XML処理エンジン Knowledge Factory

XML Processing Engine Knowledge Factory

服部　雅一　HATTORI Masakazu
末田　直道　SUEDA Naomichi

組織内に存在する情報群をXML（eXtensible Markup Language）ベースで管理・蓄積し，オンラインで要求したXMLデータを検索・加工できるXML処理エンジンを開発した。XMLデータをRDB（Relational DataBase）へ格納する手法が数多く提案されてきたが，XMLのデータモデルに適応したXMLネイティブ（特化）なアプローチで格納している。更に，XMLが半構造性であるがために検索コストが増大してしまうが，検索の高速化を図るための最適化手法を開発し，高い性能を達成した。

Toshiba has developed an Extensible Markup Language (XML) processing engine that manages and accumulates XML-based information in an organization. This engine can search and process XML data online. Many techniques for storing XML data in a relational database (RDB) have been proposed. In our engine, XML data are stored using the native database approach adapted to the XML data model. Despite the increase in search cost due to the semi-structured nature of XML, Toshiba has developed an optimization technique for speeding up searches and achieved high performance.

1 まえがき

近年IT（情報技術）の進化により，ばく大な量の情報が入手できるようになった。その一方で必要な情報が大量のデータの中に埋没してしまい，十分に活用できないという弊害も発生している。そこで，特定の個人や部門が保有するノウハウや業務データのうち，企業の経営に重要なものを蓄積して“経営資産”として活用しようとする活動，すなわちナレッジマネジメント（KM：Knowledge Management）が提案されている。

現在行われているナレッジマネジメントは，主としてテキストのキーワードにより情報検索が行われている。しかし，企業内に存在するいろいろな文書データは，構造化されたデータと構造化されていない平テキストデータが混在し，一つの文書を構成している。いわゆる半構造データになっていることが多い。これら半構造データをうまく蓄積・管理することにより，より高度なナレッジマネジメントを実現できると考えられる。

半構造データを表現する方法としてXMLがある。XMLは柔軟な拡張性と連携性を備えた標準のドキュメント記述言語である。

XMLドキュメントをベースにした高度なナレッジマネジメントシステムを構築するためには，XMLデータの効率良い格納と検索の方式が必要になってくる。われわれは，XMLデータの特性を考慮した処理エンジンKnowledge Factory（KF）を開発した。ここでは，KFのXML問合せにおける検索最適化処理方式を中心に，そのベースとなるデータ格納方式，XML問合せ言語について述べる。

2 XMLデータの管理方式

XMLの管理には，いくつかの方式が提案されている（**図1**）。いちばん単純な方式としては，XMLデータをそのままテキストファイルとして管理する方法がある。これは非常に簡便な方法であるが，データ数やサイズが大きくなると格納効率が悪くなったり，XMLの特性を生かした検索が困難である。

二番目の方式として，現在広く利用されているRDBにXMLデータをうまく管理させるというものである。三番目の方式は，構造データを管理するために開発されたOODB（Object Oriented DataBase）を使ってXMLデータを管理させるというものである。

最後の方式は，XMLの持っている半構造データをうまく管理できるXMLデータに特化した，XMLネイティブなデータベース（DB）で管理することである。

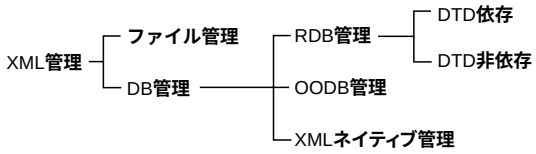

図1．XMLデータの管理方式　半構造であるXMLデータを管理するにはいくつかの方法が提案されている。
Approaches for management of XML data

2.1 RDB管理のアプローチ

RDBでXMLデータを管理する方式が広く提案されている。RDBは表形式を基本としているため，XMLのような木構造のデータを格納するにはいろいろな工夫が必要である。XMLデータをRDBに格納する場合，DTD（Document Type Definition）を前提とするか否かで大別できる。

DTD依存の方法は，XMLデータの論理的なデータ構造が変化すれば，関係スキーマも変更する必要がでてきて，半構造データを扱うにはふさわしくない。また，DTD依存・非依存にかかわらず，問合せにおける階層上のパスが長くなると，それに比例した結合演算が必要になり，検索の負荷が高くなることが知られている。

2.2 OODB管理のアプローチ

階層化した構造データに親和性の高いOODBで管理する方式でも，RDBと同様にスキーマを前提としているので，データ構造の変更に柔軟に対応するのが困難である。

2.3 XMLネイティブ管理のアプローチ

XMLデータは階層化されたデータ構造であるため，データモデルが表形式であるRDBで格納するには無理がある。

XMLデータには次のような特性があるため，XMLのデータモデルに合ったDBを構築する動きがでてきている。

⑴ DTDがないXMLデータも許容する。

⑵ 要素間順序に意味がある。不定数の繰り返しも発生する。

⑶ 部分構造の照合検索が頻繁に行われる。

われわれのアプローチは，基本的にXMLネイティブな管理機構を開発し，高速検索を実現するとともに，XMLデータの加工，再構成も可能にすることを目指している。

3 KFの概要

KFは，次の機能を備えている。

3.1 格納・更新機能

ユーザーやアプリケーションから作成されたXMLデータをXML DBへ格納する機能と，格納されたXMLデータを更新する機能である[1]。特長は次のとおりである。

⑴ KFにおけるDBの論理イメージは，大きな階層構造を持ったXMLデータである。複数の異なった構造の文書を，一つのDBの中で統一的に管理できる。また，部分的なXMLデータに対しては，パスというアクセス手段を提供する。このような特長により，任意の部分構造の単位で挿入，追加，削除，更新が行える。

⑵ XML DB上の任意の部分構造にXMLスキーマを設定することができる。これにより妥当性のチェックが行える。

3.2 検索機能

XML DBから，検索条件を指定して条件に合致するXMLデータを検索する“単純検索”と，加工条件を指定してXMLデータ群から新たな複合コンテンツとしてXMLデータを作成する“複合検索”に分類される。これらの検索条件は，KF-QL（KF Query Language）という検索言語により記述される。特長は次のとおりである。

⑴ 高度加工に優れたKF-QLにより，XMLデータ群の集計・分類では能力が発揮される（**図2**）。

⑵ KF-QL自体はXMLで表現されており，KF-QLで記述された問合せをXMLデータとしてログ化し，XML DBに格納することも可能である。過去の問合せを蓄積，再利用するようなアプリケーションも容易に構築できる。これは従来のKMが“What”情報の共有であったのに対して，“How（分析の方法など）”情報の共有も可能にすることを意味している。

⑶ KFには，問合せをパラメータ化する機能がある。XML DBへの一連の加工プロセスをパラメータ付き問合せで実装しておけば，XML DB上で多次元分析ツールを容易に実現することができる。

```
<kf:query>
 <kf:select>
  <patent title="$t" xAxis="$c1" yAxis="$c2">
 </kf:select>
 <kf:from path="uix://root/特許DB">
  <特許>
   <タイトル>$t</タイトル>
   <キーワード>$k1</キーワード>
   <キーワード>$k2</キーワード>
  </特許>
 </kf:from>
 <kf:from path="uix://root/概念DB">
  <概念 name="機能">
   <概念 name="$c1">
    <kf:star><概念 name="$k1"/></kf:star>
   </概念>
  </概念>
 <kf:from path="uix://root/概念DB">
  <概念 name="技術">
   <概念 name="$c2">
    <kf:star><概念 name="$k2"/></kf:star>
   </概念>
  </概念>
 </kf:from>
</kf:query>
```

図2．問合せの例　「文書と概念情報を組み合わせて分類して，複数文書を多次元テーブル化する」問合せの例である。
Example of query data

4 KFの検索処理方式

4.1 最適化処理の方式

検索処理は，ユーザーの検索条件を記述した問合せを入力とし，検索条件に合致するデータ群を出力とする処理である。一般に**図3**のようなステップを経ることが多い。

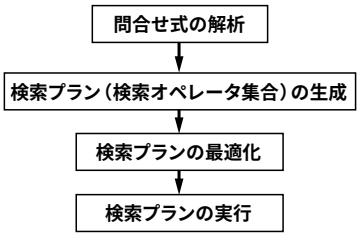

図3．一般的な検索処理　従来の検索処理では，このような4ステップで実現されている。
Search processing in general database management system

検索処理中には，DB上の数多くのデータにアクセスする必要があり，複雑な検索要求の場合，処理時間が長くなる可能性がある。そこで検索プランの最適化処理が必要になる。例えば，ファイルアクセスの最小化や結合演算コストの最小化などの観点が挙げられる。

RDBにおけるSQL（Structured Query Language）の問合せ最適化技術が長年研究・開発されてきたが，XMLにそのまま援用するにはいくつかの問題がある。例えば，以下のようなXMLの特性に対応した最適化処理が必要となる。

⑴　データモデルを階層構造に変更する必要がある。
⑵　階層上の部分構造への照合検索が必要である。
　ⓐ　同一構造を持つ要素の発生階層は一定でない。
　ⓑ　同一構造の入れ子が発生する可能性がある。
　ⓒ　上記の理由から，検索した情報の明確な位置関係がユーザーにとって見えないので，あいまいなパスでの検索が必要である。
⑶　要素順番を指定した検索を考慮しなければならない。
⑷　キーワードなど語彙（ごい）検索が多い。

KFでは，検索グラフに対する最適化ルールのパターンマッチによって，検索プランを生成・実行することを特長とした方式を実装している。

4.2　データファイル

登録・更新された実XMLデータを管理するファイルである。KFではXMLデータを，要素オブジェクト，リストオブジェクト，テキストオブジェクト，シンボルオブジェクトの4種類に分解して，データファイル上に配置して格納する。

格納要求された場合，XMLデータの構文解析を行った後，上記のオブジェクトに分解してデータファイル上に配置する。XML DB上の該当部分にXMLスキーマが設定されていれば，XMLスキーマが対象とする制約範囲を算出し，XMLデータと合成してスキーマチェックを行う。また登録・更新時に，検索のための補助情報として，実データ以外にインデックスファイルを作成・更新している。

4.3　インデックスファイル

高速に検索するために，データファイルとは別に2種類のインデックスファイルを管理している。

⑴　**要素名称生起インデックス**　DB中に格納されているタグ名や属性名をキーとして，DB中に配置されている対応要素オブジェクトのID（IDentifier）を返すインデックスである。
⑵　**データ生起インデックス**　XMLデータ内に発生するテキスト情報をキーとして，DB中に配置されている対応要素オブジェクトのIDを返すインデックスである。テキスト情報の部分文字列をキーワードにした検索も可能にするため，N-gram⑵解析(注1)と形態素解析をミックスした文字列切り出しを行っている。

4.4　検索グラフと最適化ルール

この最適化処理は，複数の制約条件を表す検索グラフに対して，最適化ルールをパターンマッチさせながらコストの少ない制約条件を選択し，その値を具体化させて，その具体化された状況を伝播（でんぱ）させながら検索プランを生成する方式を採っている。

4.4.1　検索グラフ　検索グラフとは，KF-QLで記述された問合せ式を構文解析し，双方向リンクとノードを含むネットワークとして生成したものである（**図4**）。図4の左図は，「指定されたパス（" uix://root "）中に任意（<kf:star>）に出現する" 特許 "情報（<特許>）に対して，下位の" 名称 "情報（$t）が" 検索 "という文字列を含んでいるならば（<kf:cmp ...>），" 名称 "情報を抽出して" 文献 "情報として構成せよ（<kf:select>）」という検索要求を表している。

われわれは，XML DBに対する検索要求を，以下の操作制約モデルに分類して検索グラフを設計した。

⑴　**ノード階層制約**　ノード間の階層制約を記述する。
⑵　**ノード種別制約**　指定ノードの種別を記述する。
⑶　**ノード比較制約**　ノードの比較条件を記述する。
⑷　**ノード順序制約**　コレクションとして自ノードの順序に関する制約を記述する。

```
<kf:query>
  <kf:select>
    <文献>$t</文献>
  </kf:select>
  <kf:from path="uix://root">
    <kf:star>
      <特許>
        <名称>$t</名称>
      </特許>
    </kf:star>
  </kf:from>
  <kf:where>
    <kf:cmp op="like"
            param1="$t"
            param2="検索"/>
  </kf:where>
</kf:query>
```

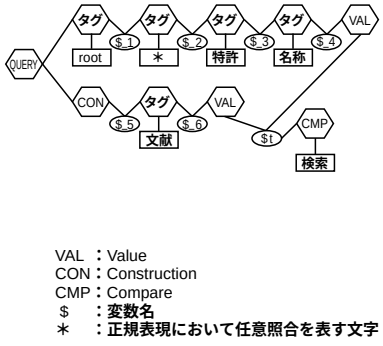

図4．問合せと対応する検索グラフの例　左は，KF-QL（KFが提供する問合せ言語）で書かれた問合せの例，右はそれに対応した検索グラフの例を示す。
Example of query data and corresponding search graph

（注1）　2バイト連続で文字列を切り取る手法。

図4左図を上記操作制約モデルに基づいて表現したものが図4右図の検索グラフである。各6角形で示されたノードをGノードと呼んでいる。

4.4.2　最適化ルール　現時点でKFに実装している最適化ルール集合の一部を**表1**に示す。ここで言う最適化ルールは，ルール番号，Gノードの種類，コスト，適用条件（IF），その時の検索オペレータ（THEN）から構成されており，現時点で20種類程度の最適化ルールが組み込まれている。

4.4.3　検索オペレータ　現時点で9種類の検索オペレータが組み込まれており，実装している検索オペレータ集合の一部を**表2**に示す。DB中のデータをアクセスする様々な手段に対応したものになっている。例えば，“ PathExpand2 ”は，要素名称インデックスを使って，(1)のノード階層制約を高速に制約充足する手段である。

表1．最適化ルールの例
Example of optimization rules

番号	Gノード	コスト	IF	THEN
01	タグ	1.0	rootを持つ	PathInstの適用
02	タグ	0.5	親は具体化	PathExpand1の適用
03	タグ	0.2	要素名称生起インデックスの存在	PathExpand2の適用
⋮	⋮	⋮	⋮	⋮
32	CMP	0.1	データ生起インデックスの存在	Findの適用

表2．検索オペレータの例
Example of search operators

検索オペレータ名	処理概略
PathInst	パス“ root ”を取り出す。
PathExpand1	要素名称をキーにして下位要素オブジェクト集合を展開する。
PathExpand2	要素名称をキーにして上下要素オブジェクト対の集合を生成する（要素名称生起インデックスを使用する）。
⋮	⋮
Find	データ生起インデックスを使って要素オブジェクト集合を生成する。

4.5 検索処理の概要

検索処理の概要を**図5**に示す。この処理は，検索プラン生成と最適化を同時に行う山登り探索的なアプローチであり，局所最適解に陥る可能性はある。

4.6 検索速度に関する実験

他のXML処理エンジンやXML DBと同様の問合せにて性能比較を行ったが（他のエンジンやDBには複合検索の機能がなかったため，単純検索にて比較），5～100倍の性能差が見られた。また，データボリュームに対してほぼ線形の性能特性を示した。

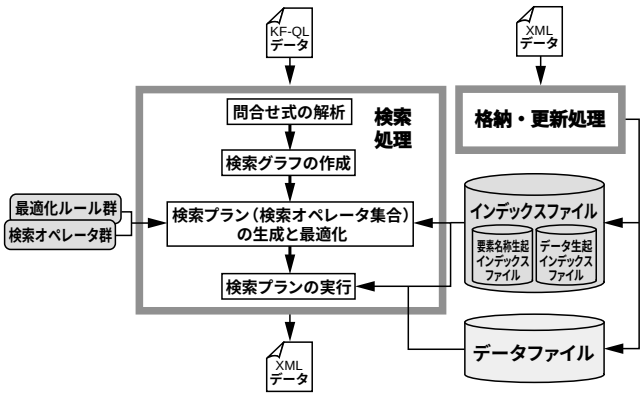

図5．KFにおける検索処理　検索プランの生成と最適化が中心ステップである。インデックスファイルや最適化ルールを駆使して効率の良い検索プランを生成する。
Search processing in Knowledge Factory

5 あとがき

KFは，XMLデータ群からの集計・分類で特に能力が発揮される問合せ言語KF-QLと，いろいろな検索パターンでも安定した高速性能を達成する最適化手法に特長がある。

KFは，組織内に大量に存在する半構造文書データをXMLで管理し，これらを共有，再利用，再構成できるXMLベースのKMシステムの構築を目的に開発された。上記のKFの特長は，KMシステムでは特に意味を持つものであり，各種アプリケーションも効率よく開発されている。

問合せ言語については，W3C（World Wide Web Consortium）において標準化活動が進められており，XQuery（問合せ言語の一種）がワーキングドラフトの段階にきている(2)。KF-QLも基本的にXQueryと類似した言語構造を持つので，この最適化技術もXQuery向けに移植することは可能である。

文　献

(1)　服部 雅一，ほか．ナレッジマネジメント向けXML処理エンジン．東芝レビュー．**56**，5，2001，p.23－25.
(2)　W3C．XQuery: A Query Language for XML．http://www.w3.org/TR/xquery.

服部　雅一　HATTORI Masakazu
研究開発センター 知識メディアラボラトリー研究主務。ナレッジマネジメントシステムの研究・開発に従事。情報処理学会，人工知能学会会員。
Knowledge Media Lab.

末田　直道　SUEDA Naomichi, Ph.D.
研究開発センター 知識メディアラボラトリー研究主幹，工博。人工知能，ナレッジマネジメントの研究・開発に従事。情報処理学会，人工知能学会，情報知識学会，AAAI会員。
Knowledge Media Lab.