

使用 Java 和 XML 实现数据移植

渠本哲, 张 凝, 王潜平

(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221008)

摘 要:异构数据源的移植问题是当前数据处理领域内研究的一个热点,通过数据移植能有效地利用信息资源、实现数据共享。作为数据表示和交换的工业标准,XML的应用日益广泛,JDOM是一个应用Java语言对XML文档进行读写等操作的API,结合两者的优点,提出了一种以XML文件作为中介实现数据移植的方法。该方法结合Java和XML技术,使用JDOM和JDBC接口实现了数据移植功能,并给出了详细的设计与实现。

关键词:Java;XML;JDOM;JDBC;数据移植

中图分类号:TP311.13

文献标识码:A

文章编号:1673-629X(2006)09-0015-03

Implementation of Data Migration with Java and XML

QU Ben-zhe, ZHANG Ning, WANG Qian-ping

(College of Computer Science and Technology, China University of
Mining and Technology, Xuzhou 221008, China)

Abstract: The migration of heterogeneous data sources, which can exploit information resources effectively and realize data sharing, is one of the vexed points in the current data processing field. As the industrial standard of data presentation and exchange, the application of XML is becoming more prevalent. JDOM is a new API for reading, writing, and manipulating XML document within Java code. Combining their virtues, introduces a method for data migration through XML document. The method integrates Java with XML, realizes data migration with the interface of JDOM and JDBC, and a detailed design and implementation are given.

Key words: Java; XML; JDOM; JDBC; data migration

0 引 言

XML(eXtensible Markup Language)即可扩展标记语言,是互联网联合组织(W3C)创建的一种表示数据的工业规范。XML规范实现了数据表示的结构化和层次化,在当前的软件开发中XML在数据的存储和传递方面已经得到了越来越广泛的应用。数据库数据与XML格式之间数据的转换,在当前的软件开发中已经占据了越来越重要的位置^[1]。文中利用JDOM,实现了通过XML文件进行两任意数据库管理系统之间的数据移植。

1 主要实现技术

1.1 JDOM 技术

现在Java语言中处理XML文档的方法主要有DOM(Document Object Model), SAX(Simple API for XML), JDOM(Java Document Object Model)几种^[2],其中JDOM是最适合Java开发人员使用的一种处理XML文档的方式。JDOM是用Java开发的一个开源API,用来读写和操

作XML文档,其具有编程语言的唯一性,只能使用Java操作,而其它几种XML API都是跨语言的,同时要支持几种编程语言,因此JDOM执行效率较高,而且JDOM针对Java语言进行了优化,充分利用了Java语言中的许多优点,使用Java的一些特性如方法重载、集合和映射等,充分提高了JDOM的性能^[3,4]。JDOM与其它的标准如SAX和DOM的互操作性很好,甚至可以说它就是对这些API的简化抽取,JDOM从这些已存在的API中提取它们最好的概念,建立了一系列新的类和接口供人使用,而且和它们相比JDOM更直观、简单并容易使用。

1.2 JDBC 技术

JDBC(Java DataBase Connectivity)为Java程序访问数据库提供了一个统一的接口,它是Java与数据库系统通信的标准API,由一组用Java语言编写的类和接口组成,通过JDBC可以完成数据库连接、执行SQL命令等操作。JDBC为数据库应用开发人员提供了一组标准API,使开发者能够用纯Java语言编写数据库应用程序,使用JDBC API可以很容易地把SQL语句传送到任何数据库中并执行。同时,Java语言程序还可以通过JDBC来访问计算机上已定义的ODBC数据源。JDBC API是Java中数据库应用的基础,它建立了Java数据库程序同各种不同数据库的沟通机制。

收稿日期:2005-12-20

作者简介:渠本哲(1981-),男,江苏丰县人,硕士研究生,研究方向为XML数据库、网络与分布式计算;王潜平,教授,研究方向为数据库, CSCW。

2 数据移植原理

本方法实现数据移植的基本思想就是把 XML 文件作为中介,首先从源数据库中读取数据并按照 XML 格式写入到一个 XML 文档,然后读取并分析该 XML 文件再将数据添加入目标数据库。本方法充分利用了 Java 的平台无关性和 XML 的格式统一性,从而可以完成两台不同机器上不同系统不同数据库之间的数据移植。如图 1 所示。



图 1 数据移植原理

3 具体实现

3.1 使用 JDBC 实现对数据库操作的类

本例中要使用 JDBC 对数据库进行查询和插入操作,因此需创建一个纯 Java 类,该类适用于任何数据库管理系统,而且能够处理任一 SQL 命令对数据库进行操作。将该数据库操作类命名为 manipulateDB,其包括 3 个成员函数,描述如下:

(1)public manipulateDB(String driver,String URL):连接数据库。两个参数分别为 JDBC 驱动接口类名和数据库广义地址。

(2)public ResultSet selectFromDB(String SQL):执行数据库查询操作。参数为实现查询操作的 SQL 语句。返回一个 ResultSet 对象,其中包括了查询的结果。

(3)public void insertIntoDB(String SQL):执行数据库插入操作。参数为实现插入操作的 SQL 语句。

3.2 使用 JDOM 实现对 XML 文档操作的类

本例中使用 JDOM 对 XML 文件进行写入和读取操作,因此需创建一个操作 XML 文件的 Java 类。该 XML 文档操作类命名为 manipulateXMLDoc,其中包括了两个成员函数,描述和解释如下:

● public void writeToXML (String rootName, String tableName, String fileName, ResultSet rs):生成并写入 XML 文档。rootName 是 XML 文档中根元素名称,tableName 是 XML 文档中元素项名称,fileName 是 XML 文件名,rs 是要写入 XML 文件的数据库查询结果数据集。使用 JDOM 生成 XML 文件需要如下步骤:

(1)创建一个 XML 文档。

```
Element XMLRoot = new Element(rootName);
Document jdomXMLDoc = new Document(XMLRoot);
```

(2)添加元素和子元素。

```
Element student = new Element(tableName);
XMLRoot.addContent(student);
student.addContent(studentContent);
```

(3)生成 XML 文件。

```
XMLOutputter xmlDoc = new XMLOutputter(" ",
```

```
true,"GBK");
```

```
xmlDoc.output(jdomXMLDoc, new FileOutputStream(
(new File(fileName))));
```

● public Vector readFromXML (String fileName):从 XML 文档中读取数据。fileName 是要读取的 XML 文件名。本函数返回数据值的向量列,其形式如“value1”,“value2”,“value3”。使用 JDOM 读取 XML 文件内容需要如下步骤:

(1)指定一个 XML 文件。

```
SAXBuilder sb = new SAXBuilder();
```

```
Document jdomXMLDoc = sb.build(new FileInputStream(
(fileName)));
```

(2)得到 XML 文档根元素。

```
Element XMLRoot = jdomXMLDoc.getRootElement();
```

(3)读取 XML 文档的元素和子元素。

```
List studentFirList = XMLRoot.getChildren();
Element studentEle = (Element)studentFirList.get(i);
valueStr = studentEle.getText();
```

3.3 实现读取数据库数据写入 XML 文件操作的类

本例中先对源数据库进行操作。首先通过数据库操作类建立与数据库的连接,通过查询 SQL 语句得到数据,然后将查询结果以参数的形式传递给 XML 文件操作类,再生成 XML 文件^[5]。如图 2 所示。

创建一名为 readData 的类实现上述操作。基本步骤如下:

(1)读取数据库数据。

```
manipulateDB sourceDB = new manipulateDB (driver,
url);
```

```
ResultSet rs = sourceDB.selectFromDB(sql);
```

(2)写入 XML 文档。

```
manipulateXMLDoc XMLDoc = new manipulateXMLDoc();
```

```
XMLDoc.writeToXML (rootName, tableName, fileName,rs);
```

3.4 实现读取 XML 文件写入数据库操作的类

接着本例需对目标数据库进行操作。首先通过 XML 文件操作类读取 XML 文档内容,然后将得到的内容以参数形式传递给数据库操作类,通过插入 SQL 语句添加数据,从而实现数据移植。如图 3 所示。

创建一名为 writeData 的类实现上述操作。基本步骤如下:

(1)读取 XML 文档内容。

```
manipulateXMLDoc XMLDoc = new manipulateXMLDoc();
```

```
Vector sqlVec = XMLDoc.readFromXML(fileName);
```

(2)写入数据库数据。

```
manipulateDB destinationDB = new manipulateDB(dri-
```

er,url);

destinationDB.insertIntoDB(sql);

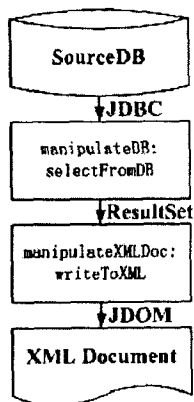


图2 读取数据写入XML文件

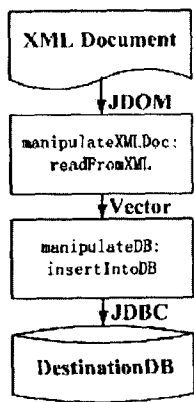


图3 读取XML文件写入数据

4 应用实例

本例中使用 JBuilder9 作为开发平台,实现了不同机器之间从 Access 2003 到 SQL Server 2000 数据库之间的数据移植。在 Access 中建立一个 student 数据库,并在其中建立一名为 student 的表,并输入若干数据。

先执行 readData 类生成 XML 文档,其格式如下:

```
<? xml version="1.0" encoding="GBK"? >
<Students>
  <Student>
    <Name>ZHE</Name>
    <ID>2005786</ID>
    <Sex>M</Sex>
    <City>Jiangsu</City>
    <Telephone>05163889086</Telephone>
  </Student>
  .....
</Students>
```

再将此 XML 文档拷入目标机器,执行 writeData 类将

数据存入 SQL Server 数据库,数据移植操作即完成。

5 小结

文中尝试使用 JDBC 和 JDOM 技术实现了一个用 XML 文件作为中介进行数据移植的应用,而且通过一个实例演示了如何使用 JDOM 对 XML 文件进行读写操作。

当今世界的数量正在飞速膨胀,应用系统也在不断增大,应用与数据库之间、数据库与数据库之间的数据交换日益频繁,数据格式的不统一已经成为了软件业发展的障碍,因此将杂乱无章的数据处理成格式规范的 XML 格式文件,将为数据的交换和处理提供极大的方便。Java 作为现在的一种主流开发语言,其强大的功能加上 XML 在数据表示方面的优势,极大地提高了应用的易用性和可移植性。因此有理由相信,Java 与 XML 的协同运用在将来的系统开发中将会得到更为广泛的应用,同时 JDOM 作为一种专为 Java 语言设计的处理 XML 的方式,将会更加完善成熟。

参考文献:

- [1] 陶以政,唐定勇,何铁宁,等.基于 Java 和 XML 技术的异构信息系统数据集成框架应用研究[J].计算机应用研究,2004(5):38-40.
- [2] McLaughlin B. Java 与 XML[M].北京:中国电力出版社,2004.
- [3] Hunter J. JDOM and XML Parsing[EB/OL]. <http://www.oracle.com/technology/oramag/oracle/02-sep/o52jdom.html>,2002.
- [4] Hunter J, McLaughlin B. Easy Java/XML integration with JDOM[EB/OL]. <http://www.javaworld.com/javaworld/jw-05-2000/jw-0518-jdom.html>,2000.
- [5] Biggs W, Evans H. Simplify XML programming with JDOM[EB/OL]. <http://www-128.ibm.com/developerworks/java/library/j-jdom>,2001.

(上接第14页)

4 结束语

基于构件的软件开发方法(Component Based Software Development)有利于在软件开发中减轻重复劳动,提高软件的开发效率、质量和可维护性。

文中提出的基于 XML 的构件组装编译的解决方案,将构件组装机制中的组装描述进一步完善,对构件组装编译过程提供更好的支持,提高了软件的开发效率,增强了系统的灵活性、可扩展性和易维护性。改善和开发新的构件组装技术规范具有重要的理论和应用意义。

参考文献:

- [1] 王斌,王建新,张尧学,等.一种基于多 Agent 的 Internet 上 JavaBean 构件挖掘方法[J].小型微型计算机系统,

2003,24(12):43-52.

- [2] 杨美清,梅宏,李克勤.软件复用与软件构件技术[J].电子学报,1999,27(2):68-75.
- [3] 中科院软件研究所.中科院软件研究所网站[DB/OL]. <http://www.ios.ac.cn>.2003.
- [4] 普元公司.普元公司网站[DB/OL]. <http://www.primeton.com>.2004.
- [5] 互联网实验室.互联网实验室网站[DB/OL]. <http://www.chinalabs.com>.2004.
- [6] 窦郁宏,陈松乔.程序挖掘中需求描述的研究[J].计算机工程与应用,2002(10):53-56.
- [7] 任洪敏,钱乐秋.构件组装及其形式化推导研究[J].软件学报,2003,14(6):1066-1074.