

事例に基づく XML データベースの適用検討

服部 雅一 (東芝ソリューション株式会社)
谷川 均 (東芝ソリューション株式会社)
近藤 雄二 (東芝ソリューション株式会社)

概要 電子政府をはじめとして、電子取引、事務データ、新聞記事など様々なデータを汎用的に表現する仕組みとして、データ構造の柔軟性を特長とする XML (Extensible Markup Language) の活用が進んでいる。XML の規格策定当初はインターネット上のデータ交換が主な用途であったが、現在では XML によるデータ交換は完全に定着した。更に、企業内において XML の活用範囲が広がるにつれ、XML データをそのままの形式で管理し活用したいというニーズが高まっている。そのニーズに対応できるのが「XML データベース (XMLDB)」である。本稿では、XML が持つ高い柔軟性を活かした XMLDB についての基本思想や操作方法を述べるとともに、ドキュメント管理アプリケーション開発の事例を通して XMLDB 活用のメリットについて議論する。

1. はじめに

XML (Extensible Markup Language) は、データだけでなくタグを使ってデータの意味や構造を記述するためのマークアップ言語である。1998年2月、W3C(World Wide Web Consortium)にて XML 1.0 が策定され、今や IT を語る上で無くてはならないキーワードの一つになっている。

策定当初、XML は優れたメタ言語として、B2B などインターネット上のデータ交換分野で注目を集めた。新聞業界の NewsML、電子商取引分野における ebXML、医療業界の MML などが代表的なものと言えるが、データ交換のため XML を使ったデータ標準化が各種業界で進められた。

さらに近年、XML をデータ交換という一時的な用途に用いるだけでなく、XML データをそのままの形式で管理し活用したいというニーズも高まってきた。

管理すべきデータをその構造化の度合いから分類すると、構造データ、半構造データ、非構造データの3つに分類できる。厳格な構造を持つ売上げデータや従業員データなどは構造データに属し、メモなど構造を持たないものは非構造データに属する。

その中間に位置する半構造データとは、構造が不規則なデータでスキーマがはっきりと定義できないデータを指す。この半構造データの例として製品データや議事録が挙げられる。製品データは製品ごとに属性が異なり、議事録は開催日時や参加者など予め決められた構造もあるものの、それ以外の章立てなど構造は自由である。企業内のデータの多くが半構造データに属しており、XML はこれら半構造データを表現するのに適しているだけでなく、構造データや非構造データも横断的に表現できる。

企業が保有するデータのうちリレーショナルデータベース (RDB) などデータベースで管理されているものの割合は 10~20%程度と言われている。残りの約 80~90%の大半を占めるのは半構造データや非構造データである。このようなデータは、従来の RDB では管理しにくいデータである。そのような状況で登場したのが XML データの管理に特化した XML データベース (XMLDB) である。

本稿では、XML が持つ高い柔軟性を活かした XMLDB についての基本思想や操作方法を述べるとともに、ドキュメント管理アプリケーション開発の事例を通して XMLDB 活用のメリットについて議論する。

2. XMLDB とは

2.1. XMLDB の特長

XMLDB は XML データを管理するためのデータベースであり、その一般的な特長 3 点を記す。

① 階層的な構造を持つデータの管理に適している。

RDB は行(レコード)と列(カラム)からなる表(テーブル)でデータを管理する。それに対して XML の構造は階層的であるため、XMLDB は階層構造型データモデルを持つデータベースになっている。

4.2 で例示するが、XMLDB の基本的な操作方法についてここで簡単に説明する。データを格納する階層構造上の領域をパスで指定して XML データを登録する。また、パスに対する条件検索で XML データを参照する。データ更新時には、XML データ単位での更新に加え、XML

データ内のノード指定した部分更新が行える。このように XMLDB は階層構造を操作するのに直感的でわかりやすい操作方法を提供している。

また、製品によって実装している機能はまちまちだが、多くの XMLDB は大量の XML データでも高い信頼性でデータを管理するデータベース管理システムとして開発されている。それらの XMLDB は同時実行制御、トランザクション管理、リカバリ機能、アクセス権制限などの機能を実装している。

② データ構造の変更に対して柔軟性を持つ。

XML の柔軟性を損なうことなくデータ管理するため、XMLDB はスキーマ定義を必要としない「スキーマレス」なデータベースになっている。

ここで図 1 に示すように PC や冷蔵庫などさまざまな製品データを RDB で管理することを考えてみる。RDB では事前に正規化などきちんとしたスキーマ設計を行う必要がある。製品には“CPU”や“容量”など製品固有の属性や“メモリ”など可変個の属性があるが、これらの属性を表のカラムとして定義しなければならないため、結果として 4 つの表ができる。

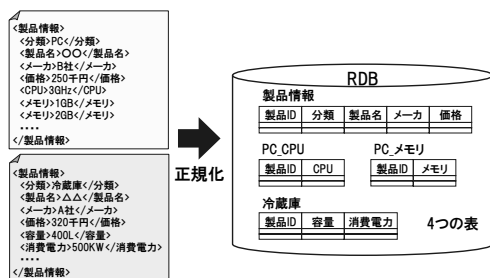


図1 RDB管理の問題点

この状況で、TV というこれまでに無い製品データの追加やPCの“3D対応”という新規属性の追加などのシステム変更が発生しうる。その場合、表全体の見直し、表の追加、表に対するカラムの追加変更といったスキーマ変更が発生してしまう。スキーマ変更が発生すると、索引の再構築も含めRDBではかなりの作業コストがかかる。また、RDBを呼び出すアプリケーション側の変更作業も発生する。例えば、表全体の見直しなど大規模なスキーマ変更が発生すると、修正対象の更新文(UPDATE文)だけでなく直接関係の無い更新文にも変更が伝播してしまい、アプリケーションの保守コストも無視できない。

一方、XMLDBでは製品データであるXMLデータをスキーマ定義無しにそのまま管理し、属性の追加に対し

ても特別な作業は不要である(図2)。XMLDBを呼び出すアプリケーション側の変更作業も、修正対象の更新文だけで大半は済み変更が伝播する確率は低い。このため、XMLDBではRDBに比べ、データ構造の変更に対して柔軟性が高いと言える。

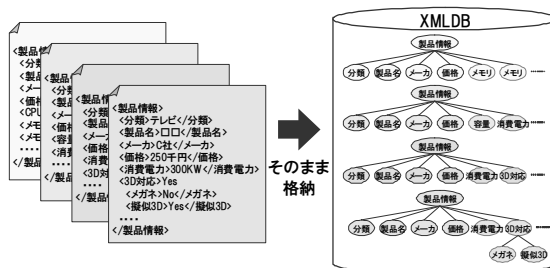


図2 XMLDB管理の場合

③ ドキュメント指向のXMLデータに対する全文検索機能が強力

ドキュメントと呼ばれるテキスト中心のXMLデータ(ドキュメント指向のXMLデータと呼ぶ)の統合的な検索のために全文検索機能は欠かせない。議事録のドキュメントでは、日付や参加者など構造データとともにテキストデータも共存しており、このようなテキストデータの検索をアシストする機能として全文検索機能が重要である。

2.2. XMLDB の分類

XMLDBには、XMLデータの管理に特化したネイティブXMLDBと、RDBにXMLデータ管理機能を強化したハイブリッドXMLDBの2種類がある。XMLDBの特長を実現するため、XMLデータの問い合わせ機能や索引設定機能など格納や検索に関する基本機能は、ほとんどの製品に備わっている。以下にネイティブXMLDBとハイブリッドXMLDBの特長を述べる。

2.3. ネイティブXMLDB

XMLデータを従来型のRDBで管理する場合、CLOB(Character Large Object)のようなテキストでの格納方式、もしくは、XMLデータの要素をRDBのレコードに対応させるマッピング方法のいずれかを選択する。しかし、CLOBでは検索時のXMLデータのパーズ自体が大きなオーバーヘッドになりXPathやXQuery[1]などの検索に対して高いパフォーマンスを発揮するのが難しい。マッピ

ング方式でも、XML のデータ構造の変更時に、それに合わせて表の定義を変更しなければならないというデメリットが発生する。そのような状況で生まれたのが、XML データの管理に特化したネイティブ XMLDB である。

ネイティブ XMLDB の多くは、XML データを DOM(Document Object Model)のようなオブジェクトのツリーに展開した上で格納する。パース済みで格納されるため、検索時においても再度パースする必要はなく、RDB におけるマッピング処理も不要である。多くのネイティブ XMLDB では、オブジェクトツリーの構造に対する索引や、テキストや数値に対する索引が自動的に付加される(オートインデックスと呼ばれる)。XPath や XQuery などの検索に対して、ネイティブ XMLDB はそれらの索引を使うことで高いパフォーマンスを発揮することができる。

現在、第2世代と言われるネイティブ XMLDB が製品化されている。第1世代と比べて技術改善が進み、数十ギガバイトから数テラバイトまでの大規模データを扱えるようになっている。また、トランザクション管理機能、RDB 並みのチューニング機能、並列検索機能なども強化されている。様々な機能は RDB に一日の長はあるものの、ネイティブ XMLDB は日々進化している。

一般的にネイティブ XMLDB は、ハイブリッド XMLDB と比べて全文検索機能が豊富で、ドキュメント指向の XML データ管理を得意としている。

2.4. ハイブリッド XMLDB

従来の RDB ではうまく XML データを管理できなかったのは前述した通りだが、RDB に XML データ管理機能を強化したハイブリッド XMLDB と呼ばれる XMLDB が登場した。すなわち、RDB と XMLDB の特長を持つという意味でハイブリッド XMLDB と呼ばれている。

ハイブリッド XMLDB ではレコードのカラムに格納した XML データに対して、専用の補助構造を用意している。つまり、検索などに際してよく使用する要素や属性に対して、内部的に索引を用意することで検索に対するパフォーマンスを向上させている。

一般的にハイブリッド XMLDB は RDB をベースとしているだけに表に近い XML データの管理に適しており、トランザクション処理が重要なアプリケーションにも適している。

図3に RDB と XMLDB の位置付けを示す。縦軸にデータの構造化の度合い、横軸に業務系と情報系の分類を示している。業務系とは航空券予約システムや在庫管理

システムなど主要業務を処理するシステムのことで、大量トランザクションが発生するのが特徴である。業務系のデータは定型的で構造がほぼ決まっているので RDB の表で管理しやすい。

一方、情報系としては全文検索やデータ分析などの検索パターンが例として挙げられる。これまでの XMLDB は主に半構造データから非構造データまでを管理する情報系で使われることが多い。逆に、高いトランザクション性能が期待される業務系システムへの適用は XMLDB の今後の課題と言える。

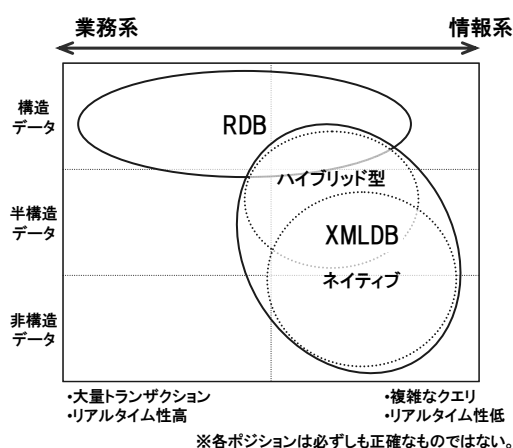


図3 XMLDB の適用領域

3. ネイティブ XMLDB TX1

東芝ソリューション(株)は、企業内の XML データを管理することを目的とした XMLDB, TX1[®]を 2005 年に商品化した[2]。テラバイト級の XML データでも管理できるのが特長で、この特長を生かして適用ソリューションを広げ、特許情報検索、新聞コンテンツ管理、電子申請情報管理、防災情報共有、規程管理など、さまざまな情報システムに適用されている。以下、TX1 の特長を列挙する。

3.1. テラバイト級でも高速検索

TX1 は、独自開発した「構造自動抽出技術」や「問い合わせ最適化技術」などの技術を基にテラバイト級の XML データでも XQuery の問い合わせを高速処理できる。さらに最新バージョンではデータベースを分割し並列検索する「分散並列検索技術」の開発により数十テラバイトクラスの大容量データでも高速処理することが可能となった。

3.2. 多彩な全文検索手段

ドキュメント指向の XML データ検索のために全文検索機能は欠かせない。TX1 は全文検索機能が豊富で、XQuery の全文検索拡張言語である XQuery and XPath Full-Text[3]もサポートしている。

キーワードを含むデータを高速かつ正確に取り出す全文検索機能を実現するには語彙索引が必要である。TX1 では、N グラム方式と形態素解析方式を XPath で指定するパスごとに使い分けることができる。N グラム方式は隣接する文字列で索引となる文字列を切り出す方式で漏れのない厳密な検索ができる。一方、形態素解析方式は言語の規則や辞書に従い意味のある単語で索引となる文字列を切り出す方式でテキストの意味を考慮した検索ができる。

また、日本語の表記方法は多様で指定したキーワードのままでは意図したデータを検索できないことがある。そのため、大小文字や全半角文字のように表記が違って同じ意味となる文字を区別なく検索する同一視検索や、同じ意味を持つ単語を同時に検索する同義語検索機能を提供している。さらに、複数のキーワードの距離を使った検索やキーワードの出現順序を指定する検索などの近傍検索機能や tf-idf 計算式を使って重要度の高いデータから取り出すことを可能とするスコアリング機能を提供している。

頭で思い浮かべた曖昧なフレーズでデータを検索できれば、全文検索はより簡単に行なえる。TX1 の自然言語検索機能では自然文をそのまま検索条件にすることができる。例えば、「XML を使って統合したシステムの例」という自然文をキーワードにした場合、自然文を解析して得られた重要単語(“XML”“使って”“統合”“システム”“例”など)を含むデータを取り出すことができる。

さらに、ある文章を元にして関連するデータを検索したいという要求も高い。TX1 の関連データ検索機能は自然言語検索機能の拡張機能で、指定されたデータから自動的に重要な単語を選び出し、近い意味を持つデータを取り出すことができる。例えば、特許提案時に作成した提案メモの文章をキーワードにして、公知例調査における類似特許の検索を行なうことができる。

関連の強い内容のデータを取り出したい場合や、確実に検索できるキーワードがわからない場合などの検索ではフィードバック検索機能が有効である。フィードバック検索機能では、まずキーワードに指定された自然文による検索を行ない、検索結果の上位ランクのデータから

重要語を関連語として抽出し、検索条件に追加して二次検索を行なう。

3.3. DB に求められる信頼性を確保

TX1 はデータベース管理システムとして求められる信頼性を確保するため、同時実行制御、トランザクション管理、リカバリ機能、アクセス権制限などの機能を実装し、大量データでも安心してデータベースを運用できる。同時実行制御には、MVCC (Multi Version Concurrency Control : 多版型同時実行制御)を採用している。参照と更新でロックが競合しないのでマルチユーザ環境でのスループットが向上する。アクセス権制御は、XML データを最小単位として XPath で指定するパスごとに権限の設定が可能である。

また、オンラインバックアップをはじめ、DB のインポート/エクスポートなどの運用機能も装備している。高可用性システム構築のため、クラスタソフトウェアと組み合わせることにより、可用性の高いシステムを構築することができる。

3.4. データ統合向けの機能

データ統合向けの機能として TX1 では RDB、IBM® Lotus Notes(以下、Notes)、CSV ファイルからデータを収集し XML 変換して TX1 へ登録するデータ連携機能を提供している。このデータ連携機能を活用すれば、既存システムはそのまま運用しながら、複数システムに分散したデータを一元管理するデータ統合システムを短期間で構築できる。

また、データ連携に特化した他社製データ連携ミドルウェアを利用することも可能である。これを使えば、上述したデータ連携機能ではサポートしていないタイプのデータを変換して TX1 へ登録することができる。また、TX1 から取り出したデータを他の DB へ登録することもできる。

4. XMLDB の基本的な操作方法

以下、ネイティブ XMLDB の TX1 を使って、XMLDB の基本的な操作方法について説明していく。

4.1. データモデル

まず、DB のデータモデルについて言及しておく。RDB

の基礎になっているものは、40年前 E. F. Codd によって打ち立てられたリレーショナルデータモデルである。一方、XMLDB のデータモデルはまだ発展段階にあり、デファクトスタンダードも含め標準化は進行中である。ほとんどの XMLDB では、データの基本単位は(例えば 1 ファイルの)XML データであり、RDB のレコードにほぼ相当する。RDB の表に相当するものは、XMLDB ではコレクションやフォルダと呼ばれて複数の XML データを格納する。

XML の高い記述能力を最大限に引き出すべく W3C で標準化された XQuery は、RDB の SQL に相当する問い合わせ言語であり、コレクションやフォルダを起点とした問い合わせを行う。

4.2. API の概要

問い合わせ言語と異なり標準規格のない API は、XMLDB ごとに独自方式を採用している。TX1 では、検索用 API は RDB での開発経験が有効に生かせるように JDBC をベースに、更新用 API は XML 更新の標準 API である DOM をベースにしている。XQuery の拡張仕様 XQuery Update Facility[4]が XML データを更新するための仕様として策定中だが、全ての XMLDB がこの仕様準拠しているわけではない。また、XMLDB と RDB の間で大きく違うのは、問い合わせ言語が SQL か XQuery かという点と、API を使ってやり取りするデータが XML データかレコードデータかという点であり、システム構成面で大差はない。

① XML データの検索

TX1 による Java 言語を使った検索のサンプルプログラムを記載する。RDB でサポートされている JDBC との違いは小さい。

```
// (1)XMLDBに接続
TSXConnection conn =
    TSXDriverManager.getConnection("jxapi@srv1:9199",
        "username","password");
// (2)ステートメントを作成
TSXStatement stmt = conn.createStatement();
// (3)検索の実行 (XQuery)
TSXResultSet rset =
    stmt.executeQuery(
        "for $e in db('testSpace')//book
        where $e/publisher='Addison-Wesley' return $e");
// (4)ヒット件数の表示
```

```
System.out.println(rset.getResultCount());
// (5)検索結果の表示
while (rset.next()) {
    // (6)文字列として取得
    System.out.println(rset.getXMLString());
}
rset.close(); // (7)リザルトセットをクローズ
stmt.close(); // (8)ステートメントをクローズ
conn.close(); // (9)接続をクローズ
```

② XML データの登録

TX1 による Java 言語を使った XML データ登録のサンプルプログラムを記載する。基本的な処理の流れは検索と同じであるが、検索により登録位置を決めて XML データを登録する流れになる。

```
// (1)XMLDBに接続
TSXConnection conn =
    TSXDriverManager.getConnection("jxapi@srv1:9199",
        "username", "password");
// (2)ステートメントを作成
TSXStatement stmt = conn.createStatement();
// (3)検索の実行 (XPath) 登録位置を決める
TSXResultSet rset =
    stmt.executeXPath("db('testSpace')/bib",true);
// (4)XML登録処理
rset.next();
TSXNode node = rset.getNode();
node.appendChild("<book year='1994'><title>TCP/IP Illustrated</title><author><last>Stevens</last><first>W.</first></author><publisher>Addison-Wesley</publisher><price>65.95</price></book>");
conn.commit(); // (5)コミット
rset.close(); // (6)リザルトセットをクローズ
stmt.close(); // (7)ステートメントをクローズ
conn.close(); // (8)接続をクローズ
```

5. XMLDB の活用分野

XMLDB を活用できるか否かを見極めるもっとも重要なポイントは、運用しながらデータ構造の変更や拡張が必要なシステムであるかどうかである。XMLDB のためのデータ設計やプログラミングなどの解説やガイドが少ないのも一因ではあるが、現在でも RDB と比べて XMLDB の普及度は低いのが実情である。RDB を使い慣れた設計者やプログラマが、XMLDB の活用を検討すべ

きケースを以下に挙げる。

① 開発時に仕様が固まらないケース

今日のビジネス環境の激しさから、顧客要求の変化への迅速な適応、新たなサービスやデータとの統合、など IT システムにも“アジリティ”（俊敏さ）が求められている。データ構造の変化に対して追従する柔軟なデータベースとして XMLDB はこのような目的に合致する。

例えば、ビジネス要件も流動的で開発時には仕様が固まらなくても、短期間にカットオーバーしなければならないシステム開発案件が増大している。このような案件で XMLDB が採用されるケースも増えている。例としてインターネット上のオンラインショッピングサイトの構築を考えてみる。ショッピングサイトは、取り扱う商品数は膨大で商品ごとに属性が違うはずである。サイト構築後も取り扱う商品数が年々増加することを考えると、このような状況で正規化作業は困難である。このようなときは「スモールスタート」の発想で、商品データ DB には XMLDB の適用を検討すべきであろう。

その延長ではあるが、インターネットを介してアプリケーションを提供する SaaS(Software as a Service)においても XMLDB を適用するケースが増えるだろう。SaaS の大きな魅力の 1 つはすぐに使える利便性にあるが、ユーザの機能やデータ構造に対する要求に応じてカスタマイズできないと、その魅力は失われてしまう。今後は SaaS のデータ基盤にも XMLDB の適用を検討すべきであろう。

② データが複雑な階層構造を持つケース

データ交換のための XML データの標準化が進んでいる業界でも RDB を使わずに、XMLDB でデータを管理するケースが多い。例えば、新聞記事(NewsML)、放送コンテンツ(J/Meta)、診療データ(MML)、さらには特許公開公報などが挙げられる。これらは明示的なスキーマ定義は存在するものの、定義されているデータ構造が複雑で、さらに業務向けに独自タグを必要に応じて入れる可能性もある。このようなデータ構造は RDB の表(テーブル)にマッピングしにくく正規化作業が面倒になる。XMLDB が選ばれるのはこのような理由による。

XMLDB(TX1)の事例として、新聞コンテンツ管理システムを紹介する。このシステムは NewsML で記述された新聞データ(記事原稿や写真)を効率的に管理・運用するための基盤システムであり、XMLDB はカスタマイズされた NewsML データを管理している。この基盤システムを使って完成した記事の保管、検索や配信などを効率的に実現している。

③ 企業内データを統合するデータハブ

企業では業務のシステム化が進み、業務で使われているさまざまなデータが電子化されているが、データが個別システム内に閉じて管理・利用されるため、情報資産の潜在価値を十分引き出せない現状がある。そこで既存データを再利用に適した XML 形式に変換してデータ統合するアプローチで XMLDB が利用される。

XMLDB(TX1)の事例として、保守販売情報統合システムを紹介する。このシステムでは、異なるシステムで管理されている障害情報、製品情報、販売情報を XMLDB を使ってデータ統合した。ここでは、名寄せなどにコストがかかる前処理は最低限にして、まずデータを XMLDB にためてしまう。その代わり検索時に、同一視検索や同義語検索などの全文検索の諸機能呼び出す XQuery を使って不統一なキーワードや条件を吸収する戦略をとった。これによりミニマムコストでデータ統合ができた。

このような環境が出来上がると、XQuery を使ってデータを組み合わせることで新たなデータを作成することもできる。例えば、ある顧客クレームからクレーム原因となった該当部品を含む製品一覧のドキュメントを作成したり、ある顧客クレームから今後クレームが発生しそうな案件一覧のドキュメントを瞬時に取り出したり、様々なドキュメントを作成することができる。このようなコンテンツ系のマッシュアップにおいて、作成効率の高い XQuery は今後有望なコンテンツ処理言語になると思われる。

④ データやドキュメントを細粒度で管理するケース

契約書や保険書の約款は、共通の条文や条項などを組み合わせたもので、新規作成時には既存ドキュメントからのコピーと修正で対応することが多い。このような運用では、既存ドキュメントを一部修正すると、その派生ドキュメントも全て手作業で修正しなければならない。このような手作業ではコストがかかり、1 つのミスで企業の信用を失墜させることにもなりかねない。マニュアルなどの技術系ドキュメントでも同様で、製品バージョンアップのたび、あるいは不具合を修正するたび、マニュアル書換えや Web サイトの関係ページを修正しなければならない。

このような課題に対して、ドキュメントを部品化し、必要な部品を組み合わせる形でドキュメントを作成できれば、個々の部品を修正するのみで当該の部品を利用する全てのドキュメントの内容更新を一元管理できる。ま

た、新規作成のコストも削減できる。

ここでXMLDB活用の具体的なメリットをマニュアルを例に考えてみる。マニュアルは、技術系ドキュメントのXML規格であるDITA(Darwin Information Typing Architecture)[5]で書かれているものとする。DITAではドキュメントをトピックという複数の部品に分解し、マップによってドキュメントを構成する。XMLDBではトピックとマップをデータ管理し、XQueryを使ってマップ内のトピックへの参照データを頼りにトピックを組み合わせて、製品バージョンごとのマニュアルを自動生成することも可能である。これならば個々の部品を修正するのみで当該の部品を利用する全てのドキュメントの内容更新が自動化できる。また、手作業の修正が必要であっても、部品の属性として派生に関する版データが埋め込まれていれば、派生元の部品一覧などの検索ができるので修正効率もアップする。

XMLDB(TX1)の事例として、規程管理・公開システムを紹介する。業務規程や商品販売マニュアル、社内手続き集などの社内ドキュメントを条項単位でXML化してXMLDBで管理する。閲覧時には、曖昧検索や同義語検索など多彩な全文検索機能を使って表題や条文などの見たいところを表示する。また、セキュリティ上の観点から、閲覧者の条件によって特定のドキュメントが閲覧できないようにアクセス制限を設定している。

6. ドキュメント管理アプリケーションへの適用

6.1. 番組情報基盤システムの概要

XMLDB(TX1)を効果的に活用した事例として、テレビ局の番組情報基盤システムを紹介する[6]。本システムは5章で挙げたケースのいずれにも少なからず該当しているが、主に企業内データを統合するデータハブに相当する。

近年、ハードウェアの進歩やインフラの整備など、テレビ業界を取り巻く環境は大きく変化している。例えば、テレビ局は番組コンテンツをデジタル管理し、視聴者もインターネットを通して番組コンテンツなどの動画を自由に視聴できるようになっている。そのため、テレビ局には環境変化に対応した多種多様なサービス提供が求められている。

このような状況で、このテレビ局では番組情報を資産として二次利用、三次利用など有効活用できるシステムの基盤構築が必要になり、複数のシステムに分散してい

た番組に関わる情報をXMLDBで一元管理した。本システムは、今後ますます多種多様化する視聴者の要望に 대응するためにテレビ局が打った重要な布石のひとつとなっている。

6.2. システムの構成

図4は番組情報基盤システムの構成図である(ラテ:新聞などのラジオ、テレビ欄用のデータ)。番組情報を管理するXMLDBを中心に、利用者及び外部システムとのインターフェースを提供している。

6.3. システムの特長

① 番組情報のXMLデータ

本システムで扱っている番組情報は、放送番組などのコンテンツ流通に必要とされる標準メタデータ体系J/Meta[7]をベースに、このテレビ局独自のカスタマイズを行った約400項目からなる数十万のXMLデータである。XMLDBを使うことで標準規格と独自データの共存を容易に実現している。これらのデータには、番組タイトル、出演者、EPGなどの番組の内容に関する情報、制作会社やスタッフなど制作に関する情報、使用した著作物や利用した楽曲など権利に関する情報など様々な情報が含まれる。

② 番組情報の入力

現状、番組情報は様々な既存システム上に存在している。例えば、放送予定の番組情報は編成を管理する営業放送システム、番組を保存したテープ媒体の保管場所や内容の情報はテープ管理システムにそれぞれ蓄積されている。本システムでは、これらの外部システムとデータ連携ミドルウェアを用いた連携により、番組情報の生成と更新を行っている。

また、番組に関連する情報には、番組情報の他にも字幕データ、台本など放送業務で扱う様々な情報があるが、そのほとんどはMicrosoft®Word(以下、Word)やMicrosoft®Excel(以下、Excel)などで作成された非定型データ(番組関連ファイル)である。本システムでは、これらの番組関連ファイルもXMLデータに変換してXMLDBで一元管理している。

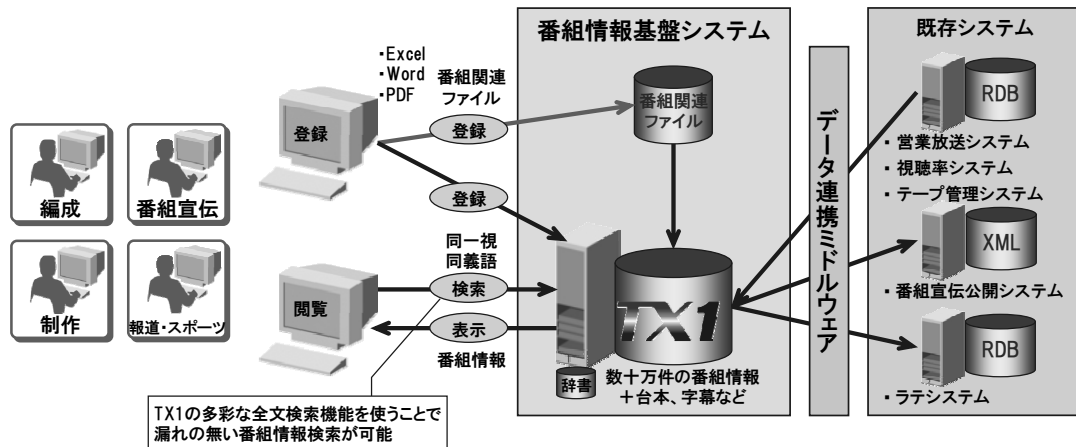


図4 システム構成図

③ 全文検索との組み合わせによる漏れのない検索

本システムでは、400もの膨大な項目から「放送日や出演者など項目を指定した検索」と「キーワードによる全文検索」を組み合わせた検索によって、効率よく目的の番組情報を探し出すことができる。ここでもXMLDB(TX1)の全文検索機能が持つ様々なオプションを使うことで検索漏れを防止している。例えば、同義語辞書を利用した同義語検索によって、表現の揺らぎがあった場合でも検索漏れを防ぐことができ、また、ひらがなカタカナ同一視検索や異体字(名字によく使われる旧字など)同一視検索を用いることによって、表記に揺れがあった場合でも漏れなく検索することができる。これらはXQuery and XPath Full-Textで問い合わせを記述する。

これ以外にも、XMLDB(TX1)には、データ中のwhitespace(タブ, 改行, 復帰, 空白)と全角空白をすべて除いて検索する機能がある。本システムでは、番組関連ファイル(Word, Excelなど)からXMLデータへの変換が行われるために、次のような見映えを揃えた、テキスト内に空白や改行を含んだXMLデータが作成される場合がある。

```

□<abstract>↓
□□□XMLはタグを使ってデータを表現した構造↓
□□□化データである。↓
□</abstract>↓

```

この例の場合「構造化データ」という単語の中に改行と空白文字を含むため、通常、「構造化データ」をキーワードとする全文検索ではヒットしないのだが、本システムでは、このようなデータでも漏れなく検索することが出来る。

④ 番組情報の出力

蓄積された番組情報は、利用者による検索、閲覧だけでなく、番組情報を必要とする番組宣伝公開システムなど他のシステムへもXMLデータとして配信している。その際、膨大な項目数の番組情報から、必要なデータ項目だけを抜き出し、外部システムが求める形式で出力することを容易に実現できる。本システムでもXSLTを利用することで、様々な外部システムとの連携に柔軟に対応している。

6.4. XMLDB 導入のメリット

このシステムでは、異なる既存システムで管理されている様々なデータをデータ構造の変更にも強いXMLDBを使ってデータ統合した。また、データ連携ミドルウェアと組み合わせることで、既存システムを変更すること無しに短期間でシステム構築することができた。

番組情報は、将来に亘って蓄積され利用されるものであり、今後ビジネスや運用の変化によってデータ項目の追加、変更が発生することが考えられる。そのような場合にも本システムは、XMLDBのメリットを生かして柔軟に対応することができる。例えば、番組情報に新しいデータ項目が追加された場合でも、XMLDBには新しい構造でそのまま格納できる上、XQueryとXQuery and XPath Full-Textを組み合わせることによって、新しい構造のXMLデータも漏れなく検索することができる。

7. おわりに

これまでXMLDBについての基本思想や操作方法を述べるとともに、ドキュメント管理アプリケーション開発

の事例を通して、XMLDB 活用のメリットについて述べてきた。上述したように、XMLDB の活用分野はドキュメント管理系を中心にかなり広いのだが、RDB の普及に比べれば XMLDB の普及はまだまだと言わざるを得ない。XMLDB 普及を推進するためには、やはり XML や XMLDB に関する技術力を持つエンジニアの育成が必要と感じており、啓蒙活動も含め長期的な視点で進めていくつもりである。

一方、XMLDB への追い風が吹き始めている。まず、現在進行形だが Office Open XML や OpenDocument Format などオフィススイート用のファイルフォーマットが XML で標準化されている。今後、オフィス系ドキュメントの XML 化が進むはずである。

また、ソフトウェア開発やシステム開発においても XML 化の流れが加速している。システムは、プレゼンテーション、ビジネス、データという代表的な三層アーキテクチャで分解して説明されるが、近年、Web Service, Ajax, BPEL/BPMN, XForms など XML ドリブンのコンポーネントが、各層で OSS, あるいはプロプライエタリ製品として数多く生み出されている。今後、これらコンポーネントと XMLDB を駆使することで設定のみの“ゼロコーディング”による超短期システム開発の可能性が生まれている。

このように追い風は吹いている。XMLDB が普及するためには、XMLDB を使ったシステムの開発方法論の確立や XMLDB の高いトランザクション処理への対応などが急務であると思われる。

参考文献

- 1) W3C: XQuery 1.0: An XML Query Language,
<http://www.w3.org/TR/xquery/>
- 2) 東芝ソリューション(株): XML データベース,
<http://www.toshiba-sol.co.jp/pro/xml/>
- 3) W3C: XQuery and XPath Full Text 1.0,
<http://www.w3.org/TR/xpath-full-text-10/>
- 4) W3C: XQuery Update Facility 1.0,
<http://www.w3.org/TR/xquery-update-10/>
- 5) DITA XML.org community site, <http://dita.xml.org/>
- 6) 小川淳: 番組情報の XML 化による一元化と番組制作・再利用業務の効率化, 東芝ソリューションテクニカルニュース, Vol.14, pp.4-5(2008).
- 7) 財団法人マルチメディア振興センター: 汎用メタデータ体系 J/Meta,
http://www.fmmc.or.jp/fmmc-html/jmeta/PDF/JMetaPresentation_fm0.pdf

服部 雅一 (正会員)

E-mail: masakazu.hattori@toshiba.co.jp

1990 年 (株)東芝入社. 1999 年から XML データベースの研究開発に従事. 現在, 東芝ソリューション(株)プラットフォーム事業部に勤務. 第 55 回電気科学技術奨励賞(電気科学技術奨励賞委員会会長賞), 平成 20 年度関東地方発明表彰(日本弁理士会会長奨励賞)受賞, 平成 21 年度 喜安記念業績賞受賞.

谷川 均 (非会員)

E-mail: Tanigawa.Hitoshi@toshiba-sol.co.jp

1993 年 (株)東芝入社. 以来, ナレッジマネジメントおよび XML データベースの開発に従事. 現在, 東芝ソリューション(株)プラットフォームソリューション事業部に勤務. 第 55 回電気科学技術奨励賞(電気科学技術奨励賞委員会会長)受賞, 平成 21 年度 喜安記念業績賞受賞.

近藤 雄二 (非会員)

E-mail: Kondo.Yuji@toshiba-sol.co.jp

1992 年 (株)東芝入社. 2002 年から XML データベースの開発に従事. 現在, 東芝ソリューション(株)プラットフォーム事業部に勤務.

投稿受付: 2010 年 9 月 30 日

採録決定: 2010 年 11 月 16 日

編集担当: 柴山 悦哉 (東京大学)