XML 文書の文書構造と内容を用いた部分文書の抽出手法

絹谷弘子[†] 波多野賢治[†]
吉川正俊[†,††] 植村俊亮[†]

XML の出現によりネットワーク上に流通，公開されている構造化文書の検索はますます重要になっている．現在の構造化文書検索は，選択条件および出力文書構造を XML 問合せ言語を用いて宣言的に指定する方法，もしくは Web サーチエンジンにみられる情報検索技術による全文検索がほとんどである．前者は利用者があらかじめ検索対象とする文書の論理構造についての知識を必要とし，後者の検索単位は物理構造上の単位であるファイルに固定されている．そのため利用者が文書の論理構造を意識せずに問合せとの関連性の高い文書部分を取り出すことができない．本論文では，利用者の問合せとの関連性が高く，しかも論理構造上の単位となる文書部分の検索を「文脈検索」と呼び，(1) 論理構造上の単位となる文書部分の特定，(2) 文書内容を用いた利用者の問合せとの関連性の高い文書部分の抽出，を実現し，その有効性を検証する．

A Retrieval Method for Partial XML Documents Using Their Structures and Contents

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

The advent of XML makes retrieving techniques of structured documents on the network more and more important. However, current retrieval methods are the use of query language by specifying selection conditions and output structures or the use of keywords of traditional Information Retrieval methods. For the former methods are required by users to know the document structures beforehand. The latter methods are required to retrieve a whole documents. Therefore users are not able to retrieve partial documents highly related to users' query without considering document structures. In this paper, we propose a new method in order to retrieve appropriate partial XML documents without having the knowledge of documents' structures beforehand. We call this method "Context Search". The process of our context search consists of two steps: (1) identification of partial XML documents which are coherent and meaningful unit; and (2) evaluation of the relevance of the identified partial documents against queries. We describe our developed algorithms to identify result partial documents as an instantiation for context search methods, and we report our evaluation experiment to verify the effectiveness of our method.

1. はじめに

XML（Extensible Markup Language）[21),24),26)]の出現によりネットワーク上に流通，公開されている構造化文書の検索はますます重要になっている．現在の構造化文書検索は，選択条件および出力文書構造を XML 問合せ言語を用いて宣言的に指定する方法と Web サーチエンジンにみられる情報検索技術によるキーワード入力による全文検索に分類される．前者の問合せ言語を用いた検索では，文書の論理構造についての知識と問合せ言語の構文に従った検索式を記述する必要があり，末端利用者が利用するには複雑である．一方，情報検索技術に基づく全文検索は利用者に検索対象とする文書の論理構造についての知識や複雑な問合せ言語を必要とせず，利用者は現在の Web サーチエンジンのように，入力キーワードの論理結合を問合せとするため特別な準備を必要としない．また検索結果がランキングされることで，利用者に検索結果の有用性の 1 つの尺度を提供する．しかし，XML 問合せ言語では，検索結果の出力構造も指定できるのに対し，

† 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)
†† 国立情報学研究所ソフトウェア研究系
Software Research Division, National Institute of Informatics (NII)

検索単位は物理構造上の単位であるファイルに固定されていて，構造に依存した単位を検索結果にすることができない．

全文検索のファイル単位の検索結果には次の問題点がある．第 1 に検索対象文書が大きく，入力キーワードとの関連性が高い部分がそのごく一部である場合，その部分のキーワードの出現頻度が高くてもファイル全体としてスコアが低くなるため，検索結果のランキングが低くなり検索漏れとなる可能性が大きい．第 2 に検索結果はつねにファイル全体を表示するため，その中のどの章や節が関連性が高い部分であるかを識別できない．構造化文書に対しての全文検索では，各文書を走査することで文書中のタグを認識できるため，文書中の章や節などの構造をシステムが解析することが可能である．よって，システムが解析した文書構造を利用することで各構造を単位とした全文検索を行うことができる．我々は構造化文書検索には末端利用者に専門的な問合せを要求しない柔軟な検索単位での検索方法が重要であると考える．我々の研究の目的は，XML 文書の文書構造を利用し，構造から文書中の文脈の境界をみつけることであり，文書中の構造に基づき XML 文書を分割し，今までの文書検索で行われてこなかった，より粒度の小さい部分文書を対象とした検索を行うことで，入力したキーワードに局所的に関連する部分を抽出することである．

本論文では，利用者の問合せとの関連性が高く，しかも論理構造上の単位となる文書部分の検索を「文脈検索」と呼び，(1) 論理構造上の単位となる文書部分の特定，(2) 文書内容を用いた利用者の問合せとの関連性の高い文書部分の抽出，によって実現する手法を構築する☆．我々は文脈検索を行うために，XML 文書の文書構造と内容を用いた部分文書の抽出手法を提案し，提案手法の有効性を検証する．

以下，2 章では，本研究の背景と関連研究について，3 章では本論文で用いるデータモデルについて，4 章では部分文書特定方法について，5 章では，部分文書抽出法とこの手法の有効性を検証した実験結果について述べ，6 章で本研究のまとめと今後の課題について述べる．

2. 背景と関連研究

本章では，我々の提案する文脈検索の背景と関連研究を述べる．

☆ 文脈検索は，一般的な自然言語としての文章が持っている文脈までを意図したものではなく，文書構造を表すタグ名の並びが示す境界を文書構造から得られた文脈として意図したものである．

XML の登場により，構造化文書の利用が急速に拡大し，XML に対応した構造化文書の管理，格納，検索方法が必要となっている．

構造化文書データベースと情報検索システムの統合利用に関する研究は，これまでにもなされてきているが，構造化文書として主に SGML [12)~14)] 文書を対象としてきたため，各文書の構造は文書型定義（DTD）によってあらかじめ定義されているという前提のもとに論議されてきた[10),16),17),20),22)]．しかし，XML 文書では，DTD や現在標準化が進められている XML Schema [23)] によって文書の構造を定義している文書（妥当な XML 文書）だけではなく，定義しない文書（整形式の XML 文書）も多く，XML 文書に応じた検索方法の提案が必要となる．

現在の構造化文書検索は，選択条件および出力文書構造を XML 問合せ言語[2)]を用いて宣言的に指定する方法と Web サーチエンジンにみられる情報検索技術によるキーワード入力による全文検索に分類される．

2.1 XML 問合せ言語における情報検索機能

XML 問合せ言語はこれまでに多数提案されているが，主にデータ指向の XML 文書における問合せ向けであり，文書指向の XML 文書が必要としている情報検索の問合せ向けに必要なキーワードの重み付けや結果の順位付けを表現する機能はほとんどない．さらに，利用者があらかじめ文書構造についての知識を持っていて，検索結果の構造を宣言的に指定することを前提としている．

W3C が提案している XQuery [27)] では，基本仕様にはキーワード検索は想定されていない．ユーザ定義関数で全文検索機能を追加することを想定しているが，まだ仕様策定中である．

構造化文書の検索を情報検索としてとらえた場合 XML 問合せ言語に情報検索機能を追加する必要性があるため，XML 問合せ言語への情報検索機能拡張が提案されている．

XML 問合せ処理とキーワード検索を統合する手法が Florescu らによって提案されている[6)]．この研究では，XML 検索とキーワード検索との統合利用を目的として XML–QL [3)] を拡張し，利用者が文書構造を知らない場合の問合せを想定しているが，取り出す要素型の条件を利用者が指定する必要がある．

XIRQL [7)] は，宣言的 XML 問合せ言語である XQL [18)] を情報検索技術を利用できるよう拡張した言語であり，入力キーワードと問合せに重みを指定することができる．さらに，関連性指向の検索のために，問合せ結果の文脈単位となる要素型をデータベース管

理者があらかじめ定めて，この要素型を単位として索引を作る．XIRQLでは，文脈検索を想定してはいるが，利用者があらかじめ文書構造についての知識を持つことが前提となっている．また，具体的な実装法については言及されていない．

このように問合せ言語を用いた検索では，利用者があらかじめ検索対象とする文書の論理構造についての知識を持ち，問合せ言語の構文に従った検索式を記述する必要があり，末端利用者が利用するには複雑である．我々の提案する手法は，末端利用者にこれら問合せ言語の知識を必要としない点が以上の研究と異なる．

2.2 情報検索技術に基づく検索エンジン

従来の情報検索技術は，主にプレーンテキストを対象としてきたため，内部に表題，著者，抄録，章や節などの構造があってもシステムが自動的にこれらの構造を識別することは困難で，文書構造を利用した情報検索は，あらかじめ同じ文書構造を持つSGML文書などの構造化文書に限定されていた．また，検索結果はつねに文書全体であるという前提があった．しかし，XMLをはじめ構造化文書に対しての情報検索では，各文書を走査することで文書中に記述されている章や節などの構造をシステムが解析することができるため，システムが解析した文書構造を利用して各構造を単位とした情報検索を行うことができる．文書構造を利用した情報検索の研究はXMLの出現により，ますます重要となっている．

XMLを利用した検索エンジンやXML文書を対象とした検索エンジンがいくつか公開され始めている．XSet[28]は主記憶上のデータベースと検索エンジンの組合せでXMLをデータ格納言語として利用している．また，XML文書検索に情報検索の技術を導入し文書の各要素型の特徴量を文書の葉にあるテキストに索引づけをして，上位構造の要素ノードは下位ノードの出現値を積算するBUS（Bottom Up Scheme）[20]を利用した検索システムXRS[19]がある．XYZFind[5]は，利用者の問合せを支援する対話的なシステムで，利用者に問合せ結果のスキーマを示すことで検索結果の絞り込みを支援している．これらXML検索エンジンの試みも利用者が入力する少ない検索用語からより良い検索結果を求める目的は我々と共通する．しかし，いずれもDTDで定義した文書構造を持つXML文書を前提にしているところが我々の研究とは異なる．我々の提案する手法は，整形式の文書も対象とした検索手法である．

先行研究[29]において，我々は情報検索技術を用いXML文書構造を表すいくつかの要素をDTDを利用してシステム管理者が指定し，その要素型によって文書を区切って索引を作成し，それらと利用者が入力した問合せの類似度を計算することで，キーワードとの関連性の高い部分文書を検索結果とする手法を提案した．この手法は，妥当なXML文書集合ですべての文書が同一のDTDに従っている場合に有効である．しかし，整形式のXML文書や文書ごとにDTDが異なる場合は，情報検索単位となる部分文書を特定の要素型で指定することが不可能なため，情報検索単位をシステムが自動選定する方法が必要となる．そこで，本論文では先行研究で対象としなかった整形式のXML文書や多様な妥当なXML文書に対象を拡大し，利用者の入力した問合せに関連する最適な部分文書を検索する手法を提案する．

3. XMLデータモデル

本章では本論文で利用するデータモデルと構造化文書検索モデルについて述べる．本論文ではXPath[25]で用いられているデータモデルと記法を利用し，構造化文書検索モデルとして元文書の論理木構造を保持するモデルを採用する．

3.1 XPathデータモデル

XPathデータモデルでは，XML文書をノードの木として扱う．7種類のノードの中で，ここでは根ノード，要素ノード，属性ノードとテキストノードに絞って論じる☆．XMLでは，文書中に展開可能な外部実体として，他のXMLファイルを取り込むことができる．したがって，1つ以上のXMLファイルからXML表現が作られる．XML文書内のすべてのノードで定義する文書順という順序は，展開可能な実体を展開後のXML表現において各ノードのXML表現の最初の文字が現れる順序を表したものである．我々はこのように実体を展開後のXML表現を検索対象とし，以後これらをXML文書と呼ぶ．その結果，検索対象となる論理構造としてのXML文書は1つ以上の物理構造としてのファイルで構成される．また，我々の想定する検索で指定するキーワードは，XML文書内の文字列値と比較し，各ノードの持つ名前は比較対象としない．次に，XPathデータモデルにおける各ノードにおける文字列値と，直接文字データを値として持つ要素ノードと属性ノードについて述べる．

(1) **文字データ**：各ノードには，文字列値を決定する方法がある．テキストノードの文字列値は，文字

☆ 他の名前空間ノード，処理命令ノードとコメントノードは，本研究においては本質ではないので省略する．

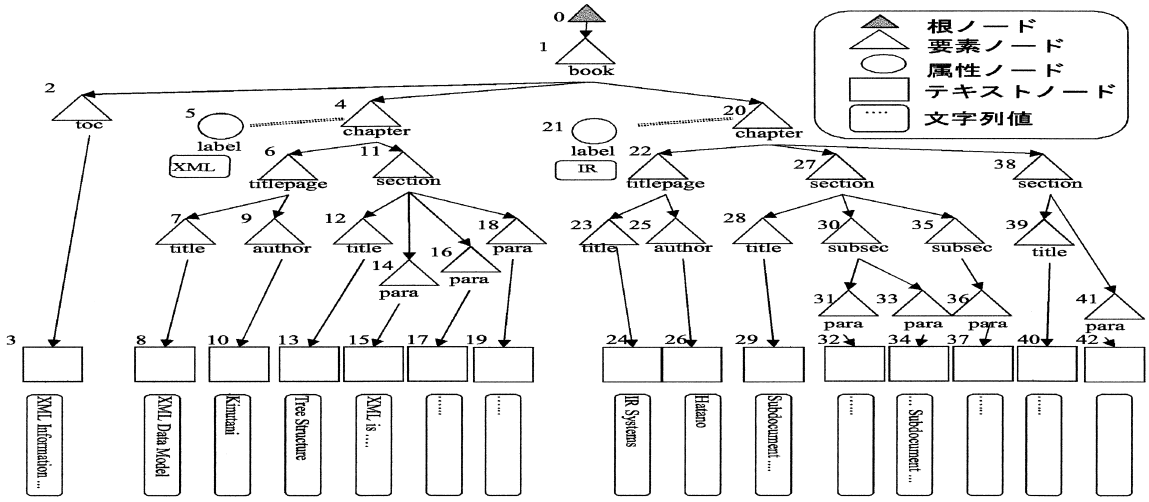

図 1 XML データモデルに基づく XML 文書インスタンスの木構造表現
Fig. 1 A tree structure representation of an XML document instance based on the XML data model.

データである．根ノードは木構造の根であり，文字列値はすべての子孫テキストノードの文字列値を文書順に連結したものである．文書要素型の要素ノードは，根ノードの子である．展開後の XML 表現の要素型はすべて要素ノードを持つ．要素ノードの子として，その要素型内容の要素ノード，テキストノードを持つ．また要素ノードの文字列値は，要素ノードのすべての子孫ノードの文字列値を文書順に連結したものである．要素ノードは，関連する属性ノードの集合を持ち，これら属性ノードの親になるが，属性ノードは，要素ノードの子ではない．さらに要素ノードは，展開された名前を持つ．展開された名前は，名前空間の URI（あるいは null）と局所的な名前を表す文字列からなる．各属性ノードは正規化後の文字列値を持つ．したがって，直接文字データを持つノードは，属性ノードとテキストノードだけである．

(2) **要素ノードと属性ノード**：一般に属性ノードが持つ文字データは，属性名と属性値の組をデータベースで管理可能な値として扱うことが可能である．一方テキストノードが持つ文字データは，自然言語で書かれた文として扱うことができる．
しかし，DTD を設計し，DTD に従った XML 文書を作成する場合は，要素型と属性の扱いに明確な方針を持つが，DTD を持たない整形式の XML 文書では，要素型と属性の扱いは様々であり，使い方に共通認識が得られていない．したがって，本論文では，属性ノードとテキストノードが持つ文字データに差がないものと仮定する．

図 1 は XML 文書インスタンス例であり，**図 2** はその論理構造を図示したものである．図 1 の各ノードの番号は文書順を表している．

3.2 構造化文書検索モデル

構造化文書検索モデルには，重複のないリストモデルと近接ノードモデルがある[1)]．我々の論理構造上の単位となる文書部分の特定のための基本方針は，後者の近接ノードモデル[17)]に近い．我々は，論理木構造を保持した部分文書を文書単位とする．すなわち，各部分文書は，必ず 1 つの最上位ノードとなる要素型を持つ XML 文書とする．したがって，部分文書をその最上位要素ノードの番号 n を用いて指定することができる．ノード番号 n のすべての子孫ノードと属性ノードが表す木構造に対応した文書部分を，ノード $\#n$ の部分文書と呼ぶ．

我々が提案する文脈検索は，(1) 論理構造上の単位となる文書部分の特定と部分文書の作成，(2) 文書内容を用いた利用者の問合せとの関連性の高い文書部分の抽出，で構成される．まず次章では，論理構造上の単位となる文書部分を特定し，検索対象部分文書の作成方法について述べる．

4. 部分文書特定法

XML 文書では，DTD や XML Schema によって文書の構造を定義している文書（妥当な XML 文書）と，定義しない文書（整形式の XML 文書）がある．妥当な XML 文書では，その文書が指定している DTD や XML Schema を見ることで文書構造が分かる．一方，整形式の XML 文書では，各 XML 文書インスタン

```
<book>
 <toc>XML Information Retrieval</toc>

 <chapter label='XML'>
  <titlepage>
   <title>XML Data Model</title>
   <author>Kinutani</author>
  </titlepage>
  <section>
    <title>Tree Structure</title>
    <para> XML is becoming widely used.</para>
    <para>We have developed algorithms. </para>
    <para>A structure is represented as a tree.</para>
  </section>
 </chapter>
```

```
 <chapter label='IR'>
  <titlepage>
    <title>IR Systems</title>
    <author>Hatano</author>
  </titlepage>
  <section>
    <title>Boolean model</title>
    <subsec>
     <para>We propose a heuristic method</para>
     <para>Context search</para>
    </subsec>
    <subsec>
      <para>Result subdocuments</para>
    </subsec>
  </section>
  <section>
    <title>The vector space model</title>
    <para>Exact matching may lead</para>
  </section>
 </chapter>
</book>
```

図 2 XML 文書インスタンス例
Fig. 2 An XML document instance.

スを走査しなければその文書の持つ構造をシステムが理解することができない．したがって，論理構造上の単位となる文書部分を特定するにあたり，妥当な文書はシステムあるいはシステム管理者が DTD や XML Schema を解析し，整形式の文書は個々の文書の構造を個別に解析する必要があるが，システム管理者による解析は負担が大きいため，計算機を利用して自動的に解析される必要がある．

以下の節で XML 文書が特定の DTD に従っている妥当な XML 文書集合と特定の DTD との対応のない XML 文書集合に分けて論理構造上の単位となる文書部分の特定方法について論じる．

4.1 特定の DTD に対応した XML 文書の部分文書特定

ここでは，特定の DTD に従っている XML 文書集合において各文書中から論理構造を考慮した文書部分を部分文書として特定する方法について述べる．この方法は，XIRQL[7]で採用している方法と同様システム管理者の解析を必要とする．本論文では，この方法を「選択ノードアプローチ」と呼ぶ．検索対象とする XML 文書が特定の DTD に従っている妥当な文書の場合は，文書の走査なしに，DTD を見ることによって文書構造を理解できる．したがって，システム管理者が論理構造上の単位となる文書部分を最上位の要素ノード名で指定することができ，それを検索対象部分文書とすることができる．例をあげて部分文書の特定の方法を説明する．

例 1 **図 3** は，“book” を最上位要素型とした XML 文書の DTD である．図 2，図 1 は，この DTD に従った XML 文書インスタンスとその木構造表現である．この DTD から，この文書中に出現する要素型名と要素の出現順序，入れ子関係が分かる．DTD 中で要素型の複数回出現を + , * で指定している展開された要素型名☆は，“section”，“subsec”，“chapter”，“para” である．これらの繰返しは，文書内容の区切りを表している．ここで，“para” は，子ノードがテキストノードだけのため，テキストの区切りではあるが，内容としての区切りと見なすには，粒度が小さすぎることから子ノードがテキストノード以外のノードを含む “section”，“subsec”，“chapter” が，構造上の境界となる要素ノードと考えられる．また，展開された要素型名からシステム管理者は，内容の境界と判断することができる．したがってこの場合，#4，#11，#20，#27，#30，#35，#38，#39 の部分文書を論理構造

```
<!ELEMENT book    (toc, chapter*)>
<!ELEMENT toc     (#PCDATA)>
<!ELEMENT chapter (titlepage, section*)>
<!ATTLIST chapter label #IMPLIED>
<!ELEMENT titlepage (title, author)>
<!ELEMENT section (title, (subsec* | para*))>
<!ELEMENT title   (#PCDATA)>
<!ELEMENT author  (#PCDATA)>
<!ELEMENT subsec  (para*)>
<!ELEMENT para    (#PCDATA)>
```

図 3 文書型定義（DTD）例
Fig. 3 A sample Document Type Definition (DTD).

☆ この XML 文書インスタンスには名前空間の指定がないので，名前空間を表す URI は null となる．

上の単位とすることが DTD から適当と考えられる．さらに各部分のメタ情報に関する部分文書，#2，#6，#22 をシステム管理者が論理構造上の単位として追加する場合もある☆．

以上が特定の DTD に対応した XML 文書の部分文書特定方法である．部分文書の特定はシステム管理者の解析の手法により，対象 XML 文書によって異なる．しかし，文書中に使われている要素型，属性名を手がかりに，前もって決めることができる．

4.2 特定の DTD との対応のない XML 文書の部分文書自動特定

ここでは，DTD のない整形式の XML 文書や，従う DTD が多様な XML 文書集合において，各文書中から論理構造上の単位となる文書部分を特定する方法について述べる．部分文書を特定するための方法としては，(1) すべての XML 文書インスタンスから共通のスキーマを抽出してシステム管理者が部分文書を指定する方法，(2) 各 XML 文書インスタンスから部分文書を自動的に特定するためのアルゴリズムを構築する方法，が考えられる．前者は，整形式の XML 文書インスタンスからスキーマを抽出する研究[8),9)]を適用できる．しかし，各 XML 文書インスタンスごとに抽出したスキーマが異なる場合は，抽出されたスキーマごとにシステム管理者が論理構造上の単位となる文書部分を指定する必要があるため，XML 文書構造の種類が少数の場合に適するが，種類が増加するとシステム管理者の負担が増加する．後者の方法は XML 文書構造の種類が多い場合にも適用できるため，我々は後者を採用する．

我々は，先行研究[15)]において標準語彙を名前空間で指定した要素型に対して検索条件を満たす部分文書を文書構造に基づいて特定する手法を提案した．先行研究では，標準語彙，たとえば Dublin Core の意味での要素型 “title” に検索文字列を含むという検索条件から該当する要素型 “title” が使われている周辺の文脈を知ることを目的とし，構造上の最小文脈単位となる部分文書の根ノードである文脈ノードを文書構造に基づき求めるアルゴリズムを提案した．このとき，各入力キーワードと指定された要素型に対応した要素ノードから文脈ノードを特定していたため，指定した要素型が抽出部分文書中に複数存在しないという前提をおいた．しかし，本研究では要素型を指定しないため，抽出部分文書中に出現する指定した要素型に対応した要素ノードの唯一性の条件を削除し，対象テキストノードあるいは属性ノードの祖先ノードの要素型が同名の兄弟要素ノードを持たないことを条件とした．文字列を直接持つノードの論理木構造上の出現位置から木構造を上にたどり，各祖先ノードが同名の兄弟要素ノードを持たない最大の部分木の根ノードを文脈ノードとし，この部分木に対応する部分文書を最小文脈単位として極小部分文書と定義する．この文脈ノードは，DTD における内容モデル中で特定の要素の出現が複数回指定されている場合 +，* をヒューリティックに置き換えたものである．先行研究[15)]では各入力キーワードと指定された要素型に関連する部分文書を特定することを目的としていたが，本研究では，入力キーワードと照合されるのが XML 文書中の文字列値であることから，あらかじめ抽出候補となる部分文書として文書中の検索対象となる文字列値を持つノードに関する極小部分文書を求めておき，これらの部分文書を対象としてキーワードに関連する部分文書を検索することが目的である．

我々の XML データモデルでは，すべての文字列値はテキストノードか属性ノードが持っている．一方検索対象の XML 文書インスタンスの構造について意識しない利用者の，XML 文書内容に関する問合せは，従来の HTML サーチエンジンと同様，いくつかのキーワードの論理結合であると想定する．利用者の要求は，入力キーワードを内容に持つ文書であり，入力キーワードを構造に持つ，すなわち要素型名や属性名として持つ文書ではない．そのため，入力キーワードと一致する文字列値を直接持つ各テキストノードと属性ノードに対して次に定義する文脈ノードを使って構造上の単位となる部分文書の特定に利用する．

我々が提案する手法で特定される部分文書が完全にシステム管理者が文書中の文脈の境界を各文書ごとに解析する結果と一致するわけではないが，このアルゴリズムによってシステム管理者の処理を軽減することができると考える．

定義 1（文脈ノード） XML 文書 D 中のテキストノードあるいは属性ノードを n とする．D 中の n に関する**文脈ノード** $c(n)$ は，n のある 1 つの祖先ノードであり，次のように定義する：

(1) 属性ノード $n \in D$ の場合，$c(n)$ は n の親にあたる要素ノードである．

(2) n がテキストノードの場合，n の親ノードまたは祖父母（親の親）ノードを $p(n)$ とし，最上

☆ XIRQL では，構造化文書モデルとして重複のないリストモデルを採用しているので “section” に着目すると {#1，#2}，{#4，#5，#6 の部分木 }，{#11 の部分木 }，{#20，#21，#22 の部分木 }，{#27 の部分木 }，{#38 の部分木 } に対応した部分文書に分割している．

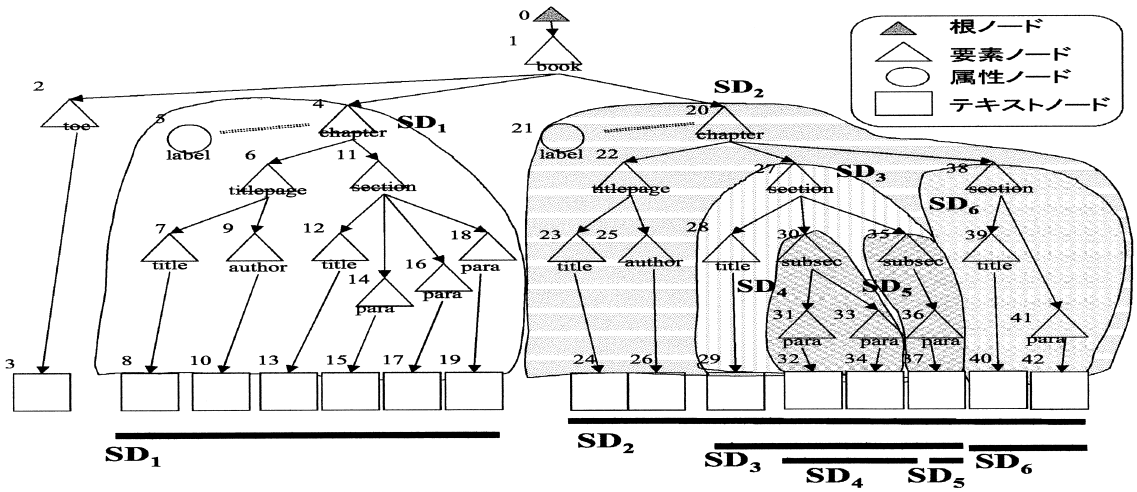

図 4 文脈ノードと対応した部分文書 $SD_i(i = 1, \ldots, 6)$
Fig. 4 Context nodes and partial documents $SD_i(i = 1, \ldots, 6)$.

位の要素ノードを n_d としたとき，$c(n)$ は，経路 $(p(n), n_d)$ 中のノードのうち，次の条件を満足し，しかも $p(n)$ に最も近いノード m である：

m の同じ親を持つ兄弟関係にある要素ノードで，m と同じ展開された要素名を持つものが存在する．ただし，このようなノード m が存在しない場合は，n_d を $c(n)$ とする．

なお，$p(n)$ を n の親ノードとするか祖父母ノードとするかは次の方法によって選択する．

(a) n に兄弟ノードがある場合，すなわち n の親ノードの内容が混在内容である場合は，$p(n)$ は n の親ノードとする．

(b) n に兄弟ノードがない場合は，$p(n)$ は n の祖父母ノードとする．

定義 1 の (2)(b) によって，n に兄弟ノードがない場合は，n の親ノードは文脈ノードとはなりえないことになる．この場合，n の文字列値がそのまま親の要素ノードの文字列値となるが，この親の要素ノードは一般的に文の境界を表す要素である可能性が低く，1 文を構造上の文脈の境界単位としては粒度が小さすぎると考えるためである．

定義 2（極小部分文書） XML 文書 D 中のテキストノードあるいは属性ノード n に対して，定義 1 で定義した文脈ノード $c(n)$ を根とする D 中の部分木に対応した文書を n の**極小部分文書**と呼ぶ☆．

☆ n の**極小部分文書**は他のノードの極小部分文書を含むことも，また他のノードの極小部分文書に含まれることもある．

表 1 ノードと対応する文脈ノード，部分文書
Table 1 Correspondence between nodes and partial document nodes.

テキストノード	文脈ノード	極小部分文書	ノードを含む部分文書
#3	#1		
#8,#10, #13, #15, #17,#19	#4	SD_1	SD_1
#24, #26	#20	SD_2	SD_2
#29	#27	SD_3	SD_2,SD_3
#32,#34	#30	SD_4	SD_2,SD_3,SD_4
#37	#35	SD_5	SD_2,SD_3,SD_5
#40	#38	SD_6	SD_2,SD_6
属性ノード	文脈ノード	極小部分文書	ノードを含む部分文書
#5	#4	SD_1	SD_1
#21	#20	SD_2	SD_2

XML 文書中のすべてのテキストノードあるいは属性ノードに対する極小部分文書の集合を構造の上から文脈の単位として特定された部分文書集合とする．

次に我々の行う文脈検索を定義する．

定義 3（文脈検索） XML 文書集合に対する文脈検索とは，XML 文書集合の各文書インスタンスに対し 4.1 節，4.2 節で述べた方法によって文書構造上文脈の単位として特定された部分文書から利用者の問合せとの関連性の高い部分文書を抽出する検索行動である．

例 2 図 2，図 1 の XML 文書インスタンスを例にして文字列値を直接持つ各ノードに対応する文脈ノードを示す．**図 4**，**表 1** がテキストノード，属性ノードとそれらの文脈ノード，特定された部分文書との関係である．XML 文書インスタンス中に入力キーワード

と一致する文字列が存在する場合，極小部分文書として特定された 6 個の部分文書 $SD_i(i = 1, \ldots, 6)$ が文書構造上抽出される☆．

5. 部分文書抽出法

前章で，文脈検索の (1) 論理構造上の単位となる文書部分の特定法について述べた．本章では，(2) 利用者の問合せとの関連性の高い文書部分の抽出法について述べる．まず，文脈検索における検索モデルについて述べる．

5.1 検索モデル

我々の文脈検索では利用者に負担の少ない現在の HTML サーチエンジンと同様なキーワードの論理結合（AND，OR）による単純な問合せを想定する．

定義 4（単純問合せ） $t_i(i = 1, \ldots n)$ を検索に用いるキーワード，θ を論理演算子（**AND，OR**）とする．単純問合せは次の式で表す，ただし演算子の優先順位は，**AND，OR** とする：

$t_1 \ \theta \ t_2 \ \theta \ \ldots \ \theta \ t_n$

例 3 次の問合せを，利用者が検索システムに入力し，検索システムがこの問合せとの関連性の高い文書部分を検索結果とする．

‘XML’ **AND** ‘$model$’

この問合せの意味は，XML 文書集合からキーワード “XML” と “model” に関する内容を持つ文書部分をみつけることである．

次にこの検索モデルに全文検索に用いられているブーリアンモデルを適用する．

5.2 ブーリアンモデルによる文脈検索

ブーリアンモデルでは，入力キーワードと文書内容を比較し，入力キーワードを含むか否かを判定し，問合せ条件を満たす文書が検索結果となる．したがって，4 章で特定された部分文書を対象として，定義 4 の意味での問合せ条件を満たす文書が検索結果となる．

例 4 図 2，図 1 に示した XML 文書インスタンスを例に，例 3 の問合せを満たす部分文書を考える．キーワード “XML” は，ノード番号#3，#5，#8，#15 で示されているノードの文字列に含まれる．これらのノードに対応した極小部分文書は，表 1 から SD_1 である．一方キーワード “model” は，ノード番号#8，#29，#40 に含まれ，対応した極小部分文書は，SD_1，SD_3，SD_6 である．さらにこれらの部分文書を含む SD_2 も “model” を含む部分文書の抽出候補である．したがって “XML” と “model” を含む部分文書は 2 つのキーワードを同時に含む部分文書 SD_1 となる．つまり例 3 の検索結果として SD_1 の部分文書が特定される．

5.3 ブーリアンモデルによる検索システム

ブーリアンモデルにおける文脈検索の実装は，従来の経路索引，転置ファイルに我々の文脈ノードの値を付加し，ノード間の集合演算を行って検索結果を導くことができる．**表 2**，**表 3** がノードと文脈ノード付き経路索引とノード付き転置ファイルの例である．我々は，これらの索引をデータベースの表として格納し，SQL を使って部分文書を形成するノードを求める実装によって単一のキーワードを入力して，そのキーワードに関連する部分文書を SQL で記述できた．しかし，取り出された部分文書中に出現する入力キーワード数を考慮した検索結果を求めるためには，検索結果を羅列するだけではなく，ランキングのアルゴリズムを作成する必要があるが，情報検索システムにおけるランキングアルゴリズムとの比較検証を必要とする．したがって本手法の有効性の検証には，データベースシステムの SQL を利用した方法での検索システムを適用せず，検索結果に該当キーワード数を考慮した既存の検索システムを利用することにした．

既存の検索システムを利用するにあたり，検索対象となる部分文書を独立した文書と見なし，各文書ごとにあらかじめ入力キーワードとの関連性を計算するために各キーワードの頻度を計算した索引を作成し，利用する方法をとった．この索引を利用することで，入力キーワードの出現回数を考慮した検索結果を求めることができる．

5.4 評価実験のためのプロトタイプシステム

我々の提案する部分文書抽出法の有効性を検証することに目的を絞り評価実験のためのプロトタイプシステムを構築した．このシステムは，XML 文書から部分文書を作成する処理と索引の構築，部分文書検索に分けられる．**図 5** が本システムの概略図である．

5.4.1 DOM 木からの部分文書を作成

XML 文書から部分文書を作成する処理は，次のとおりである：

(1) 表記統一：XML 文書中の不要な空白を除去し文字コードの統一を行う（canonicalize）．

(2) XML 文書解析と DOM 木構築：XML プロセッサ（Apache Xerces version 1.2.2☆☆）を利用して展開可能な実体を展開し，DOM 木を主記

☆ 一般にキーワード検索において元の文書自体を検索結果とする必要性は少ない．必要ならば元文書も抽出単位と考えると 7 個となる．

☆☆ http://www.apache.org/

表 2 文脈ノード付き経路索引
Table 2 Path index with context nodes.

文書 ID	ノード	経路式	リージョン	文脈ノード
1	#1	/book[1]	(0, 757)	#1
1	#2	/book[1]/toc[1]	(7, 42)	#1
1	#3	/book[1]/toc[1]/text()[1]	(12, 37)	#1
1	#4	/book[1]/chapter[1]	(44, 354)	#1
1	#5	/book[1]/chapter[1]/@label	(44, 354)	#4
:	:	:	:	:
1	#40	/book[1]/chapter[2]/section[2]/title[1]/text()[1]	(660, 682)	#38
1	#41	/book[1]/chapter[2]/section[2]/title[1]	(653, 689)	#38
1	#42	/book[1]/chapter[2]/section[2]/para[1]/text()[1]	(697, 720)	#38

表 3 転置ファイルへのノード番号追加
Table 3 Inverted file with context nodes.

キーワード	ノード名	文書 ID	ノード	位置
algorithms	#text	1	#17	255
boolean	#text	1	#29	468
data	#text	1	#8	90
exact	#text	1	#42	698
ir	label	1	#20	396
ir	#text	1	#20	364
model	#text	1	#8	95
model	#text	1	#29	477
model	#text	1	#40	678
xml	#text	1	#3	13
xml	label	1	#5	53
xml	#text	1	#8	86
xml	#text	1	#15	195
:	:	:	:	:

憶上に作る．

(3) DOM 木中の各テキストノードと属性ノードから文脈ノードを計算し，文脈ノードを再上位ノードとする部分文書を作成してディスクに格納する．

5.4.2 索引構築と文脈検索

部分文書をファイル形式で出力し，索引構築と文脈検索は，従来から行われている全文検索の手法を利用する．今回は，フリーソフトである namazu version 2.0.5☆を利用した．索引構築と文脈検索の処理は次のものである：

(1) 索引構築：部分文書すべてを読み込み索引ファイルを作成する．

(2) キーワード検索：問合せキーワードを論理式，正規表現で指定する．検索結果は，namazu のアルゴリズムで高いスコア順にファイル名とファイルの先頭数行が出力される．スコアは TF/IDF で求められる．

☆ http://www.namazu.org/

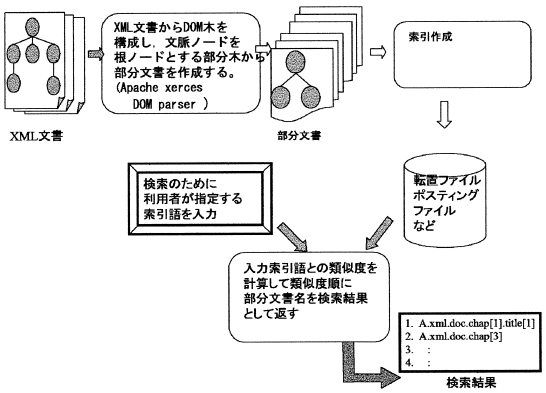

図 5 XML 文書検索システムの概略図
Fig. 5 Our retrieval system for XML documents.

5.5 評価実験

本節では，提案した部分文書分割手法の有用性を評価するために行った実験について述べる．

実験に用いたデータは，我々の所属している研究室で独自に作成したテストコレクションである．いまだ，XML 文書の評価用テストコレクションで公開されたものは存在しないため，W3C の XML に関連する HTML 形式の 17 個の仕様書☆☆を XML 文書に変換したものである．問合せ/解答セットは研究室で作成した次の 3 種類である．

- **問合せ/解答セット 1**
  質問文 XHTML の互換性の問題は将来どう解決されるのか?
  問合せキーワード XHTML compatible issue future direction.
  解答 `xhtml1-20000126.xml` の 5 章および 6 章
- **問合せ/解答セット 2**
  質問文 XML のエンティティの文字コードは UTF-8 のほかに何が利用できるのか?

☆☆ http://www.w3.org/TR/

問合せキーワード XML entity character encoding UTF-8.

解答 REC-xml-19980210.xml の 2.2 節および 4.3.3 項，付録 F 章と，REC-xml-20001006.xml の 2.2 節および 4.3.3 項，付録 F.1.

- **問合せ/解答セット 3**

質問文 XML の要素型名や属性名に使える文字には何があるのか?

問合せキーワード attribute element type name charactercode qualify

解答 REC-xml-names-19990114.xml の 2，3，4 章および付録 A.3 と xml-19980210.xml および xml-20001006.xml の 2.2，2.3，3，3.1 節および付録 B.

本実験では，提案した文脈検索を行う場合の検索対象となる部分文書の違いによる検索結果の比較を目的とし，次の 3 種類の方法によって実験を行った．ただし，2〜3 文（語数 25 未満）の部分文書は，粒度が小さすぎることから検索対象から除外した☆．

(1) 全中間要素ノードアプローチ：全中間要素ノードを根ノードとする部分木に対応した部分文書．すなわち，部分文書が元文書と一致する元文書の最上位要素ノードだけを除外し，その他の要素ノードに対応した部分文書．この方法は，最も効率は悪いが単純な方法である．

(2) 選択ノードアプローチ：4.1 節に述べた選択ノードアプローチで作成した部分文書．XML 文書を定義する DTD がある場合で，管理者による解析の結果選ばれた要素ノードに対応した部分文書であるため，取り出したいと管理者が期待する部分文書を検索対象とすることができる．**図 6** がテストコレクションの構造を定義している DTD であり，この DTD を解析した．構造上の単位となる要素型として “title”，“authlist”，“abstract”，“status”，“div1”，“div2”，“div3” を指定した．

(3) 文脈ノードアプローチ：4.2 節に述べた文脈ノードを根ノードとする部分文書．特定の DTD との対応のない XML 文書に利用する．

5.6 実験結果

我々のテストコレクションから得られた部分文書数は，(1) 全中間要素ノード 4,344，(2) 選択ノード 1,145，(3) 文脈ノード 1,172 である．

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT spec
  (title, authlist, abstract, status, body, back)>
<!ATTLIST spec
    w3c-doctype (cr|note|pr|rec|wd)      #IMPLIED
    other-doctype CDATA                  #IMPLIED
    status (int-review|ext-review|final) #IMPLIED>
<!ELEMENT title (#PCDATA) >
<!ELEMENT authlist (author+) >
<!ELEMENT author (name, affiliation?, email?) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT affiliation (#PCDATA) >
<!ELEMENT email (#PCDATA) >
<!ELEMENT abstract (p+) >
<!ELEMENT status (p+) >
<!ELEMENT body (div1+)>
<!ELEMENT div1 (div2*|p+)>
<!ATTLIST div1 id NMTOKEN #REQUIRED>
<!ELEMENT div2 (div3*|p+)>
<!ATTLIST div2 id NMTOKEN #REQUIRED>
<!ELEMENT div3 (p+)>
<!ATTLIST div3 id NMTOKEN #REQUIRED>
<!ELEMENT back (div1+)>
```

図 6 テストコレクションを定義する DTD
Fig. 6 The DTD of our reference collection.

実験結果の評価にあたり，本実験においては，検索結果として取り出した部分文書が正解部分文書より大きい文書や小さい文書があり，検索結果文書ごとに正解部分文書と適合しているか否かを一意に判定することができない．また全中間要素ノードを対象部分文書とした (1) の場合は，結果の上位を粒度の大きいファイルが占める結果となっている．これは使用した検索ソフト namazu のスコアの計算において，入力キーワードの頻度計算に文書の大きさを考慮していない点が考えられる．

そこで本実験では検索結果のスコアを文書の大きさで調整した値を新たなスコアとして定義し，検索結果をランキングしなおした．さらにテストコレクションの正解部分集合のうち検索システムによって検索された正解部分の割合を表す再現率として，正解を含む（あるいは正解に含まれる）粒度の違う部分文書が出現した場合，該当順位までにそれらの部分文書によって検索された正解部分の割合の合計（最大 1）を利用することにした．一方，システムによって検索された部分文書のうち正解部分文書集合と合致している部分の割合を表す精度には各検索結果文書に含まれる正解の割合を，(検索された適合文書部分)/(検索された適合文書部分 + 検索されなかった適合文書部分 + 検索された非適合文書部分) の式で求めて利用することにした．各検索結果文書が正解が否かを 0，1 では表せないため，正解部分の割合を利用し，本実験では再現率と精度を新たに次のように定義した．

☆ 単語数にはこれといって基準があるわけではない．しかし，文脈の変化を把握するためには 3〜5 文は必要であるといわれている．さらに，除外した部分文書を含むより粒度の大きい部分文書は検索対象となっている．

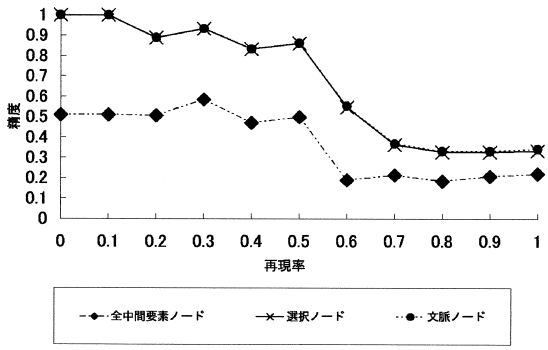

図 7 粒度違いを考慮した再現率–精度グラフ（問合せ 1〜3 平均）
Fig. 7 Recall-precision graph (average of query/answer set 1〜3).

定義 5（再現率，精度） a をテストコレクション中の正解部分文書数，$\{SD_1, \ldots, SD_a\}$ を正解部分文書集合，$\{RD_1, \ldots RD_k\}$ をスコアの高い順に並べた k 個の検索結果文書リストとする．また文書 X の文書の大きさを $|X|$ と表す．スコアが k 番目の検索結果文書 RD_k の再現率 R を，

$$R = \frac{1}{a}\sum_{j=1}^{a}\frac{|(\bigcup_{i=1}^{k} RD_i) \cap SD_j|}{|SD_j|} \qquad (1)$$

とする．この定義式の分子の意味は，1 番目から k 番目までの検索結果文書の正解文書中のリージョンに関する和集合と正解部分文書集合の共通部分の割合を各正解文書を 1 として合計したものである．したがって，分子の最大値は正解文書数 a となる．
スコアが k 番目の検索結果文書 RD_k の精度 P を，

$$P = \frac{1}{k}\sum_{i=1}^{k}\frac{|RD_i \cap (\bigcup_{j=1}^{a} SD_j)|}{|RD_i|} \qquad (2)$$

とする．この定義式の分子の意味は，k 個の検索結果文書に含まれる正解の割合の合計である．

本実験では，この定義によって再現率–精度のグラフを描いた．**図 7** の再現率–精度グラフは問合せ/解答セット 1〜3 の検索結果の平均を，(1)「全中間要素ノードアプローチ」，(2)「選択ノードアプローチ」，(3)「文脈ノードアプローチ」について比較したものである．文書の大きさを調整した後の検索結果の順位から文書の大きさが小さいほど上位に位置付けられ，正解に近い大きさの部分文書が上位を占めることが分かった．また，検索結果図 7 から (1) 全中間要素ノードを対象部分文書とするより，(2)，(3) のようにある程度検索されるべき部分文書を限定した方が検索精度が良いことが分かった．

これらの検索対象部分文書は，文書構造から文脈の単位となる文書であり，検索精度が良いことから文書中の内容に関しても意味ある単位となっていると考えられる．したがって，キーワードの出現密度が他の部分より高い部分を求めることで入力キーワードに関連した部分を取り出すことができたと考える．また (2) と (3) による手法を比較すると検索対象部分文書数において (2) が若干少なかったが検索結果には両者に著しい差が認められなかったことから，我々が提案した文脈ノードを利用した部分文書を対象とする手法によって DTD をみてシステム管理者が検索対象部分文書を指定する手法に近い検索結果を導けることが分かった．

これらの実験結果から，構造化文書から利用者の問合せに最適な部分文書を取り出すために，DTD が利用できる場合は，システム管理者が部分文書を最上位の要素を指定することで，DTD が利用できない多様な XML 文書に対しては，我々が提案する手法を使って求めた極小部分文書を対象として検索する方法が有効であることを示すことができたと思われる．

6. おわりに

本論文では，XML 文書から利用者の問合せに最適な部分文書を文書構造と文書内容の両者を利用して取り出す手法について述べた．従来の文書検索では，検索結果はつねに文書全体であったため，構造化文書の構造を利用した検索は，利用者が構造についての知識をあらかじめ持っている場合に限定されていた．しかし，インターネットの検索エンジンの利用の拡大にみられるように，利用者が入力する少ない入力キーワードでしかも利用者が必要とする最小の部分文書を提供することは，末端利用者が情報の洪水にのみこまれることを防ぐためにも，重要な技術である．

本論文で提案した部分文書抽出手法は，

(1) 利用者の問合せは，従来のインターネットでの検索エンジンの利用法と同じであり，利用者が新たに問合せのための準備を必要としないこと．
(2) 従来の情報検索システムが培ってきた技術を利用できること．
(3) 特定の DTD に対応していない XML 文書であっても，文書構造を走査するだけで検索対象部分文書の作成が可能なこと．
(4) 文書構造中のすべてのノードについて部分文書を作成する必要がないこと．
(5) 元文書のごく一部に利用者の問合せに強く関連した部分がある場合，その部分を検索できること．

以上の利点を有する．しかし，本提案手法は，次の課

題をかかえている．

(1) 部分文書を抽出するために格納する索引あるいは，ファイルが膨大になる危険性がある．格納と処理効率については，部分文書抽出と情報検索の索引作成部分を工夫する必要がある．

(2) 検索対象部分文書の作成方法は，文書の種類によりいろいろな方法が考えられる．今回は，特定の要素型について部分文書を作る方法と文脈ノードを利用して部分文書を作る方法について実験した．しかし，テストコレクションが少ないこともあり，他のテストコレクションによる評価が必要である．さらに部分文書間に重複を許す代わり，元文書の根に近い文書部分については検索対象からはずれる可能性が多い．一般に元文書のメタ情報はこの部分にあるので検索の目的により重複のないリストモデルとの組合せ利用も必要である．

(3) 本手法では，部分文書間に包含関係があるため，重なりあった部分については，検索結果に繰り返し出現する．そのため，どの粒度の部分文書が最適であるかを検討する必要がある．部分文書のランキングについて検討した知見[11]を検索システムに反映させることが急務であると思われる．さらに，システム評価のために新たに定義した再現率，精度の有効性を検証しなければならない．

(4) 我々は，利用者が文書構造を知らずに構造化文書検索を行うには，本論文で提案した文脈検索をきっかけとした対話的な検索行動を支援する枠組みが重要と考える．我々が提案する文脈検索で，利用者の問合せとの関連性の強い文書部分を求めることができる．しかし，利用者に部分文書を検索結果として見せるだけでは，不十分であると考える．取り出された部分文書と元文書の関係も重要である．元文書中に検索結果を表示する場合，関連性の強い部分と関連性のない部分が見分けられる方法も必要と考える．さらに検索結果の絞り込みや再利用を支援するための枠組みの中で，我々の提案する文脈検索で得られた部分文書の格納方法や表示のためのインタフェース，さらに元文書との整合性のとり方などは，今後の課題である．

謝辞 本論文で利用した XML 文書のテストコレクション作成を手伝っていただいた，奈良先端科学技術大学院大学情報科学研究科マルチメディア統合システム講座のスタッフ，学生，そして OB 諸氏に感謝します．

本研究の一部は，文部省科学研究費基盤研究（B）（2）（課題番号：11480088），基盤研究（C）（2）（課題番号：12680417），奨励研究（A）（課題番号：12780309）によるものである．ここに記して誠意を表します．

参 考 文 献

1) Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, pp.61–65, Addison Wesley (1999).
2) Bonifati, A. and Ceri, S.: Comparative Analysis of Five XML Query Languages, *SIGMOD Record*, Vol.29, No.1, pp.68–79 (2000).
3) 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/
4) Deutsch, A., Fernandez, M., Florescu, D., Levy, A. and Suciu, D.: A Query Language for XML, *WWW8 / Computer Networks*, Vol.31, No.11-16, pp.1155–1169 (1999).
5) Egnor, D. and Lord, R.: Structured Information Retrieval using XML, *Proc. ACM SIGIR 2000 Workshop on XML and Information Retrieval* (2000).
6) Florescu, D., Manolescu, I. and Kossmann, D.: Integrating Keyword Search into XML Query Processing, *9th International World Wide Web Conference* (2000).
7) Fuhr, N. and Grossjohann, K: XIRQL: A Query Language for Information Retrieval in XML Documents, *SIGIR'01: Proc. 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp.172–180 (2001).
8) Goldman, R. and Widom, J.: DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases, *VLDB'97, Proc. 23rd International Conference on Very Large Data Bases*, pp.436–445 (1997).
9) Goldman, R. and Widom, J.: Approximate DataGuides, *Proc. Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats* (1999).
10) Grossman, D., Holmes, D., Frieder, O. and Roberts, D.: Integrating Structured Data and Text: A Relational Approach, *American Society of Information Science* (1997).
11) 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 Sys-*

tems (*DASWIS-2001*) (2001).
12) ISO: ISO 8879: 1986. *Information Processing — Text and Office System — Standard Generalized Markup Language* (*SGML*) (1986).
13) JIS X 4151: 1992 文書記述言語 SGML (Standard Generalized Markup Language)，日本規格協会 (1992).
14) 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).
15) Kinutani, H., Yoshikawa, M. and Uemura, S.: Identifying Result Subdocuments of XML Search Conditions, *Proc. 2000 Kyoto International Conference on Digital Libraries: Research and Practice*, pp.232–239 (2000).
16) Myaeng, S.-H., Jang, D.-H., Kim, M.-S. and Zhoo, Z.-C.: A Flexible Model for Retrieval of SGML Documents, *Proc. 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.138–145 (1998).
17) Navarro, G. and Baeza Y., R.A.: Proximal Nodes: A Model to Query Document Databases by Content and Structure, *Information Systems*, Vol.15, No.4, pp.400–435 (1997).
18) Robie, J., Lapp, J. and Schach, D.: XML Query Language (XQL) (1998). http://www.w3.org/TandS/QL/QL98/pp/xql.html
19) Shin, D.: XRS: XML Retrieval System (2000). http://www.dlb2.nlm.nih.gov/~dwshin/xrs.html
20) Shin, D., Chang, H. and Jin, H.: Bus: An Effective Indexing and Retrieval Scheme in Structured Documents, *Proc. Digital Libraries '98*, pp.235–243 (1998).
21) 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).
22) Voltz, M., Aberer, K. and Böhm, K.: Applying a Flexible OODBMS-IRS-Coupling to Structured Document Handling, *Proc. 20th International Conference on Data Engineering*, pp.10–19 (1996).
23) World Wide Web Consortium: XML Schema Part 1: Structures (2001). http://www.w3.org/ TR/xmlschema-1
24) World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (1998). http://www.w3.org/ TR/1998/REC-xml-19980210
25) World Wide Web Consortium: XML Path Language (XPath) Version 1.0 (1999) http://www.w3.org/ TR/xpath
26) World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (Second Edition) (2000). http://www.w3.org/TR/2000/REC-xml-20001006
27) World Wide Web Consortium: XQuery: A Query Language for XML (2001). http://www.w3.org/TR/2001/WD-xquery-20010607
28) Zhao, B. and Joseph, A.: XSet: A High Performance XML Search Engine (2000). http://www.cs.berkeley.edu/~ravenben/xset
29) 波多野賢治，渡邉正裕，吉川正俊，植村俊亮：情報検索技術を用いた部分文書構造の自動抽出，情報処理学会論文誌：データベース，Vol.42, No.SIG8(TOD10), pp.36–46 (2001).

(平成 13 年 6 月 21 日受付)
(平成 13 年 10 月 19 日採録)

（担当編集委員　国島 丈生）

絹谷　弘子（学生会員）

1976 年お茶の水女子大学理学部数学科卒業．1997 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同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，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 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 年同大学院自然科学研究科博士後期課程修了．博士（工学）．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．XML データベース，情報検索に関する研究に従事．ACM 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を経て，1993 年より奈良先端科学技術大学院大学情報科学研究科助教授，現在に至る．1989〜1990 年南カリフォルニア大学客員研究員．1996〜1997 年ウォータルー大学客員准教授．2000 年から国立情報学研究所ソフトウェア研究系客員助教授．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．

植村 俊亮（正会員）

1964 年京都大学工学部電子工学科卒業．1966 年同大学院工学研究科修士課程修了．同年通産省工業技術院電気試験所（現，電子技術総合研究所）入所．1988 年東京農工大学工学部数理情報工学科教授．1993 年奈良先端科学技術大学院大学情報科学研究科教授，現在に至る．工学博士．1970〜1971 年マサチューセッツ工科大学客員研究員．データベースシステム，自然言語処理，プログラム言語の研究に従事．電子情報通信学会，ACM，IEEE 等各会員．