XMLによるナレッジナマネージメントのためのテキスト版OLAPとその構築環境

野々村 克彦[†] 服部 雅一[†] 櫻井 茂明[†]
磯部 庄三[†] 末田 直道[††]

近年，我々はITの進歩によって，多量の情報を入手できるようになったが，その反面，本当に必要なデータを得ることが困難になってきている．企業内に存在する重要な情報やノウハウを企業内の知識資産として蓄積，再利用することが重要である．この仕組みをナレッジマネージメント（KM）と呼んでいる．従来のKMはテキストのキーワード検索を基本に行われている．しかし企業内の多くの文書は半構造データである．これらのデータをうまく操作することによって，高度なKMを実現できる可能性がある．XMLは強力な半構造を表現する言語である．XMLによってシステムがある程度意味を理解できれば，いろいろな視点での分析が可能になる．すでに数値情報をベースにしているRDB，DWHの世界ではOLAPが存在している．我々はXMLネーティブなデータベース（KF）を開発して，文書データに対しても同様な高度な分析を行うことのできるテキスト版OLAPの仕組みとその構築環境を提案する．これらは，KFの特徴を生かして，ドリルダウン，スライシングなどのメカニズムを容易に構築できる．実験の結果，我々の構築環境を利用すると従来の方式に対して約3倍の生産性を示した．

Text-based OLAP Using XML for Knowledge Management and the Development Environment

Katsuhiko Nonomura,[†] Masakazu Hattori,[†]
Shigeaki Sakurai,[†] Shozo Isobe[†] and Naomichi Sueda[††]

Recent years, we can get a lot of documents easily owing to the IT progress. On the other hand, the huge amount of information has caused the difficulty of getting the required information. So, it is vital to store and reuse important information and know-how that a specific individual or a specific section has, as knowledge assets of a company. This activity is called Knowledge Management (KM). Many KM systems use keyword search as the text retrieval method. But many documents in a company have semistructured data. By accumulating and managing the semi-structured data well, it is possible to realize more advanced KM system. XML is a powerful language for expressing semistructured data. The system will enable us to analyze the document data from various viewpoints, if it can understand semantics of the documents by XML. There has already been OLAP for numerical data in RDB or DWH. We developed a native database system (KF) for XML, and OLAP that can analyze text documents. We call this "Text-based OLAP". We also constructed a development environment for text-based OLAP applications. It supports to make a drill-down mechanism and a slicing mechanism using the features of KF. The experiments showed that our environment attained the productivity of about three times compared with conventional method.

1. はじめに

近年IT（情報技術）の進化により，莫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ージメント（KM）が提唱されている．

現在行われているKMは主として，テキストのキーワード検索で行われている．しかし，企業内に存在す

† 株式会社東芝研究開発センター知識メディアラボラトリー
Knowledge Media Laboratory, Corporate Research and Development Center, Toshiba Corporation
†† 大分大学工学部知能情報システム工学科
Department of Computer Science and Intelligent Systems, Faculty of Engineering, Oita University

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て1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 KM を実現できると考えられる．

半構造データの表現に適した言語として XML（Extensible Markup Language）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．

文書データを XML で表現し，ある程度計算機がその意味を扱うことができるようになると，きめ細かい検索，いろいろな観点からのデータ分析などが行えるようになると期待される．つまり，従来基幹系で行われていた Relational Database（RDB），Data Ware House（DWH）による OnLine Analytical Processing（OLAP）が，デジタルドキュメントをベースにした情報系でも行えるようになる．我々はこれをテキスト版 OLAP と呼んでいる．

このようなテキスト版 OLAP を実現するためには，XML データの効率の良い格納・検索方式が必要になってくる．我々は XML データの特性にあった Database Management System（DBMS）である Knowledge Factory（KF）を開発した．

本論文の目的は我々が考えるテキスト版 OLAP についての基本的考え方と，それを実現するためのベースとなる XML の DBMS である KF の基本機能を提案するとともに，KF で提供しているクエリ言語 KF-QL の機能を利用したテキスト版 OLAP 構築環境について述べ，その生産性について評価することである．

まず，2 章ではテキスト版 OLAP についての基本的考え方に関して述べ，3 章では KF の概要およびその特徴について述べ，4 章では KF 上に構築したテキスト版 OLAP の構築環境の概要について述べる．5 章ではテキスト版 OLAP 構築環境の重要な機能の 1 つである自動分析機能の実現方式について詳述し，6 章では構築環境の生産性を評価するため，比較実験を行った結果に関して考察する．

2. 研究の位置付け

2.1 XML による KM

組織内に存在する知識には，形式知と暗黙知があるといわれている．形式知とは言葉で表現できる知識を意味しており，文書形式などで存在している．一方，暗黙知は言葉や文章で表現されていないもので，ノウハウ，観点，想いなどがあげられる．KM システムは暗黙知–形式知変換をともなう知識創造のスパイラルプロセスを，うまく管理し支援する仕組みであるといえる[1)]．したがって，狭義の KM（米国流）のように，文書管理を中心に共有・再利用を支援するということにとどまらず，知識創造のための支援も強く求められている．

組織内の情報には様々なものが存在する．代表的なものに基幹系システムで利用されている構造が明確な情報がある．一方，その対極にあるのが，Web 上にあるようなテキスト情報である．しかし，組織内にある情報の多くは，日報情報，議事録，設計情報など，大まかには構造があるが，詳細にはそれぞれ異なる，いわゆる半構造データである．これら半構造データを構造化した枠組みに体系化してしまうと，自由度に欠け利用しにくいものになってしまう．また，これら情報を単なるテキスト情報として扱うと，構造的意味があるにもかかわらず，その情報を有効に利用することができないという問題点がある．XML はまさに半構造データに適したデータ表現であるといえる．以下，XML をベースにした KM のメリットを整理してみる．

- **構造の変化に強い：**情報構造は日々変化するため，情報に対する属性追加，削除が自由に行える．
- **操作できる：**プログラムなどから文書操作が容易になり，高精度な検索，分析・分類などが可能になる．
- **異種データ（モデル）の統合：**複数の情報源を組み合わせた複合文書の作成が容易になる．また，文書/数値/メタデータ付き画像・音声などの異種データの統合も容易になる．
- **情報の交換が容易：**企業内における知識フロー（ナレッジチェーン）環境での標準交換フォーマットとして利用できる．

2.2 テキスト版 OLAP とは

基幹系システムにおいては，膨大なデータを RDB などに蓄えて，生産管理，販売管理，給与管理などいろいろな業務タスクを処理している．しかし今後ますます競争が激化していく中で生き残っていくためには，単に業務効率のみならず，経営意思決定にも，これら膨大なデータも利用できる仕組みを構築することが重要であるという認識が一般的になった．そこで，これら大量に蓄積されたデータを，いろいろな観点で，オンラインで分析できる OLAP が注目されてきた．

通常 OLAP はオンライン性を高めるために事前にサマリ情報を作成し，様々な分析軸で抽出したデータを多次元データベースとして DWH に格納することで実現されている．

OLAPにおいては多次元データベースに格納されている次元属性を自由に切り替えて，その断面を表示したり（スライス），縦横軸を回転させて（ダイス）様々な視点から表示させることができる．また，注目した項目に対して詳細に掘り下げてみること（ドリルダウン）をダイナミックに行うことができる．

我々が目指しているテキスト版OLAPも，基本的には従来のOLAPと同様に，多次元のデータに対しスライシング，ダイシング，ドリルダウンをダイナミックに行うことを可能にする．テキスト版OLAPを実現することにより，社内に多く蓄積されている，文書データからも有用な情報を抽出できる可能性が広がると考えられる．たとえば，日常の業務活動の中で発生する業務報告書，週報，トラブル情報，お客様アンケートなど従来，一過性にしか利用されていなかった情報も，分析データとして利用できることになる．

しかし，データベースに格納されている情報はXMLで表現されており，RDBのデータモデルである表構造モデルと異なり木構造である．そして，そこに格納されている多くの情報はテキスト情報であるため，以下のような機能が必要となる．

(1) **高度でかつ高速な検索・分析機能：**RDBと同様に高度な検索・再構成を高速に行う機能．
(2) **分析可能なデータへの変換機能：**テキスト情報から，なんらかの計数処理ができるデータを抽出する．また意味あるタグ情報として新たに生成する機能．
(3) **自動分析機能：**スライシング，ドリルダウンなどの分析プロセスをユーザインタフェースでサポートする機能．
(4) **過去の分析方法の再利用機能：**過去の分析事例を修正・再利用できる機能．

テキスト版OLAPアプリケーションを開発する方法として，XML文書データをRDBに格納し，RDBMSで提供しているOLAP開発環境で構築する方法，既存のXMLデータベース（XQL/XPathサポート）を利用し，DOM APIやXSLTなどを利用して構築する方法，XMLの特徴を考慮に入れたOLAP開発環境（テキスト版OLAP開発環境）を利用する方法が考えられる．我々は開発の生産性，システムの柔軟性という観点でテキスト版OLAP開発環境を提供する方式をとった．これらの方式の比較は6章で述べる．

我々は上記のテキスト版OLAPを構築するために必要な機能のうち，(1)と(4)はKFの機能で実現させた．(2)と(3)はKF上にテキスト版OLAP構築環境として実現している．

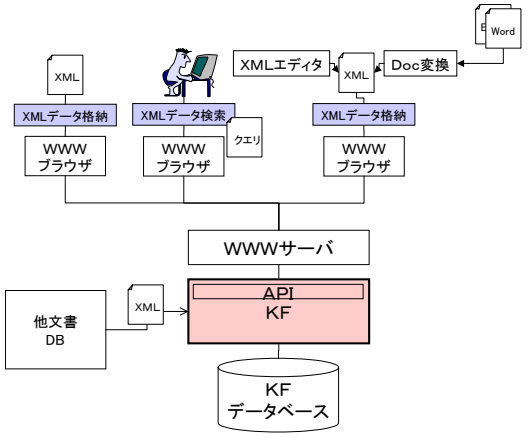

図1 KFの機能構成
Fig. 1 Configuration of knowledge factory.

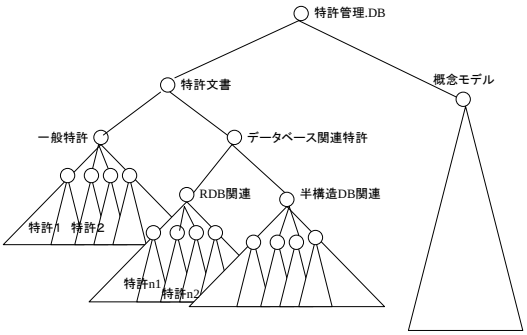

図2 KFデータベースの論理的格納構造
Fig. 2 Logical image of KF database.

本論文では，特に「(3)自動分析機能」について5章で，その実現方法を詳述し，他の機能は概要にとどめることにする．

3. KFの概要

KFは**図1**のとおりに実装されている．現状の実行環境はWWWサーバのバックエンドで動作している．KFの機能は格納機能と検索機能とに大きく分類される．

3.1 格納機能

ユーザやアプリケーションから作成されたXMLデータをXMLデータベースへ格納する機能である．

特徴1 複数の異なった構造を統一管理

KFにおけるデータベースの論理イメージは，大きな階層構造を持ったXMLデータである．**図2**は特許情報などを格納したときの論理的格納構造である．このように複数の文書が1つのデータベース（特許管理．DB）に格納されている．こ

表 1 格納機能コマンド
Table 1 Command list for storing XML data.

機能	コマンド	パラメータ	概要
追加	setSchema	path, schemaData	指定されたパス（path）の下に XML データが従うべきスキーマデータ（schemaData）を指定する
	regist	path, xmlData	指定されたパス（path）の下に XML データ（xmlData）を挿入する
更新	update	path, xmlData	指定されたパス（path）以下を path も含めて XML データ（xmlData）と置き換える
削除	remove	path	指定されたパス（path）以下を path も含めて削除する
取得	getXML	path	指定されたパス（path）以下の XML データを取得する
	getSchema	path	指定されたパス（path）の下に XML データが従うべきスキーマデータを取得する

こでは，各特許情報のコンテンツの格納位置が階層構造上，どこの位置にきてもよいことを示している．

また，特許情報のほかにも例のように「概念モデル」情報といった異なるスキーマ（構造）のデータも同一データベース内に混在して格納でき，異なったスキーマをまたがった検索，再構成をシームレスに行うことができる．

特徴 2 きめ細かい部分木の操作

部分的な XML データに対してアクセスするにはパスというアクセス手段を提供するとともに任意の部分木単位に挿入，削除，更新が行える．

また，XML データベース上の任意の部分木に XML スキーマを設定することができ，これにより妥当性のチェックが自動的に行える．現在は XDR（XML Data Reduced）というスキーマ言語をベースにしている．

KF における格納機能における操作コマンドは**表 1** のとおりである．

3.2 検索機能

特徴 3 高度な加工に優れたクエリ言語 KF-QL

W3C において，XML クエリ言語の標準化活動が実施されている．1998 年 11 月に開催された「QL'98 The Query Language Workshop」[2]において，約 70 種類のクエリ言語が発表された．その後，標準化を進めるべく 1999 年 9 月に「XML Query Working Group」[3]が結成された．代表的な問合せ言語は XQL [4]，XML-QL [5]などがある．上記，XQL，XML-QL を統合する形で Quilt [6]があり，現在この Quilt をベースにして XQuery が Working Draft 段階にある．

我々は KF-QL を提案している．機能的には XQuery のサブセット☆ +独自仕様である．

KF-QL は SQL と同様に SELECT 節，FROM 節，WHERE 節からなっており，以下の機能を有する．

(1) 単純検索：FROM 節で XML データベース上のタグの値，属性値などを変数でバインドし，WHERE 節に変数を用いた比較文を記述することで，単純検索を行うことができる．なお，SELECT 節でクエリ結果を XML 形式で定義できる．

(2) 複合検索：FROM 節を複数利用することで，複数コンテンツを結合して新たな XML データを生成することができる．

(3) まとめあげと集約関数呼び出し：groupBy 節に変数を並べることで，その変数に対応するタグの値別に纏め上げることができる．さらに集計，和，平均などの集約関数を用いることで，たとえば年別，カテゴリ別の集計などを求めることができる．

(4) クエリの入れ子：クエリの中にサブクエリを記述することができる．これにより，クエリの連鎖や複雑なまとめあげを表現できる．

(5) クエリの変数化：条件付けに用いる値のパラメータ化が可能である．このことにより 1 つのクエリで様々な条件付け，クエリ処理を行うことができる．

特徴 4 検索グラフを用いた高速検索機構

クエリの最適化は RDB における SQL のクエリ最適化技術が長年研究・開発がなされている．RDB が今日，多くの分野で利用されているのも，この最適化技術の進展があったればこそといっても過言ではない．XML のような半構造データでは，RDB の方式をそのまま利用することが難しく，以下のような研究がなされている．

XML ネーティブなデータベースとして我々の研

☆ XQuery における条件表現（IF-THEN-ELSE）や Function 機能は含んでいない．

究と近いアプローチとしてはスタンフォード大学のLore[7]がある．

Loreにおける最適化は，まず与えられたクエリから論理クエリプランをSet，Glue，Chainなどの論理オペレータを用い生成し，それをトップダウン，ボトムアック，ハイブリッドの戦略を用いて物理クエリプラン（Scan，Lindex，Vindexなどのオペレータ）を生成する．これら生成されたプランをコストモデルを利用して評価し最適プランを選択する方式を提案している[8]．

我々のアプローチもクエリをグラフに展開（Loreにおける論理クエリプラン）する．Loreは上記3つの戦略をベースに物理クエリプランを作成するが，KFでは，このグラフに対してコストを考慮したヒューリスティックルールを用い，展開しやすいノードをグラフパターンマッチによって求める方式を採っている．つまり，クエリを制約グラフととらえ制約充足問題に定式化した方式を提案している．本方式によりクエリを単純にトップダウン，ボトムアップに探索するのに比べ，クエリの種類によっては1000倍近い高速化を実現できている．この方式の概略は以下のとおりであるが，詳細に関しては別稿に譲りたい．

- KF-QLのクエリを検索グラフに展開する．
- 最適化のためのヒューリスティックルールとして，各ノード項目の状態（データが展開されている否か）によって，データ展開のためのコストが定められている．
- 上記ルールを検索グラフにパターンマッチさせ，最もコストの安いノードに対してデータを展開する．
- このように検索グラフのノードの状態を更新し，再びパターンマッチさせるというように，いわゆるプロダクションルールの「認識–行動サイクル」を実行して検索パスのプランニングを行う．

特徴5 KF-QLのXML化

KF-QLはXMLで表現されており，KF-QLで記述されたクエリ自体をログとしてXMLデータベースに格納することも可能である．過去のクエリを蓄積，再利用するようなアプリケーションを容易に構築することができる．これは，従来のKMが“What”の情報の共有であったのに対して，“How（分析の方法など）”の情報共有も可能にすることを意味している．KF-QLにはHTMLにおけるリンクと同じように外部クエリへのリンク機能がある．XMLデータベースを分析したい要求があるとき，XMLデータの一連の加工プロセスをクエリで実装しておけば，XMLデータベース上で多次元分析ツールが構築できる．

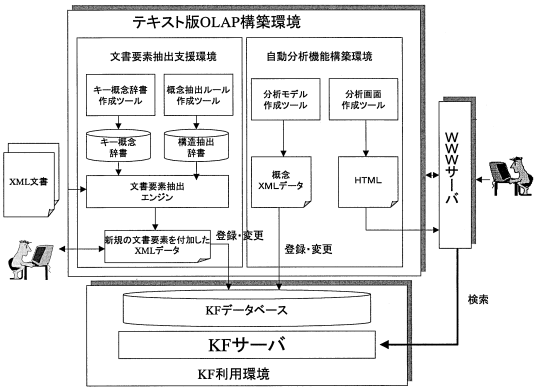

図3 テキスト版OLAP構築環境

Fig. 3 Development environment for text-based OLAP applications.

4. テキスト版OLAP構築環境

4.1 構築環境の概要

テキスト版OLAPを構築するためには，2.2節で述べた(1)〜(4)の機能が必要であると考えられる．

(1)の「高度でかつ高速な検索・分析機能」に関しては前節で述べたKFが有している[特徴3]と[特徴4]でほぼ実現可能である☆．また(4)「過去の分析方法の再利用機能」に関しては[特徴5]で実現できる．

そこで，残り(2)「分析可能なデータへの変換機能」と(3)「自動分析機能」を容易に構築できるように**図3**で示す構築環境を提供している．

これは，図1で示すKFの利用環境の上に，「文書要素抽出支援環境」と「自動分析機能構築環境」を構築している．

以下，文書要素抽出機能およびその構築環境の概略を述べ，自動分析機能に関して5章で詳述する．

4.2 文書要素抽出機能

すでに存在する文書に対して分析精度を上げるために，目的に合ったタグや文書要素を生成したいという要求は多くある．ほとんどの場合，人手でその文書を解析し新たなタグを付加して文書要素を格納しているのが現状である．文書要素抽出支援環境は，その作業をある程度自動化するためのものである．

☆ 基本的な分析に関してはKF-QLの検索再構成機能で行える．高度な分析となると(3)の機能が必要となる．

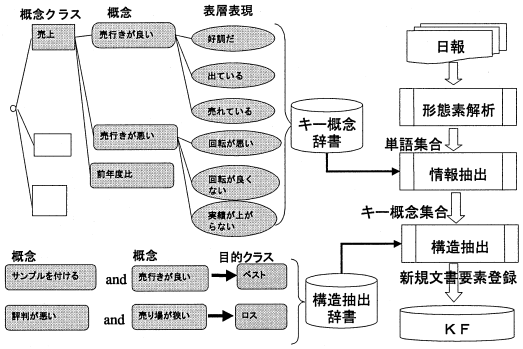

図 4 文書要素抽出の概要
Fig. 4 Outline of tagging process.

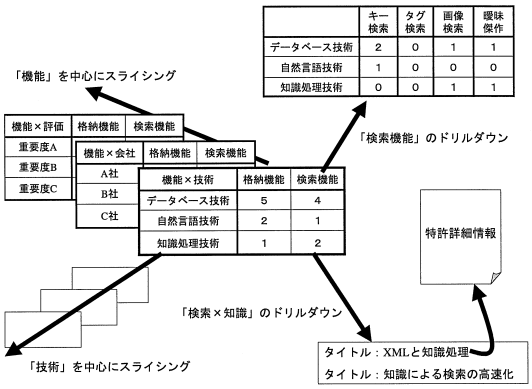

図 5 多次元分析の画面遷移
Fig. 5 Multidimensional analysis image.

つまり XML 文書情報の，あるタグの内容がテキストであった場合，このテキスト情報から分析可能な意味のある情報（分析したい意図に沿った情報）を新たな文書要素として抽出し，元の XML 文書情報に付加する．これによりテキスト版 OLAP における分析対象項目が増え，より高度な分析が可能となる．

たとえば,「営業日報」において営業活動の成功事例（ベストプラクティス）と失敗事例（チャンスロス）に分類したい場合,「自由記述」の要素タグに活動報告が自然言語で記述されていたとする．この自然言語を解析して，その日報が “ベスト” か “ロス” かのクラス分類を抽出し,「活動結果」タグの要素として抽出結果を埋め込むことにより，有効な分析可能データの 1 つになる．

4.2.1 文書要素抽出支援環境の概要

文書要素抽出を行うために「キー概念辞書」と「構造抽出辞書」の 2 つの辞書を利用する．本支援環境は，これらの辞書の作成支援と文書要素抽出エンジンからなっている．

(1) キー概念辞書作成ツール
キー概念辞書とは，**図 4** の左上のツリーが示すように,「概念クラス」,「概念」,「表層表現」の 3 階層からなっている．ユーザは，本ツールを木構造作成エディタとして上位の 2 階層（概念クラス，概念）を入力する．
また，3 階層目の表層表現に関しては，サンプルデータ（たとえば「営業日報」など）を形態素解析して，その結果を画面上に表示する．ユーザはその切り出された単語表現をカット&ペーストでツリー上に貼り付けていくことにより容易に辞書を作成することができる．
ツールは辞書に格納するとき，言い回しの汎用性を高めるために正規表現にして格納する．

(2) 構造抽出ルール作成ツール
構造抽出ルールとは，図 4 の左下にあるように「概念クラス C_i の概念 A と概念クラス C_j の概念 B が同時に対象文章範囲内に現れたら，この文章は目的クラス P を言っているであろう」とするルールである．ツールは，キー概念辞書を用いて，ユーザがカット&ペーストでルールを作成することを支援する．

4.2.2 文書要素抽出エンジン

図 4 の右側のフローは文書要素抽出エンジンにおける文書要素抽出処理の流れの概略を示している．

(1) 形態素解析部：入力された文書を単語ごとに分割して品詞付けを行う．
(2) 情報抽出部：キー概念辞書を用いて，キー概念集合を抽出する．
(3) 構造抽出部：抽出されたキー概念のうち，異なる概念クラスに属する概念を組み合わせて，目的クラス（“ベスト”，“ロス” など）を構造抽出ルールにより抽出する．

以上のように抽出された情報を新しい文書要素として営業日報に付加して，KF に格納する．

5. 自動分析機能

OLAP は多次元データに対して，ユーザの視点に従ってダイナミックにスライシング，ダイシング，ドリルダウンが行える必要がある．**図 5** に特許情報管理システムにおける，そのイメージを示している．これは特許の要素技術，機能，出願会社，出願日などのいろいろな観点で特許マップを作成し特許戦略立案のための分析支援をするものである．KF への格納データの構造を**図 6** に示す．

このような画面遷移が，簡単に構築できる環境が必

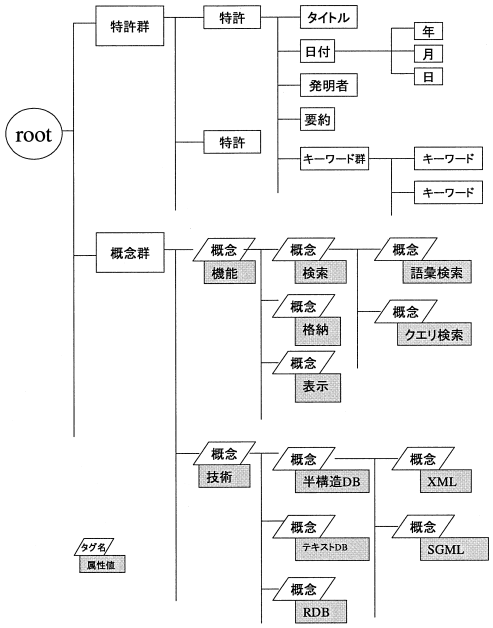

図 6 特許情報管理システムの特許文書と分析モデル
Fig. 6 Patent data and analysis model for patent analysis application.

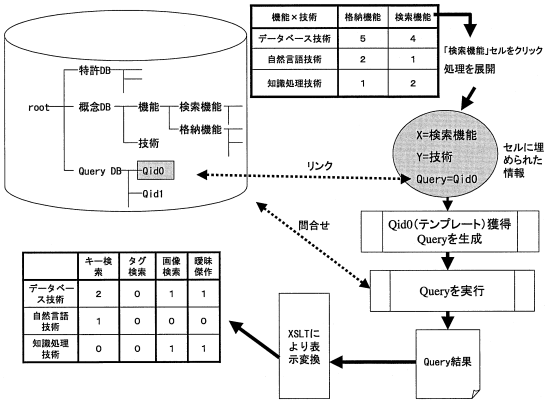

図 7 画面展開の機構
Fig. 7 Drill-down mechanism.

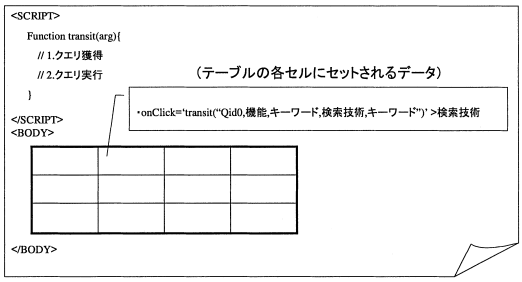

図 8 HTML の例
Fig. 8 Example of HTML.

要である．我々は，KF-QL の特徴である,「クエリの変数化」と「外部クエリのリンク機能」を利用してスタイルシート（XSLT）に，その変換ルールを組み込むことにより，これを実現している．

5.1 分析モデル

従来，OLAP を構築する際，最もコストがかかるといわれているのが，この分析モデルの構築である．この分析モデルに基づいて，多次元データベースを構築するため，いったん構築すると，変更することが容易ではなく，硬直化してしまうという欠点も指摘されている．

分析モデル表現は，基本的に木構造で表現されているため，XML で管理しやすい性質を有している．この分析モデルと文書のタグを関係付けることにより，スライシング，ドリルダウンが可能になる．つまり図 6 の特許管理システムでの分析モデルは概念群☆に記述されている．いろいろな分析軸（概念群における「機能」軸や「技術」軸）の構造がブレークダウンされており，特許情報のタグ（たとえば「キーワード」タグ）と対応付けることにより OLAP が可能となる．

このような形で格納・管理されているため，分析モデルの追加，変更も容易に実行でき，硬直化した OLAP になることを防ぐことができる．

☆ 4.2 節のキー概念辞書とは別のものである．

自動分析構築支援環境は，分析モデル作成ツールとして，対象文書（例では特許情報）の対象タグに発生する単語をリスト化し，それをカット&ペーストできるエディタをサポートしている．この編集結果を XML データに変換し，KF に登録することができる．また，部分木の追加・変更・削除もエディタ上から行うことができる．

5.2 分析ノウハウの利用

5.3 クエリの自動生成

図 7 にドリルダウンやスライシングを行う機構を示す．

(1) 表形式の表示において各セルには，そのセルがクリックされたときのアクション情報が格納されている（後述の XSLT で，その情報は HTML として埋め込まれる）．たとえば図 7 の右上の表の「検索機能」のセルがクリックされたときは，Y 軸はそのまま（「技術」軸）にして,「検索機能」をドリルダウンして，クエリのテンプレートとして，“Qid0” を用いるという情報がセットされている．

(2) **図 8** に示した HTML 内の “onClick” のスク

```
<?xml version="1.0" encoding="Shift_Jis" ?>
<newTag>
  <head>
    <grpX>$x</grpX>
    <grpY>$y</grpY>
    <tagX>$tag1</tagX>
    <tagY>$tag2</tagY>
    <QID>Qid0</QID>
  </head>
  <kf:query xmlns:kf="" vars="$x $tag1 $y $tag2">
    <kf:select>
      <result>
        <X>$sub1</X>
        <Y>$sub2</Y>
        <件数>$cnt()</件数>
      </result>
    </kf:select>
    <kf:from path="uix://root/特許群">
      <特許>
        <$tag1>$val1</$tag1>
        <$tag2>$val2</$tag2>
      </特許>
    </kf:from>
    <kf:from path="uix://root">
      <概念群><kf:star>
          <概念 name="$x">
            <概念 name="$sub1"><kf:star>
                <概念 name="$val1" />
            </kf:star></概念>
          </概念>
      </kf:star></概念群>
    </kf:from>
    <kf:from path="uix://root">
      <概念群><kf:star>
          <概念 name="$y">
            <概念 name="$sub2"><kf:star>
                <概念 name="$val2" />
            </kf:star></概念>
          </概念>
      </kf:star></概念群>
    </kf:from>
    <kf:groupBy vars="$sub1 $sub2 $x $y" />
  </kf:query>
</newTag>
```

(1)結果の再構成
・検索結果をまとめる

(2)検索対象の指定
・変数$tag1、$tag2で与えられた要素値を$Val1、$Val2とする
・タグ名を変数化により汎用クエリーを実現

(3)X軸の探索範囲指定
・変数$xで与えられた概念名の、子供の概念($sub1)の下のどこかに$val1と一致するものを探す
・<kf:star>は任意のどこかを指定する。

(4)X軸の探索範囲指定

(5)変数の纏め上げ

図 9 クエリ・テンプレート
Fig. 9 Template of query data.

リプトが実行される．引数より，クエリテンプレート “Qid0” をリンクより獲得し，X，Y 軸の項目を引数より得て，クエリを生成して，実行する．実行する関数は以下の引数をセットする．

transit(“Qid,X-axis,tag1,Y-axis,tag2”)

Qid：クエリテンプレート名
X-axis：X 軸の概念
tag1：X 軸の文書における分析対象タグ名
Y-axis：Y 軸の概念
tag2：Y 軸の文書における分析対象タグ名

(3) クエリのテンプレートを**図 9** に示す．図 9 (2) で，検索対象領域を変数（$tag1,$tag2）で指定している．これは，KF-QL の特徴であるクエリの変数化によって，汎用的なクエリを実現している．これにより，軸を変えたクエリが可能になる．

(4) クエリの実行結果を得る．“Qid0” は XSLT と対応付けられている．**図 10** にその処理の概要を示す．クエリ結果を読み取り，XSLT に従って

```
<xsl:stylesheet>
  <xsl:template match="QID">
    <!- 変数QIDの設定 -->
  </xsl:template>
  <xsl:template match="grpX">
    <!- 変数grpXの設定 -->
  </xsl:template>
  <xsl:template match="newTag">
    <xsl:for-each order-by="X" select="result">
      <!- X軸サブカテゴリー覧配列aryX作成 -->
    </xsl:for-each>
    <xsl:for-each order-by="Y" select="result">
      <!- Y軸サブカテゴリー覧配列aryY作成 -->
      <!- 件数配列aryCnt作成 -->
      <xsl:eval>genHtml(this)</xsl:eval>
    </xsl:for-each>
  </xsl:template>

<xsl:script><![CDATA[
Function genHtml(this){ // HTML生成
  Txt+="<TH><DIV onClick='transit(¥""+QID+"","+grpX+"","+tagX+","
                                +aryY[I]+ ","+tagY+ "¥") >"+aryY[I]+"</TH>"
}
]]></xsl:script>
</xsl:stylesheet>
```

問合せ結果

```
<?xml version="1.0" encoding="Shift_Jis"?>
<?xml-stylesheet type="text/xsl" href="map.xsl"?>
<head>
  <grpX>機能</grpX>
  <grpY>技術</grpY>
  <tagX>キーワード群/キーワード</tagX>
  <tagY>キーワード群/キーワード</tagY>
  <QID>Qid0</QID>
</head>
<newTag>
  <result>
    <X>視聴機能</X>
    <Y>データベース技術</Y>
    <件数>2</件数>
  </result>
  <result>
    ⋮
  </result>
    ⋮
</newTag>
```

図 10 XSLT の例
Fig. 10 Example of XSLT.

図 8 のような HTML を生成して表示する．作成するテーブルのセルに対応した値が for ループによって作成している様子を示している．“genHtml” が HTML を生成する関数である．

5.4 自動分析機能構築支援環境

本支援環境は，前節で述べた自動分析の機構や，XML，スクリプトの知識がなくても，簡単にスライシングやドリルダウンの画面を作成することを支援する．構築作業は，以下のとおりである．

(1) 分析軸の選定
すでに登録した分析モデルの各概念のルートがリスト表示され，必要な概念を選択する．

(2) 可能な分析軸 (X,Y) の組合せの設定
各概念間での組合せを指定する（デフォルトはすべての組合せ）．つまり，意味のない組合せを排除する．

(3) HTML の生成
ツールはクエリ・テンプレートと XSLT テンプレートを利用し，ユーザが指定した上記の情報をベースに，図 8 のような分析画面トップページの HTML を生成する．

6. 構築環境の考察

我々は，以上述べたように，XML ネーティブデータベースである KF と，その問合せ言語である KF-QL を開発し，それをベースにテキスト版 OLAP 構築環境を構築した．

6.1 従来方式との比較

従来，こうしたテキスト版 OLAP アプリケーションを構築しようとした場合，2.2 節で述べたように我々の方式に加え，以下のような方式が考えられる．

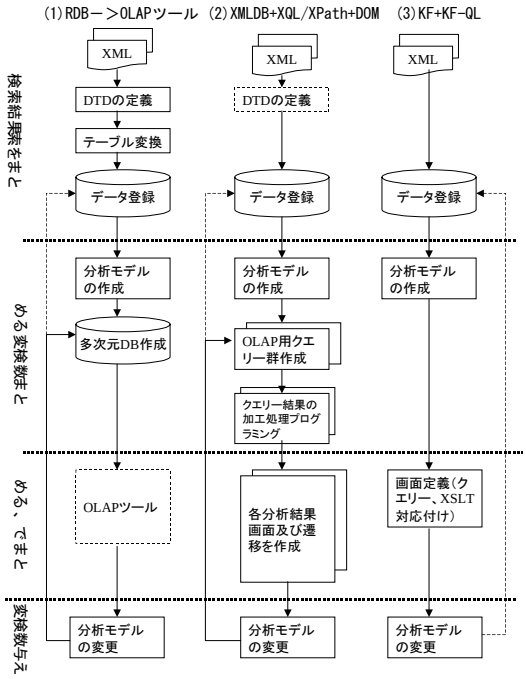

図 11 開発プロセスの比較
Fig. 11 Comparison of development process.

- RDBMS から OLAP ツールを使用する方式
- XMLDB☆+XQL/XPath+DOM を使用する方式

図 11 に上記に方式と KF をベースとしたテキスト版 OLAP 開発環境の開発プロセスの流れを図示した．

(1) データベース作成フェーズ
- XML データを RDB に格納する方法は DTD（Document Type Definition）を利用する方法[9]や，DTD を利用しない方法[10)~12)]などいろいろ提案されている．DTD を用いない方法は，大量のデータになった場合，応答性に問題があり，現状の技術レベルでは現実的ではないと考える．したがって，DTD でスキーマ定義する方式で行うとすると，将来の拡張も考慮に入れたスキーマ設計には多大なコストを要することになる．
- XDBMS の場合，DTD を前提にするツールと前提にしないツールが存在する．
- KF の場合は基本的には DTD を前提にしていないため，XML データ格納は容易に行うことができる．

(2) 分析モデル作成フェーズ
- 分析モデルを作成する作業は基本的に 3 方式とも同じであると考える．RDB の場合，分析モデルに基づいて多次元データベースを作成することになる．したがって，多次元データベースはこの分析モデルに従って作成されるため，この作業には多くのノウハウが必要になる（OLAP で最も重要でコストがかかる作業であるといわれている）．
- XDBMS の場合も，以下のフェーズで述べるように分析モデルに従った機構にある程度依存して作り込む必要があるため，ここでの作業は RDB と同様，かなり精緻に分析する必要がある．
- KF の場合は，少々の分析モデルの変更は，KF-QL と XSLT との変数化で汎用的枠組みを与えているため，分析モデルをインクリメンタルに強化していくことが可能で，プロトタイピング的なアプローチで構築することができる．

(3) 分析画面作成フェーズ
- RDB-OLAP は OLAP ツールとして強力な環境を提供している．
- XDBMS は画面遷移機構は XSLT などにより作り込まなくてはならない．また，XQL/XPath をベースにしたクエリは基本的にはパス検索であるためデータの結合，生成は DOM の API などを用いてプログラムを作り込むことになり多大な開発工数が発生する．
- KF は前節で述べたように，KF-QL とクエリの外部リンク機構を利用して，画面遷移で必要となるクエリ，スタイルの対応付けを行う環境を提供することで OLAP を実現できる．したがって，基本的にはプログラムレスで分析画面を作成することができる．

(4) 分析モデル変更フェーズ
- RDB-OLAP の場合，分析モデルの変更追加が難しく，硬直化した OLAP になってしまうとの指摘が多くなされている．近年，フレキシブルな構築環境を標榜するツールも出現してきている．
- XDBMS では分析モデルの変更は，作り込みのプログラムに影響を与える可能性が大

☆ 市販されている XQL/XPath をベースにした XML ネーティブデータベースを XMLDB と記述する．

きい．そのあたりをつねに見通しながら変更を進めていく必要があり，コスト高の要因になる．

- KF は分析モデルを任意に追加・修正することにより，ダイナミックに対応することができる．ただし，現在対応しているクエリのテンプレートは，分析モデルに対応する要素を数え上げて集計するクエリである．たとえば，売上高の括りを 1 億円単位に集計するか，10 億円単位に集計するか，ダイナミックに切り替えることには，現在対応できていない．これに対応するためには，XML データベースに新たに，売上高クラスタグを作り，事前に生の売上データから，括りに対応するクラス値（例：A：0〜10 億，B：10〜20 億，…）をクエリにより生成することにより対応している．図 11 の点線の矢印が XML データベースで行っているのはその意味である．これは，新規タグを必要に応じて逐次追加していくことが可能なため，このような方式を採っている（他の 2 方式にもデータベースにフィードバックしているのは，同様な理由によるもである）．

6.2 開発実験

テキスト版 OLAP 構築環境の生産性を評価するため，以下のような実験を行った．

(1) 実験内容

5 章の説明で利用したいろいろな観点軸で特許マップを作成し分析できる特許情報分析システムを例として，まったく同一な仕様を以下の 2 方式でシステム構築をしてその生産性を比較した．

- KF のテキスト版 OLAP 構築環境を利用してシステムを構築．
- 前節で述べた XDBMS での開発に対応させるため，KF をベースにし，KF-QL の機能のうち，XQL/XPath に対応する機能のみを使い，また，クエリの外部リンク機能や，パスの変数化機能を利用せずに（KF のサブセット）DOM API などを利用しシステムを構築．

(2) システム規模

- 特許情報：約 1,000 件
- 特許情報の要素数：53 個/特許
- データベース容量：約 7.3 MB
- 分析モデルの大項目：機能，技術，会社，評価，出願年
- 分析モデルの深さ：Max 4

表 2 開発工数比較（単位：人日）
Table 2 Comparison table of development cost.

開発フェーズ	KF-OLAP	XDBMS
データベース作成	3	3
分析モデル作成	4	12
分析画面作成	2	10
分析モデル変更	1	4
合計	10	29

(3) 実験結果

実験開発の結果は**表 2** のとおりであり，約 3 倍の生産性を実現した☆．

ここで，「分析モデルの変更」は，出願年を 2 年括りで分析したいという要求に対して対応した結果を示している．

また，特許情報システムとほぼ同程度なデータボリュームである「消費者クレーム分析システム」においても，約 2 週間の工数で実現することができた☆☆．

7. 考察と今後の課題

本論文では，次の 2 点について述べた．

1 つ目は，我々が考えるテキスト版 OLAP についての基本的考え方と，それを実現するためのベースとなる XML の DBMS である KF の基本機能を提案した．

2 つ目は KF で実現している KF-QL の機能（XQuery にサポートされていない機能）を利用したテキスト版 OLAP 構築環境についても述べ，開発効率が向上することを実験評価した．

［テキスト版 OLAP 高度化に向けて］

- **曖昧検索による分析：**より柔軟な検索・分析を行うために類似検索の機能は必須である．類似語辞書の活用とともに，KF の検索最適化方式の改良を行っている．
- **XML クエリ言語標準化への貢献：**既述のように XQuery の標準化作業が進んでいる．KF も XQuery をサポートすることを考えている．しかし，XQuery は「クエリの変数化」と「外部クエリのリンク」には対応していない．この機能の有用性を use-case を示しながら提案していきたい．

☆ 同一システム開発における習熟度の問題は KF での開発を先行して行ったので XDBMS の方が有利になっている．

☆☆ 4.2節で述べた文書要素抽出のための辞書構築の工数は含まれていない．

しかし「外部クエリのリンク」に関してはKF-QL自体XML表現になってることに依存している面がある．XQueryX（XQueryのXMLシンタックス）がWorking Draft [13]として2001/7に出ており，これをベースに検討を進める．

［構築環境の高度化に向けて］

- **画面生成ツールの高度化：**6.1節(4)で述べたように，ドリルダウンや，スライシングのためのクエリテンプレートは，分析モデルに対応する要素を数えあげて集計するものである．集計粒度を変更するには，分析モデルの体系を，その粒度に対応させて文書データを更新して分析する必要がある．これを分析画面からダイナミックに行えるようにするためには，この粒度も変数化してテンプレート化する必要がある．また，そのときのユーザインタフェースも容易なものにする必要がある．

参 考 文 献

1) 野中郁次郎ほか：知識管理から知識経営へ，人工知能学会学会誌，Vol.16, No.1, pp.4–14 (2001).
2) QL'98: The Query Language Workshop. http://www.w3.org/TandS/QL/QL98/
3) W3C: W3C XML Query Working Group. http://www.w3.org/XML/Group/Query
4) Microsoft: XQL. http://www.w3.org/TandS/QL/QL98/pp/xql.html
5) Deutsch, A., et al.: XML-QL:A Query Language for XML. http://www.w3.org/TR/NOTE-xml-ql/
6) Chamberlin, D.D., et al.: Quilt: An XML Query Language for Heterogeneous Data Sources, *3rd International Workshop on the Web and Databases*, Vol.WebDB2000, pp.53–62 (2000).
7) McHugh, J., et al.: Lore: A Database Management System for Semistructured Data, *ACM SIGMOD Record*, Vol.26, No.3, pp.54–66 (1997).
8) McHugh, J. and Widom, J.: Query Optimization for XML, *VLDB'99, Proc. 25th International Conference on Very Large Data Bases*, September 7–10, Edinburgh, Scotland, UK, pp.315–326, Morgan Kaufmann (1999).
9) Shanmugasundaram, J., et al.: Relational Databases for Querying XML Documents: Limitation and Opportunities, *Proc. 25th International Conference on Very Large Data Bases*, pp.302–314 (1999).
10) Florescu, D.: Storing and Querying XML Data using an RDMBS, *IEEE Data Engineering Bulletin*, Vol.22, No.3, pp.27–34 (1999).
11) Yoshikawa, M., et al.: XRel: A Path-Based Approach to Storage and Retrieval of XML Documents using Relational Database, *ACM Trans. Internet Technology*, Vol.1, No.1 (2001).
12) Deutsch, A., et al.: Storing Semistructured Data with STORED, *SIGMOD Record* (*ACM Special Interest Group on Management of Data*), Vol.28, No.2 (1999).
13) W3C: W3C XML Syntax for XQuery 1.0 (XQueryX). http://www.w3.org/TR/xqueryx

(平成13年12月20日受付)
(平成14年 3 月28日採録)

（担当編集委員 國島 丈生）

野々村克彦

1991年東京工業大学工学部情報工学科卒業．1993年同大学大学院理工学研究科情報工学専攻修士課程修了．同年,（株）東芝入社．現在，同社研究開発センター知識メディアラボラトリー研究主務．知識処理技術，XMLデータベースによるナレッジマネージメントの研究および開発に従事．

服部 雅一（正会員）

1988年慶応義塾大学理工学部管理工学科卒業．1990年同大学大学院修士課程修了．同年,（株）東芝入社．現在，同社研究開発センター知識メディアラボラトリー研究主務．知識工学，高次推論技術の研究・開発を行い，現在，XMLデータベースによるナレッジマネージメントの研究および開発に従事．人工知能学会会員．

櫻井　茂明

1989 年東京理科大学理学部応用数学科卒業．1991 年同大学大学院理学研究科数学専攻修士課程修了．同年,（株）東芝入社．1998 年から 2 年間，新情報処理開発機構つくば研究センタ出向．現在，同社研究開発センター知識メディアラボラトリー研究主務．現在，テキストマイニングシステムの研究開発に従事．博士（工学）．電気学会，人工知能学会，日本ファジィ学会，日本感性工学会各会員．

磯部　庄三

1987 年東京大学工学部計数工学科卒業．1989 年同大学大学院工学系研究科計数工学専攻修士課程修了．同年,（株）東芝入社．現在，同社研究開発センター知識メディアラボラトリー．計算機アーキテクチャの研究・開発を行い，現在，XML データベースによるナレッジマネージメントの研究および開発に従事．

末田　直道（正会員）

1973 年武蔵工業大学経営工学科卒業．同年東京芝浦電気（株）（現（株）東芝）入社．同社の研究･開発部門にて知識処理技術，高次推論，ナレッジマネージメント，XML データベース等の研究に従事．現在，大分大学工学部知能情報システム工学科教授．博士（工学）．人工知能学会，AAAI 各会員．