



XML Consortium

【第7回XMLコンソーシアムWeek】

XMLDBを活用した WebAPIの実装

2008年6月6日

上村 準也 (キャノンソフト情報システム)

荒本 道隆 (アドソル日進)

松山 憲和 (PFUソフトウェア)

高橋 公一 (NEC)

Copyright © XMLコンソーシアム 2008 All rights reserved.

3部会合同プロジェクト

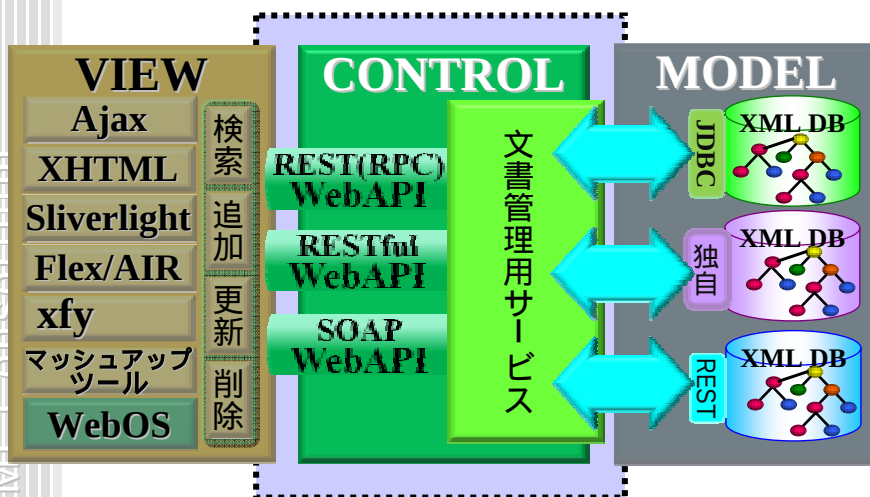


- 3部会合同プロジェクトの一端として、CONTROLに相当する部分の試作を行いました

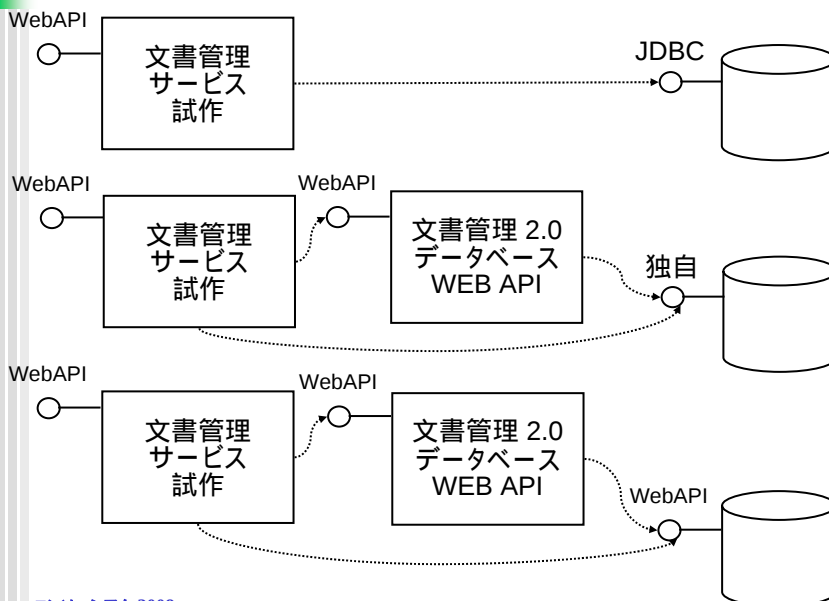
XML Consortium

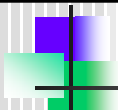
Copyright © XMLコンソーシアム 2008 All rights reserved.

システム構成 (案)



システム構成 (試作)





利用させていただいたDB



- IBM
 - DB2 v9.5
- 日立製作所
 - HiRDB HiRDB / Single Server Version 8 08-03
 - HiRDB XML Extension Version 8 08-00
- サイバーテック
 - NeoCoreXMS 3.1.3
- 東芝ソリューション
 - TX1 V2

番外: PostgreSQL 8.3.1

この資料に記載されている製品名などは各社の商標または登録商標です。また、正式な表記と異なる場合があります。

Copyright © XML Consortium 2008 All rights reserved.



実装したWebAPI



- 文書管理用REST API
 - 「文書管理 2.0 データベース WEB API」とほぼ同じ仕様のもの
- 文書管理用REST API + RSS
 - クライアントで扱いやすいデータとして、上記にRSSを加えたもの
- 文書管理用SOAP API
 - Webサービス実証部会で実験的に仕様を決めて実装したもの

Copyright © XML Consortium 2008 All rights reserved.

DBへのアクセス方法



■ ハイブリッドDBでのテーブル定義

```
-- データベースの作成
create database feature;

-- データベースへの接続
connect to feature;

-- テーブルの作成
create table xmltable (id char(8) not null primary key, doc xml);

-- データのインポート
import from xml_list.csv of del xml from xml replace into xmltable;

-- レコード数の確認
select count(*) from xmltable;
```

Copyright © XML Consortium 2008 All rights reserved.

DBへのアクセス方法



■ JDBCによるアクセス

```
Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery(query);
while(resultSet.next()){
    SQLXML sqlxml = (SQLXML) resultSet.getObject(1);
    Source returnSource = sqlxml.getSource(StreamSource.class);
    /*
    sqlxml.getBinaryStream();
    sqlxml.getString();
    sqlxml.getCharacterStream();
    sqlxml.getSource(DOMSource.class);
    sqlxml.getSource(JAXBSource.class);
    sqlxml.getSource(SAXSource.class);
    sqlxml.getSource(StAXSource.class);
    */
}
```

Copyright © XML Consortium 2008 All rights reserved.

DBへのアクセス方法



- デモ
 - ドキュメントIDを指定して文書を取得

DBへのアクセス方法



- ネイティブなDBでは
 - テーブルの定義なし、インデックスも自動
 - TX1の管理画面の例



DBへのアクセス方法



- 独自のAPIによるアクセス
 - NeoCoreXMSの場合

```
private void doGetArchiveDocument(  
    String documentID, HttpServletResponse response,  
    SessionManagedNeoConnection connection)  
    throws Exception {  
  
    String query = DefaultNamespaceContext.DECLARE_NAMESPACES  
        + "/ND/doc:doc[doc:metadata/dc:identifier=\"" + documentID + "\"]";  
    String document = XMLUtils.trimQueryResults(connection.queryXML(query));  
  
    if (document.length() == 0) {  
        response.setStatus(HttpServletResponse.SC_NOT_FOUND);  
        outputMessage(response, "文書が存在しません。[文書ID=\"" + documentID + "\"]");  
    } else {  
        response.setContentType("application/xml");  
        BufferedWriter writer = getBufferedWriter(response);  
        writer.write(document);  
        writer.close();  
    }  
}
```

Copyright © XML Consortium 2008 All rights reserved.

DBへのアクセス方法



- デモ
 - ドキュメントIDを指定して文書を取得

Copyright © XML Consortium 2008 All rights reserved.

RSSデータの作成



- XQueryを使ってRSSデータを作成
 - 必要なデータの検索と同時に作成できる

RSSデータの作成



- デモ
 - RSSリーダーで作成したデータを取得

まとめ: JDBCによるアクセス



- 既存環境の延長から簡単にアクセス
- SQLと同じ方法でXQueryを発行できる
- 低レベルAPIでXMLを操作しなくても良い
 - アプリケーション層の負担が減るのは明らかなメリット
 - SQL/XML型は問い合わせの結果をラッピングしたもの

まとめ: Web APIによるアクセス



- インターネット経由のアクセスも簡単
- クライアント・アプリケーションから、直接呼ぶこともできる
- 任意のXQueryを実行する機能によって、特別な準備なしで、欲しいデータを欲しい形で受け取れる
 - 新しいWebサービスと同じ

まとめ: 独自APIによるアクセス



- XQueryをいきなり投げれば良い
- XPath指定でデータの更新が可能
- そもそも行指向でなく、IDのようなものも意識しなくて良い

試作を終えて



- XML名前空間によって複雑さが増すことがある
- XQueryにはまだ慣れない
 - 慣れれば楽なはず、複雑だがコンパクトで、時間を節約できそう
 - SQLでいうVIEWに近い感覚もある
- ネイティブなXMLDBは確かに型を考えずに使うことができる



XML Consortium

【第7回XMLコンソーシアムWeek】



ご静聴ありがとうございました

引き続きWeb2.0部会の発表をどうぞ。