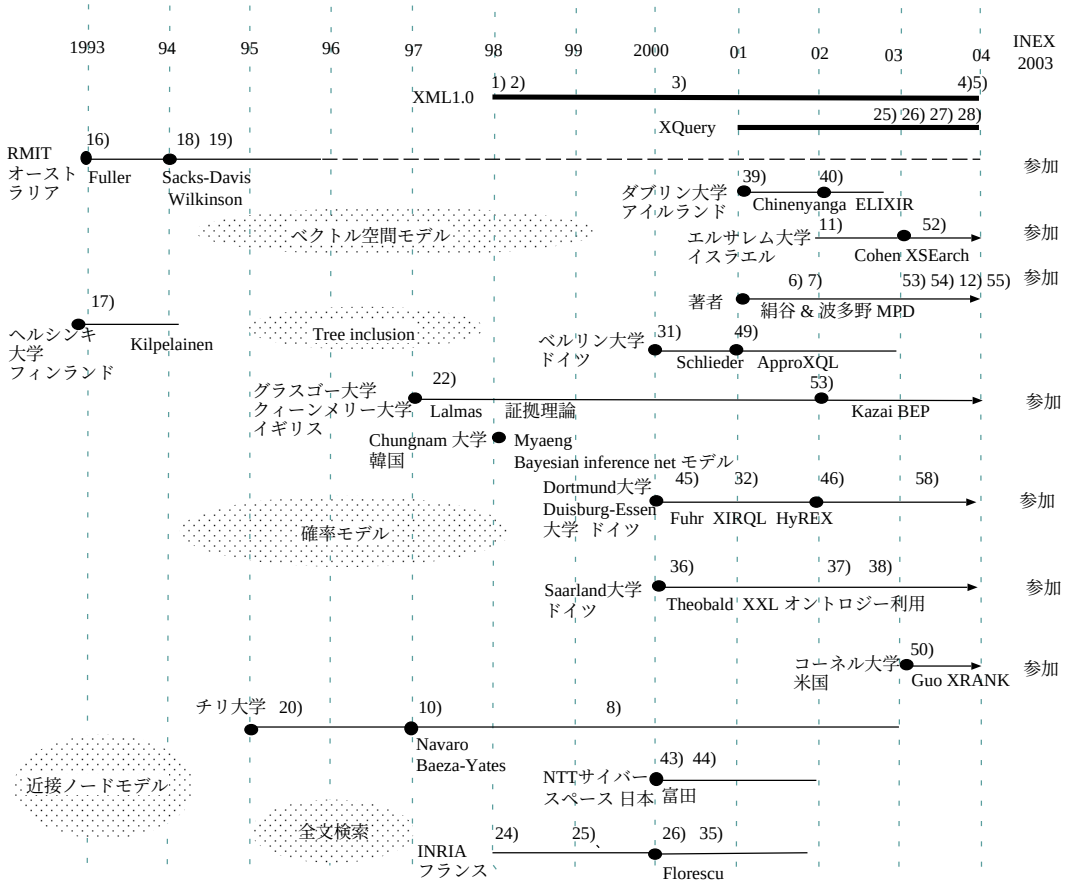

図 9 XML サーチエンジンに関する研究動向
Fig. 9 The research trend about XML search engines.

2004 年 4 月から 12 月まで，INEX2004 プロジェクトが引き続きテストコレクション作成の活動を行っている．INEX2004 では，今までに議論することのなかった 4 つの分野，関連性フィードバック，自然言語処理，多様な文書集合，対話処理，に関する活動を行う予定である．

本論文で紹介した XML サーチエンジンの基盤技術，検索指定方法，解答候補となる部分文書の決定方法，検索結果の順位付けと評価方法，に関する研究に加え，今後次のような研究が必要である．

- 部分文書管理と部分文書単位の検索に適した索引のデータ構造と格納方法
- 高速な検索処理
- XML 名前空間で識別される各種 XML ボキャブラリを考慮した検索
- ネットワーク上に分散した XML 文書の管理と検索処理

Web サーチエンジンは，利用者の人気を得ることで，技術革新が進んできた．XML サーチエンジンについても，利用者に支持されるシステムの出現により，急速な進歩をとげることになるまでは，本論文で紹介したような基盤技術の研究の蓄積が続くと考えられる．

謝辞 本研究の一部は，日本学術振興会（課題番号 14780325），文部科学省科学研究費補助金（課題番号 15017243）の支援および，科学技術振興機構の戦略的基礎研究推進事業「高度メディア社会の生活情報技術」プログラムの支援によるものである．本論文執筆にあたり，第 1 著者に研究環境を提供いただいたお茶の水女子大学増永良文教授に感謝いたします．

参 考 文 献

1) TR X 0008：1998 拡張可能なマーク付け言語 XML（eXtensible Markup Language），日本規格協会 (1998).
2) W3C: Extensible Markup Language (XML) 1.0 (1998). http://www.w3.org/TR/1998/

REC-xml-19980210. W3C Recommendation 10 February 1998.
3) W3C: Extensible Markup Language (XML) 1.0 (Second Edition) (2000). http://www.w3.org/TR/2004/REC-xml-20001006. W3C Recommendation 06 October 2000.
4) W3C: Extensible Markup Language (XML) 1.0 (3rd Edition) (2004). http://www.w3.org/TR/2000/REC-xml-20040204. W3C Recommendation 04 February 2004.
5) W3C: Extensible Markup Language (XML) 1.1 (2004). http://www.w3.org/TR/2004/REC-xml11-20040204. W3C Recommendation 04 February 2004.
6) 絹谷弘子，波多野賢治，吉川正俊，植村俊亮：XML 文書の文書構造と内容を用いた部分文書の抽出手法，情報処理学会論文誌：データベース，Vol.43, No.SIG2(TOD13), pp.80–93 (2002).
7) Hatano, K., Kinutani, H., Yoshikawa, M. and Uemura, S.: Extraction of Partial XML Documents Using IR-based Structure and Contents Analysis', *Proc. International Workshop on Data Semantics in Web Information Systems* (*DASWIS-2001*) (2001).
8) Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Addison Wesley (1999).
9) Burkowski, F.J.: Retrieval Activities in a Database Consisting of Heterogeneous Collections of Structured Text, *Proc. ACM SIGIR '92*, pp.112–125 (1992).
10) Navarro, G. and Baeza-Yates, R.A.: Proximal Nodes: A Model to Query Document Databases by Content and Structure, *Inf. Syst.*, Vol.15, No.4, pp.400–435 (1997).
11) Cohen, S., Kanza, Y., Kogan, Y.A., Sagiv, Y., Nutt, W. and Serebrenik, A.: EquiX — A search and query language for XML, *Journal of American Society for Information Science and Technology* (*JASIST*), Vol.53, No.6, pp.454–466 (2002).
12) Hatano, K., Kinutani, H., Watanabe, M., Mori, Y., Yoshikawa, M. and Uemura, S.: Keyword-based XML Fragment Retrieval: Experimental Evaluation based on INEX 2003 Relevance Assessments, *Proc. 2nd Workshop of the Initiative for the Evaluation of XML Retrieval* (*INEX*) (2004). http://inex.is.informatik.uni-duisburg.de:2003/proceedings.pdf
13) ISO: ISO 8879: 1986. Information Processing — Text and Office System — Standard Generalized Markup Language (SGML) (1986).
14) JIS X 4151：1992 文書記述言語 SGML（Standard Generalized Markup Language），日本規格協会 (1992).
15) JIS X 4151：1998 文書記述言語 SGML（Standard Generalized Markup Language）（追補 1），日本規格協会 (1998).
16) Fuller, M., Mackie, E., Sacks-Davis, R. and Wilkinson, R.: Structured Answers for a Large Structured Document Collection, *Proc. ACM SIGIR'93*, pp.204–213 (1993).
17) Kilpeläinen, P. and Mannila, H.: Retrieval from hierarchical texts by partial patterns, *Proc. ACM SIGIR'93*, pp.214–222 (1993).
18) Sacks-Davis, R., Arnold-Moore, T. and Zobel, J.: Database Systems for Structured Documents, *International Symposium on Advanced Database Technologies and Their Integration* (*ADTI'94*), pp.272–283 (1994).
19) Wilkinson, R.: Effective Retrieval of Structured Documents, *Proc. ACM SIGIR'94*, pp.311–317 (1994).
20) Navarro, G. and Baeza-Yates, R.: A Language for Queries on Structure and Contents of Texual Databases, *Proc. ACM SIGIR '95*, pp.93–101 (1995).
21) Myaeng, S.H., Jang, D.H., Kim, M.S. and Zhoo, Z.C.: A Flexible Model for Retrieval of SGML Documents, *Proc. ACM SIGIR'98*, pp.138–145 (1998).
22) Lalmas, M.: Dempster-Shafer's theory of evidence applied to structured documents: modelling uncertainty, *Proc. ACM SIGIR'97*, pp.110–118 (1997).
23) Robie, J.: XML Query Language (XQL) (1998). http://www.w3.org/TandS/QL/QL98/pp/xql.html
24) Deutsch, A., Fernandez, M., Florescu, D., Levy, A. and Suciu, D.: XML-QL: A Query Language for XML (1998). http://www.w3.org/TR/NOTE-xml-ql/
25) Deutsch, A., Fernandez, M., Florescu, D., Levy, A. and Suciu, D.: A Query Language for XML, Vol.31, No.11-16, pp.1155–1169 (1999).
26) Chamberlin, D.D., Robie, J. and Florescu, D.: Quilt: An XML Query Language for Heterogeneous Data Sources, to be published in Lecture Notes in Computer Science, Springer-Verlag (2000).
27) W3C: XQuery1.0: An XML Query Language (2003). http://www.w3.org/TR/xquery/. W3C Working Draft 12 November 2003.
28) W3C: XML Query (XQuery) Requirements (2003). http://www.w3.org/TR/xquery-requirements/. W3C Working Draft 12 November 2003.
29) W3C: XQuery and XPath Full-text Require-

ments (2003). http://www.w3.org/TR/xquery-full-text-requirements/. W3C Working Draft 02 May 2003.
30) W3C: XQuery and XPath Full-text Use Cases (2003). http://www.w3.org/TR/xmlquery-full-text-use-cases/. W3C Working Draft 14 February 2003.
31) Schlieder, T. and Meuss, H.: Result Ranking for Structured Queries against XML Documents, *DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries* (2000).
32) Fuhr, N. and Grossjohann, K.: XIRQL: A Query Language for Information Retrieval in XML Documents, *Proc. ACM SIGIR'01*, pp.172–180 (2001).
33) Amer-Yahia, S., Botev, C. and Shanmugasundaram, J.: On The Completeness of Full-Text Search Languages for XML, Technical report, Cornell University (2003). http://www.research.att.com/~sihem/TeXQuery/Completeness.pdf
34) Amer-Yahia, S., Botev, C. and Shanmugasundaram, J.: TeXQuery: A Full-Text Search Language for XML, *13th International World Wide Web Conference* (2004). http://www.research.att.com/~sihem/TeXQuery/TeXQuery.pdf
35) Florescu, D., Manolescu, I. and Kossmann, D.: Integrating Keyword Search into XML Query Processing, *9th International World Wide Web Conference* (2000). http://www9.org/w9cdrom/start.html
36) Theobald, A. and Weikum, G.: Adding Relevance to XML, *Proc. 3rd International Workshop on the Web and Databases, WebDB2000* (2000). http://www.research.att.com/conf/webdb2000/program.html
37) Theobald, A. and Weikum, G.: The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking, Lecture Notes in Computer Science, Vol.2287, pp.477–495 (2002).
38) Theobald, A. and Weikum, G.: The XXL search engine: Ranked Retrieval of XML data using Indexes and Ontologies, *Proc. 2002 ACM SIGMOD*, pp.615–615 (2002).
39) Chinenyanga, T.T. and Kushmerick, N.: Expressive Retrieval from XML Documents, *Proc. ACM SIGIR'01*, New York, pp.163–171 (2001).
40) Chinenyanga, T.T. and Kushmerick, N.: An Expressive and Efficient Language for XML Information Retrieval, *JASIST*, Vol.53, No.6, pp.438–453 (2002).
41) Cohen, W.W.: Integration of Heterogeneous Databases without Common Domains using Queries based on Textual Similarity, *Proc. 1998 ACM SIGMOD*, pp.201–212 (1998).
42) Carmel, D., Maarek, Y. and Soffer, A.: XML and Information Retrieval: a SIGIR 2000 Workshop, *ACM SIGIR Forum*, Vol.34, No.1, pp.31–36 (2000).
43) Hayashi, Y., Tomita, J. and Kikui, G.: Searching Text-rich XML Documents with Relevance Ranking, *Proc. ACM SIGIR 2000 Workshop On XML and Information Retrieval*, Athens (2000). http://www.haifa.il.ibm.com/sigir00-xml/
44) 富田準二，菊井玄一郎，林　良彦：構造化文書をランキング可能な全文検索システム，情報処理学会データベースシステム研究報告，Vol.2000-DBS-122-47, pp.361–368 (2000).
45) Fuhr, N. and Grossjohann, K.: XIRQL An Extension of XQL for Information Retrieval, *Proc. ACM SIGIR 2000 Workshop On XML and Information Retrieval*, Athens (2000). http://www.haifa.il.ibm.com/sigir00-xml/
46) Fuhr, N., Gövert, N. and Großjohann, K.: HyREX: Hyper-media Retrieval Engine for XML, *Proc. ACM SIGIR'02*, pp.449–449 (2002).
47) Baeza-Yates, R., Carmel, D., Maarek, Y. and Soffer, A. (Eds.): *Journal of American Society for Information Science and Technology* (*JASIST*), Vol.53, No.6 (2002).
48) Baeza-Yates, R., Fuhr, N. and Maarek, Y.S.: 2nd Edition of the "XML and Information Retrieval" Workshop, *ACM SIGIR Forum*, Vol.36, No.2, pp.53–57 (2002).
49) Schlieder, T.: ApproXQL: Design and Implementation of an Approximate Pattern Matching Language for XML, Technical Report B 01-02, Freie Universität Berlin (2001). http://www.inf.fu-berlin.de/inst/ag-db/publications/2001/report-B-01-02.pdf
50) Guo, L. Shao, F., Botev, C. and Shanmugasundaram, J.: XRANK: Ranked Keyword Search over XML Documents, *Proc. 2003 ACM SIGMOD*, pp.16–27 (2003).
51) Brin, S. and Page, L.: The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, Vol.30, No.1–7, pp.107–117 (1998).
52) Cohen, S., Mamou, J., Kanza, Y. and Sagiv, Y.: XSEarch: A Semantic Search Engine for XML, *Proc. VLDB'03*, Berlin, Germany, Morgan Kaufmann, pp.45–56 (2003).
53) Hatano, K., Kinutani, H., Watanabe, M., Yoshikawa, M. and Uemura, S.: Determining

the Unit of Retrieval Results for XML Documents, *Proc. 1st Workshop of the Initiative for the Evaluation of XML Retrieval* (*INEX*), ERCIM, pp.57–64 (2003).
54) 波多野賢治，絹谷弘子，吉川正俊，植村俊亮：キーワードを利用した XML 文書検索のための検索結果粒度決定法，日本データベース学会 Letters, Vol.2, No.1, pp.123–126 (2003).
55) 波多野賢治，絹谷弘子，吉川正俊，植村俊亮：統計量を用いた XML 部分文書検索システムの実装，電子情報通信学会第 15 回データ工学ワークショップ（DEWS2004）(2004).
56) Kazai, G., Lalmas, M. and Roelleke, T.: Focussed Structured Document Retrieval, *Proc. 9th International Symposium on String Processing and Information Retrieval* (*SPIRE 2002*), Lisbon, Portugal, pp.241–247 (2002).
57) 山本毅雄，橋爪宏達，神門典子，清水美都子：全文検索，技術と応用，chapter 3, 学術情報センター（編），丸善株式会社 (1998).
58) Gövert, N. and Kazai, G.: Overview of the INitiative for theEvaluation of XMLretrieval (INEX) 2002, *Proc. 1st Workshop of the Initiative for the Evaluation of XML Retrieval* (*INEX*), ERCIM, pp.1–17 (2003).
59) Kamps, J., Marx, M., de Rijke, M. and Sigurbjornsson, B.: XML Retrieval: What to Retrieve?, *Proc. ACM SIGIR'03*, pp.409–410 (2003).

(平成 15 年 12 月 20 日受付)
(平成 16 年 4 月 17 日採録)

（担当編集委員　國島 丈生）

絹谷　弘子（正会員）

1976 年お茶の水女子大学理学部数学科卒業．2002 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．同年科学事業推進機事業団戦略的創造研究事業（科学技術振興機構戦略的基礎研究推進事業）研究補助員を経て，2004 年よりお茶の水女子大学総合情報処理センター研究支援推進員，現在に至る．構造化文書データベース，情報検索に関する研究に従事．ACM，日本データベース学会各会員．

波多野賢治（正会員）

1995 年神戸大学工学部計測工学科卒業．1999 年同大学大学院自然科学研究科博士後期課程修了．博士（工学）．日本学術振興会未来開拓学術研究事業研究員を経て，同年奈良先端科学技術大学院大学情報科学研究科助手，現在に至る．情報検索システム，データベースシステムに関する研究に従事．電子情報通信学会，ACM，IEEE Computer Society，日本データベース学会各会員．

吉川　正俊（正会員）

1980 年京都大学工学部情報工学科卒業．1985 年同大学大学院工学研究科博士後期課程修了．工学博士．同年京都産業大学計算機科学研究所講師．同大学工学部助教授，奈良先端科学技術大学院大学情報科学研究科助教授を経て，2002 年より名古屋大学情報連携基盤センター教授，現在に至る．1989 年〜1990 年南カリフォルニア大学客員研究員，1996 年〜1997 年ウォータルー大学客員准教授．XML データベース，多次元空間索引等の研究に従事．電子情報通信学会，ACM，IEEE Computer Society 各会員．日本データベース学会理事．

植村　俊亮（フェロー）

1964 年京都大学工学部電子工学科卒業．1966 年同大学大学院工学研究科修士課程修了．同年電気試験所（産業技術総合研究所）．1970 年マサチューセッツ工科大学電子システム研究所客員研究員，1981 年電総研ソフトウェア部プログラム研究室長，1988 年東京農工大学教授を経て，1993 年から奈良先端科学技術大学院大学情報科学研究科教授．データ工学，データベースシステムの研究に従事．工学博士．IEEE Fellow，電子情報通信学会フェロー．現在，情報処理学会理事，日本情報考古学会理事，データベース振興センター評議員等．