XML Consortium

XQuery/XMLDB利用のお心得

2008年6月5日

XMLDB部会 技術系サブグループ

XML Consortium

活動関係者(敬称略)

IBM 大沼、高橋
オラクル 大野、中村
東芝ソリューション 望月、矢野
サイバーテック 山口
富士ソフト 小川
凸版印刷 伊藤
日外アソシエーツ 久我
日立製作所 千種
日本電子専門学校 海野
日立システム 村垣、藤春

その他多くの皆様から、お話しを頂きました。

アジェンダ

1.技術系サブグループ・活動内容
2.現状考察
3.発表内容
4.XMLDB利用時、要注意な事柄
5.XMLDBにおいて最初に知っておきたい事柄
6.現状のXMLDBを使用する際の留意事項
7.今後の課題

1．技術系サブグループ・活動内容(1)

加藤XML部会リーダーの宣言

不定型データのシステム化というテーマを厳正に考察し、XMLDBへの曖昧な幻想を排除する。
-> XMLDB普及のキャズムを乗り超えたい。

技術サブグループとしては、技術者が何を考え、どういう事柄をクリアすべきか考察

曖昧になっている点、当たり前に解決可能であると見なしている点を明らかにする。

1．技術系サブグループ・活動内容(2)

1回/月の部会おける活動で考察を深める。

- ベンダ各社のご協力によるXMLDB製品の紹介
- XMLデータ設計の実際についての紹介
- サブグループでのディスカッション
- マーケティングサブグループからの刺激
- 様々な立場のメンバ
  (ベンダ、SIer、ユーザ企業)

2．現状考察(1)(利用頻度)

XMLデータは身近に流通している

自ら作成することは必ずしもないかもしれない。
しかし、受取ることは多くなっている。

XMLDBに格納することが珍しくはなくなってきている

元データがXMLなら、XMLで格納したい。
文書型のXMLの場合、全文検索を行いたい。
WebアプリケーションはXMLデータを出力する。

2．現状考察(2)(学習環境)

日本語の情報

従来から、概要・製品紹介情報はあった。
ここに来て書籍(メンバも執筆)が出揃ってきた。
各DB固有の情報も比較的入手しやすくなった。

フリーのXMLDB

以前は体験版以外は、XIndice位しかなかった。
最近では、以下のようなXMLDBが入手可能。

Sedna、eXist、MonetDB/XQuery etc ...

フリーのRDBもSQL/XMLのサポートがなされる。

2．現状考察(3)(構築環境)

ベンダー製XMLDB

性能・機能面での強化が著しい。
製品毎に殊に機能面でのアピールポイントが
かなり異なる。
各製品の得意とする所を見極めた導入が必要。

構築ツール

XMLを出力するオフィスツール。
Ajax、リッチクライアントのような柔軟性の高い
インタフェース開発技術・ツール。
XMLアプリケーションの統合開発ツール。
RoRのような容易にDBアクセスができる統合フレ
ームワーク。

3．発表内容

これからXMLDBを利用される方々への技術的なお心得となるトピックをいくつか取り上げます。

XMLDB利用時、要注意な事柄
　エンコーディング、整形式など。

XMLDBにおいて最初に知っておくとよい事柄
　ネイティブ、ハイブリットによる違いなど。

現状のXMLDBを使用する際、留意すべき事柄
　サポート機能、XQueryなど。

4．XMLDB利用時、要注意な事柄(1)その１

(1) XMLデータのエンコーディング

ex.手作業でデータをXML化。XML宣言では、エンコーディングをUTF-8に指定。全体をShift_JISで格納。

```
<?xml version="1.0" encoding="UTF-8"?>
<goods_list>
    <goods no="1"><name>Pen①</name>
    <price unit="円">200</price></goods>
</goods_list>
```

4．XMLDB利用時、要注意な事柄(1)その２

色々なエディタで開いてみる。

```
<?xml version="1.0" encoding="UTF-8"?>
<goods_list>
    <goods no="1"><name>Pen①</name>
    <price unit="円">200</price></goods>
</goods_list>
```

[エディタ１]
Shift_JISとして認識。encodingと不一致。

```
<?xml version="1.0" encoding="UTF-8"?>
<goods_list>
    <goods no="1"><name>Pen@</name>
    <price unit="~">200</price></goods>
</goods_list>
```

[エディタ２]
UTF-8として認識。そのまま開くと、要素、属性の値が文字化け。

4．XMLDB利用時、要注意な事柄(1)その３

データ作成時の留意点

① XMLについて経験の少ない方には、エンコーディング指定と格納エンコーディングを一致させる事を徹底して頂く。

② 複数XMLデータを結合させる(コマンドなどを使って、バイナリ結合する等)際、異なるエンコーディングを混在させないよう留意して頂く。

③ 機種・エンコーディング依存文字に注意。コンバータを使用しても別エンコーディングにコンバートできない文字がある。

4．XMLDB利用時、要注意な事柄(1)その４

データ利用時の留意点

① パースした結果NGの場合、そもそもエンコーディング指定がおかしい事を疑う。

② 複数エンコーディングが混在していることもある。ただし、この場合のチェックは難しい。

③ 外字が混じっている場合、外字は、別文字に置換える、外字は絵として扱い、XMLには絵の格納位置を示すURIを指定するなどの対処が必要。

④ データが大きい場合、パーサの処理性能も考慮。

4．XMLDB利用時、要注意な事柄(2)その１

(2)整形式なXML

ex.HTMLデータを格納しようとする場合、大抵開始-終了タグの対応がとれていない。

ex.アプリケーションから出力されるデータも、必ずしも整形式であるとは保証されていない。

ex.手作業で作成されたデータの場合、稀に < や & 等が実体(文字)参照で記述されていない事もある。

これらのデータは、XMLDBへの格納前に整形式に直しておく必要がある。勿論、(1)で述べたエンコーディングにも留意する。

4．XMLDB利用時、要注意な事柄(2)その２

整形式であっても、そのまま格納すると後々の活用に支障を生じることもある。

ex.目視でのレイアウトを優先し、正味のデータ内容の後ろにスペースが充填されているXMLデータ。

ex.更新を繰り返した結果、更新内容が元の内容と離れた位置に追加されているようなXMLデータ。

これらのデータは、XMLDBへの格納前にXQueryやXSLTなどを用いて、後から取り出し易い形に直しておく必要がある。

4．XMLDB利用時、要注意な事柄(3)その１

(3)大きなXMLデータ

最近は、PCのハードディスクもテラバイト級のものが個人でも入手可能。

XMLDBも以前に比べ大容量データに対応。

実際に受取るデータもギガバイトを越えるような場合が珍しくはなくなった。

しかし、大きなXMLデータをそのままXMLDBに格納するのが必ずしも最善の保存方法とは限らない。

4．XMLDB利用時、要注意な事柄(3)その２

XMLDB製品のアーキテクチャ上、１個のデータは大きすぎず小さすぎず程々の方がよい。

保存データの利用を考慮しても、分類基準を定め、適切な大きさに分割した方がよい。

そのためには、クエリー内容とXMLデータ構造双方を勘案してDBの物理構造まで決定することが必要。

5.XMLDBにおいて最初に知っておきたい事柄(1)その１

(1)XMLデータの格納形態

①ネイティブXMLデータベースの場合１

```
<root>
  <goodsList>
    .......
  </goodsList>
  <salesList>
    .......
  </salesList>
</root>
```

ルート要素下に全データが位置付けられるタイプ。データを区分するために、データ種別毎に、例えば、商品データの集まりの場合、goodsListのようなタグを設定してデータを括ること等が考えられる。

5.XMLDBにおいて最初に知っておきたい事柄(1)その２

②ネイティブXMLデータベースの場合２

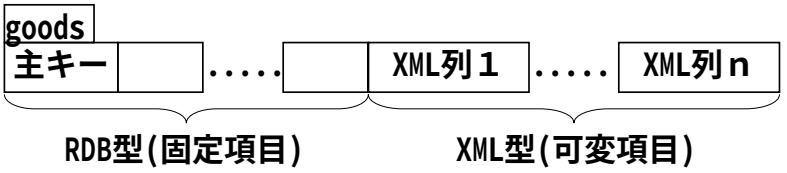
商品
```
<goodsList>
    .......
</goodsList>
```

売上
```
<salesList>
    .......
</salesList>
```

データ種別毎に、格納領域を分け、名前を付けることができる。例えば、商品情報の場合、商品という名前の領域に格納する。

5.XMLDBにおいて最初に知っておきたい事柄(1)その３

③ハイブリッドXMLDBの場合



データ項目中、変更のない項目はRDBのデータ型が設定されたカラムに、データ構造に柔軟性を求められる項目はXML Typeのカラムに格納。

5.XMLDBにおいて最初に知っておきたい事柄(1)その４

(2)データ格納場所の指定

①ネイティブXMLデータベースの場合１

```
<root>
  <goodsList>
    .......
  </goodsList>
  <salesList>
    .......
  </salesList>
</root>
```

全データがルート要素下にあるので、格納場所の指定は不要。データソースは、ルート要素からのパスを指定するだけでよい。

ex.
/root/goodsList/goods

5.XMLDBにおいて最初に知っておきたい事柄(1)その５

②ネイティブXMLデータベースの場合２

商品

```
<goodsList>
    .......
</goodsList>
```

売上

```
<salesList>
    .......
</salesList>
```

格納領域の名前を指定し、次にデータソースのパスを指定する。

ex.
doc("商品")/goodsList/goods

5.XMLDBにおいて最初に知っておきたい事柄(1)その６

③ハイブリッドXMLDBの場合

商品		
主キー	RDB型列	XML型列

テーブルとXML型列を指定することで、当該列の全てのXMLデータから成るシーケンスを得る。

ex.

XQueryの場合

```
doc("商品.XML型列")/goods
```

SQL/XMLの場合

```
XMLQUERY('$g/goods'
    PASSING "商品.XML型列" AS "g")
```


5.XMLDBにおいて最初に知っておきたい事柄(2)その１

格納データの検索と更新

製品により、特徴(優劣ではない)が大きく異なる所。

ex．ネイティブXMLデータベース

・全文一致検索を高速に
　インデックス(形態素、N-gram等)作成には時間がかかるが検索は全文一致でも極めて高速。

・完全一致検索を高速に
　インデックス作成の負荷が軽いため、データ更新に対する適応性も高い。製品によっては、更新面に特に力を入れているものもある。

5.XMLDBにおいて最初に知っておきたい事柄(2)その２

ex.ハイブリッドXMLデータベース
基幹業務と同様に、検索・更新のバランスをとった大規模トランザクション処理を重視。
なお、製品によっては全文検索機能が含まれている場合もある。

検索・更新に限らず、現状、製品間でかなり機能面のアピールポイントに違いがあるようなので、目的に合った製品の選択が必要。

6.現状のXMLDBを使用する際の留意事項(1)

(1) 今後のXMLDB普及により解決されていく事柄であると考えるもの

製品によるXQuery機能サポート範囲の違い

XQueryの更新系機能の標準化
(2008-03-14 W3C勧告候補)

XQueryの全文検索機能の標準化
(2008-05-16 W3C勧告候補)

各プログラミング言語におけるAPIの標準化

データバインディング用のフレームワーク充実

6.現状のXMLDBを使用する際の留意事項(2)

(2) XQuery式における型の認識

XQueryは型を認識する。スキーマなしでも利用できるが、その際暗黙的にどのような型として認識されるかは、演算子によっても異なるため要注意。

```
スキーマなし <sec>10</sec> eq "8"   → false
             (xs:stringとして比較)
スキーマあり(sec要素の型がxs:integer)
              <sec>10</sec> eq  8    → true
スキーマなし
  <sec>10</sec>   > 8          数値比較
  <sec>10</sec>   > <sec>8</sec>      文字列比較
```


6.現状のXMLDBを使用する際の留意事項(3)その１

(3) APとDBとの処理の切り分け

XMLDBを使用すれば、XMLデータを必要以上に分割せず、自然な形で表現できる。

そのため、半定型で複雑なデータ表現も可能。

このようなデータを検索するXQueryは、単なるI/O言語と割り切りずらく、プログラミング言語のような記述も行えるようになっている。

DBパフォーマンス、XQueryコードの可読性も考慮し、アプリケーションコードとXQueryコードとの間でロジック部分の切り分けが必要。

6.現状のXMLDBを使用する際の留意事項(3)その２

次スライドは、XML Query Use Cases の最初から数えて４番目のユースケースである。

XQueryでは、order byはサポートされているが、group byは直接サポートされていないため、まずdistinct-values関数を使ってグルーピング項目から重複を除いている。

グルーピングが複数項目(last要素とfirst要素)にわたっているため、正確にクエリー内容を追っていくには、ある程度の予備知識が必要。

ただ、一般のプログラミング言語でこれを行うのも煩雑。この位のXQueryは必要であるとも考える。

6.現状のXMLDBを使用する際の留意事項(3)その３

```
<results>
{
  let $a := doc("http://bstore1.example.com/bib/bib.xml")//author
  for $last in distinct-values($a/last),
      $first in distinct-values($a[last=$last]/first)
  order by $last, $first
  return
    <result>
      <author>
        <last>{ $last }</last>
        <first>{ $first }</first>
      </author>
      {
        for $b in doc("http://bstore1.example.com/bib.xml")/bib/book
        where some $ba in $b/author
            satisfies ($ba/last = $last and $ba/first = $first)
        return $b/title
      }
    </result>
}
</results>
```

著者毎に、その著作のタイトルをグルーピングして返す。

6.現状のXMLDBを使用する際の留意事項(3)その４

```
[http://bstore1.example.com/bib/bib.xml]
<bib>
  <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>
  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>
  ........
</bib>
```

全ページのXQueryのクエリ対象データ。

7.今後の課題

XMLDBの技術者に対するアピール

DBMSの「管理者」には、XMLDBであっても管理がそう変わる訳ではないことをアピールする。

データ構造を最初から固定しなくてもよいので、XMLDBは「開発者」にアピールできる。

「エンドユーザ」にも、フロントエンドに適切なツールがあればアピールできる。

「開発者」に対する、格納のためのXMLデータ設計方法、「エンドユーザ」に対する、ユーザインタフェース作成のためのXMLデータ利用方法の普及等が必要。