文章编号:1001 - 9081(2007)03 - 0574 - 03

基于 Excel 及数据转换服务的异构数据集成方法

罗作民,李 悦,孙淑海,周红芳
(西安理工大学 计算机科学与工程学院,陕西 西安 710048)
(liyue315@sina.com.cn)

摘 要:针对异构数据库数据集成问题,在分析基于 SOA 架构三层设计模式的基础上,结合 Excel 和数据转换服务技术,提出了比较适合于中小企业实现异构数据转换系统的体系结构。详细介绍了组件系统层异构数据源在集成子系统层整合数据、判断采用 Excel 还是 DTS 并向全局数据库系统层进行数据转换及集成的过程。

关键词:面向服务架构;Excel;数据转换服务;异构数据集成
中图分类号: TP311.12 **文献标识码**:A

Heterogeneous data integration method based on Excel and DTS

LUO Zuo-min, LI Yue, SUN Shu-hai, ZHOU Hong-fang
(*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an Shaanxi* 710048, *China*)

Abstract: To integrate the data in heterogeneous databases, based on the design mode of SOA's three heterogeneous layers, the architecture of integrating application databases system for medium enterprises and small companies was proposed. This structure adopted two kinds of data transformation service technologies, Excel and DTS.

Key words: Service-Oriented Architecture (SOA); Excel; Data Transformation Services (DTS); heterogeneous data integration

随着信息技术的不断发展,企业在网络构建上也逐渐启用目前流行的高端或实用性较强的软硬件产品。不同产品不能很好融合,遗留系统更新所产生的数据安全性、完整性、可靠性也会存在很多问题。面向服务架构(Service-Oriented Architecture,SOA)作为此类问题的解决方案应运而生。本文根据 SOA 理念采用 Excel 和 DTS(Data Transformation Services)两种互补的异构数据转换技术,提出了一种基于 SOA 架构的异构数据集成系统的体系结构。Excel[1] 无需提取元数据且适合各种报表管理,但是数据以单元格为单位进行转换,代码量繁重;DTS 技术[2,3] 能完整地转换异构数据信息,且编码量小、易于维护,但该技术仅基于元数据的提取,对元数据难以挖掘的数据库信息不适用。两种技术相互结合,克服了各自技术的不足。

1 SOA 简介

SOA 是将软件应用程序构建成为可重用的商业服务集合的新型体系结构模式。它的出现标志着设计、开发新应用程序并将其与原有业务应用程序集成的方式出现了根本性变化,它将企业应用程序的开发简化为轻松进行集成和重用的模块化业务服务。它利用一系列网络共享服务,使 IT 能更紧密地服务于业务流程。通过采用能隐藏潜在技术复杂性的标准界面,SOA 能提高 IT 资产的重用率,从而加快了开发并更加可靠地交付新的增强后的业务服务[4]。因此,为企业应用提供各种灵活商业模块的 SOA 成为重量级厂商追逐的目标,包括 SAP、IBM、BEA、Oracle、微软等在内的厂商都竞相为此推波助澜。

2 异构数据集成系统体系结构

普通管理系统通常采用三层设计模式,即数据库开发层、业务流封装层和用户应用层。对于异构数据库集成系统的开发也可以采用三层设计模式,但是数据的流动特别是异构格式数据的流动就需要采用一种中间格式数据或一种提取元数据的方法作转化。因此这里基于 SOA 理念对异构数据库之间实现自动数据交换提出了一种解决方案,即在异构三层设计模式基础上,提出一种应用数据库系统集成的体系结构[5,6],并实现以 Excel 为中间数据格式和 DTS 技术互补的转换应用。如图 1 所示。

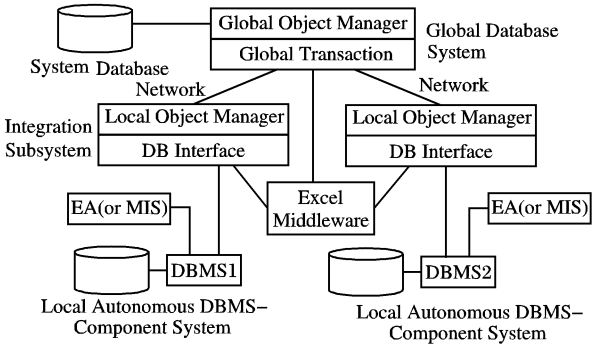

图 1 异构数据集成系统体系结构

图 1 中,组件系统 Component System 位于网络中的各个节点上。为了建立永久查询需即时更新整个多数据库系统的全局控制,在层次顶部仅设置一个全局数据库系统 Global Database System,通过全局用户引用全局查询和事务处理来使

收稿日期:2006 - 09 - 19;**修订日期**:2006 - 12 - 04 **基金项目**:西安市攻关计划项目(ZX04011)

作者简介:罗作民(1963 -),男,山西临猗人,副教授,主要研究方向:数据库技术、系统集成、Web 技术、网格计算; 李悦(1981 -),男,河北秦皇岛人,硕士研究生,主要研究方向:数据库技术、系统集成; 孙淑海(1978 -),女,陕西杨凌人,硕士研究生,主要研究方向:网格计算、数据库技术; 周红芳(1978 -),女,陕西大荔人,讲师,博士研究生,主要研究方向:数据挖掘.

用。全局查询根据访问数据的位置分割生成若干子查询,子查询不是直接发送给底层的组件系统,而是必须要经过一个集成子系统。每一个组件系统有它自己的集成子系统 Integration Subsystem,其功能是转换子查询来响应组件系统的调用(如每一条 SQL 查询语句)。自治本地事务对于全局数据库系统和集成子系统都是可见的,可以直接调用组件系统接口。全局数据库系统和集成子系统执行异构三层事务的管理模块。集成子系统与其组件系统定位在同一节点。在全局数据库系统和集成子系统之间的接口用分布式,全局数据库系统作为分布式系统被执行。Excel 中间件提供对各种异构数据库数据的导入接口;组件系统提供数据源及数据导出接口。该体系结构数据流程如图 2 所示。

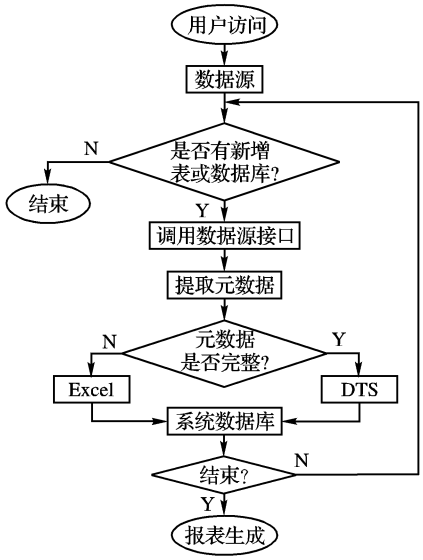

图 2　数据流程

3　基于 Excel 及 DTS 的异构数据集成

我们假设在全局数据库系统层的系统数据库使用 Microsoft SQL Server,图 1 中各组件系统层的数据库与全局数据库系统层的数据库互异。由于数据库可能出现系统操作平台上的不同,故集成子系统通过执行 Java 这种具有开放特性、跨平台的语言所撰写的程序,便可以完成数据写入全局数据库系统前各项工作。

MS SQL Server 的一个重要组件——DTS,使得管理员能够在服务器与数据库间转换信息。它整合了 Microsoft Universal Data Access 技术与 ActiveX 技术,其数据的导入、导出功能,既能应用于 ODBC 或 OLE DB 数据库(Oracle,Sybase SQL Server,Access 以及 FoxPro),也可以应用于存储类型(文本文件,Excel 及 Outlook)。由此可见,MS SQL Server 的 DTS 功能很强大,但是也存在一些问题:

1) 对于适合数据导入导出的异构 ODBC/OLE DB 数据库不能保证其数据类型的完整性。例如将 FoxPro 中的数据导入到 MS SQL Server 中。逻辑上,FoxPro 中的 NUMERIC 和 DATE 数据类型对应 MS SQL Server 中 FLOAT 和 TIMESTAMP 数据类型。但实际上,DTS 只记录了数据值,而数据类型全部以 VARCHAR 类型导入进来,而且数据全部分配可变的、上限为 255 字节的存储空间来存储,无法为数据提供合法性检验,破坏数据的完整性。

2) 如果企业内部遗留系统存在多个异构数据库,且处于不同系统平台,那么 DTS 就需要大量手工操作,而且也会出现整合数据可靠性差的问题。

3) 对于 ODBC/OLE DB 数据库、文件存储类型以外的数据格式,DTS 功能将不适用。

要解决上面提到的问题,决定在集成子系统层各类异构数据格式之间的转换采用 Excel 为其中间格式;组件系统层整合数据采用程序模拟 DTS 的解决方案实现向全局数据库系统层数据格式转换的策略。

3.1　数据库接口实现

假设图 1 中的全局数据库系统下层分布两个异构组件系统 CS1 和 CS2,其数据库分别使用 MySQL 和 IBM DB2,并且在集成子系统层分别提供二者的数据库接口。我们采用 JDBC 底层应用程序编程接口(API),在不同数据库功能模块的层次上提供一个统一的用户界面。

CS1 对应数据库接口主要代码如下:

```
System. setProperty( "jdbc. drivers",  " com. mysql. jdbc. Driver");
Class. forName( "com. mysql. jdbc. Driver") . new Instance( );
DriverManager. registerDriver( new com. mysql. jdbc. Driver ( ));
```

CS2 对应数据库接口主要代码如下:

```
System. setProperty( "jdbc. drivers",
  "com. presistentjava. JdbcDriver");
Class. forName( "com. presistentjava. JdbcDriver") . new Instance( );
DriverManager. registerDriver
  ( new com. presistentjava. JdbcDriver( ));
```

连接数据库时,在 DriverManger 对象上调用 getConnection ( )方法。

3.2　Excel 中间件实现

在数据交换过程中,目标数据往往是源数据中的部分数据,这些数据有可能是一张表中的几个字段,有可能是一张表中某些记录值,也有可能是多张表的复杂关联查询的结果。但无论如何,将采用 Excel 文件作为转换数据中间存储的媒介,把数据有序地存放在 Excel 文件中,使得数据双方都理解这一中间的数据媒介。Jakarta POI 就是一种和 Java 技术相结合用来编辑 Excel 文件的开源工具。HSSF 为 POI 提供了 Excel 格式文件的真正的 Java 执行。其步骤如下:

1)调用源数据库存储过程(或执行一条子查询 SQL 语句),选择性地提取出关键数据。

2)将提取出的数据导入. xls 文件。

Excel 中的 Excel 文件、工作表、行、单元格对应 POI 中的 workbook、sheet、row 和 cell。提取一个数据单元写入指定 Excel 文件特订单元格的主要代码如下:

```
ResultSet rs  =  stateMent. executeQuery( );
int value1 = rs. getInt(1);
POIFSFileSystem fs  =  new POIFSFileSystem
   ( new FileInputStream( < spreadsheet > ));
HSSFWorkbook wb  =  new HSSFWorkbook( fs);
HSSFSheet sheet  =  wb. getSheetAt(0);                    //工作表
HSSFRow row  =  sheet. getRow(4);                          //行
HSSFCell cell  =  row. getCell( ( short) 2);               //列
cell. setCellType( cell. CELL_TYPE_NUMERIC);
cell. setCellValue( value1);
…
FileOutputStream fos  =  new FileOutputStream ( "");
wb. write( fos);
fos. flush( );
fos. close( );
```

3.3　DTS 的实现

除了用 Excel 中间件来完成集成子系统层的数据转换服

务外,本方案还提供了通过 DTS 的方法实现图 1 中组件系统与全局数据库系统异构数据间的整合。

3.3.1 ActiveX 脚本设计

设计 ActiveX 脚本是为了指定数据库源文件的存储路径,供远程系统自动调用数据源。在设计中,我们选择了 MySQL[7] 和 IBM DB2[8] 进行了实验,其数据源调用步骤如下:

首先,在每一个组件系统的源数据库中分别设置实例数据库的路径,以文件夹形式保存。其次,在本地系统中再指定一个不同路径,用以保存实例数据库的副本,其作用是源数据库比对副本数据库,如果存在新增表文件,则开始进行数据转换服务。然后,检验是否有新增表文件存在(此过程在集成子系统层实现)。最后,将新增表文件单独保存在用来执行数据转换服务的新文件夹。

以 MySQL 为例,其数据转换服务过程设计如图 3 所示。

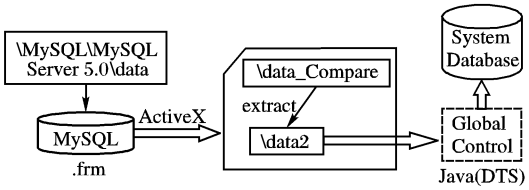

图 3 DTS 过程设计

ActiveX 脚本(JavaScript)主要代码如下:

```
Private Class FilesExtract
var dts, f1, strFullPath1, strFullPath2, strFullPath3,
  groupFileName1[ ], groupFileName2[ ];
dts = new ActiveXObject( "Scripting. FileSystemObject");
strFullPath1 = dts. GetFolder
 ("C: \ProgramFiles\MySQL\MySQL Server 5. 0\data");
strFullPath2 = dts. GetFolder( "D: \data_Compare");
strFullPath3 = dts. GetFolder( "D: \data2");
groupFileName1[ ] = dts. GetFolderFileName( strFullPath1);
groupFileName2[ ] = dts. GetFolderFileName( strFullPath2);
f1 = dts. GetFile( strFullPath1);
For ( var i = 0; i < groupFileName1[ ]. length( ); i ++)
  If not groupFileName1[ i]. exists( strFullPath2 &
    groupFileName2[ i]) then
    f1. Copy( groupFileName1[ i], strFullPath2);
  f1. Copy( groupFileName1[ i], strFullPath3);
Private function GetFolderFileName( FullPath)
var f, fc, s = "";
f = dts. GetFolder( FullPath);
fc = new Enumerator( f. files);                    //文件
For( ; !fc. atEnd( ); fc. moveNext( ))
  s += fc. item( ); s += ",";
return( s);
```

3.3.2 定位设计

设计定位程序为了让全局数据库系统了解数据转化服务的执行对象,远程系统数据库需要通过图 3 中的全局控制器得到本地数据库的新增表文件,并将其转换成系统数据库数据格式,导入系统数据库中。

Java 中,对文件夹的属性、方法的定义由类 File 提供。在进行数据转换之前,先用 File 类得到新增表文件(由于 SQL 用到的表名无需后缀,为正确地转化数据,须执行去后缀处理)。主要代码如下:

```
String newFiles = new File( "D: \data2");
String[ ] newFilesList = newFiles. list( );
for( int i = 0; i < newFileList. length( ); i ++)
  String newTableName = newFilesList[ i]. getName( );
  String name = newTableName. subString
    (1, newFilesList[ i]. LastIndexOf( ". frm"));      //去后缀处理
```

以上过程我们完成了数据库源文件的提取及系统远程调用的必要处理,接下来将对这些数据进行转换设计以完成 DTS 的功能实现。

3.3.3 DTS 设计

对于新增表文件要提取出其元数据,即要提取出“字段数”、“字段名”、“数据类型”、“数据长度”、“是否为空”等关键信息。一旦获得新增表文件的元数据后,就自动创建对应的系统数据库表。以 MySQL 向 MS SQL Server 导入元数据为例。

为了满足数据作转换后的完整性,首先要考虑两种数据库数据类型的对应关系。对于某些 MySQL 数据类型, MS SQL Server 中对应不止一种的数据类型。这里只对 MySQL 中的 DOUBLE、TIME、VARCHAR(m)这三种常用数据类型作描述。假设需要作数据转化的新增表文件中只存在以上这三种类型的数据。

主要代码如下:

```
ResultSet rs = stateMent. executeQuery( "select * from" + name);
ResultSetMetaData rsmd = rs. getMetaData( );
int numberOfColumns = rsmd. getColumnCount( );
boolean b = rsmd. isSearchable( 1);
String createTableSQL = "create table " + name + "(";
for ( int i = 1; i < = numberOfColumns; i ++)
  int j = rsmd. isNullable( i);
  if ( j = 0) String null = "";
  Else if( j > 0) null = "NOT NULL";
  if ( rsmd. getColumnTypeName( i) ! = TIME)
    if ( rsmd. getColumnTypeName( i) ! = DOUBLE)
      createTableSQL = createTableSQL + rsmd. getColumnLable( i) +
        " " + rsmd. getColumnTypeName( i) + "(" +
        rsmd. getColumnDisplaySize( i) + ") " + null + ",";
        //VARCHAR( m) 类型数据转换为 VARCHAR
    else createTableSQL = createTableSQL +
      remd. getColumnLable( i) + " " + "FLOAT" + null + ",";
      //DOUBLE 类型数据转换为 FLOAT
  else createTableSQL = createTableSQL +
    rem - d. getColumnLable( i) + " " + " SMALLDATETIME" +
    null + ",";      // TIME 类型数据转换为 SMALLDATETIME
stateMent. execute( createTableSQL);
```

这样,就完成了异构数据库数据格式间的完整转化,并将其导入到系统数据库。在用户看来整个体系结构就如同一个黑匣子,数据访问具有较强的灵活性和可靠性。

4 结语

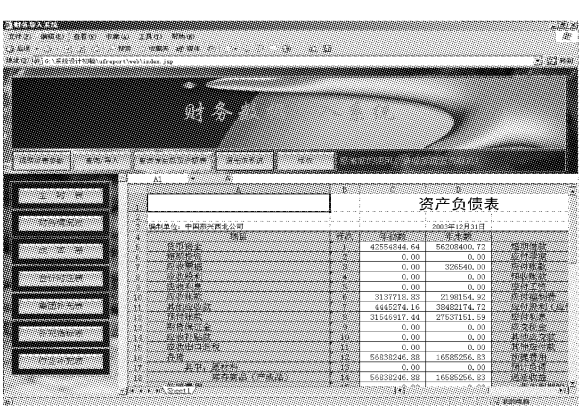

图 4 执行界面

(下转第 579 页)

以得到类似的结果。

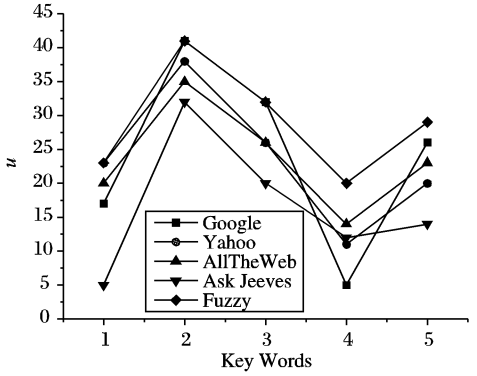

图 2　模糊积分和搜索引擎的比较

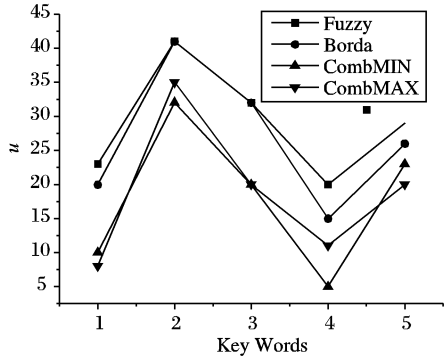

图 3　模糊积分和选举法及 MIN、MAX 法的比较

对于同样的查询和数据,将 Mysearch 和 Borda Count 及 CombMIN(CombMAX)元搜索模型结果做了对比,如图 3 所示。

从图 3 中看到,Borda 方法和 MIN 方法对文档的排序和模糊积分之间有着较大的差异。但 Borda 方法和模糊方法两者的 u 值计算却相差不大,模糊积分略好于 Borda 法,说明了这两种方法虽然对前 50 篇相关的文档的具体排序有着一定的差异,但对"哪些文档是最相关的 50 篇文档"的看法却是基本一致的。同时也反映出由于 CombMIN(MAX)算法较为简单,尤其是没有权重的考虑,因此在排序结果在 u 值上和模糊积分有较大的差距。模糊积分相对于 CombMIN(MAX)在文档融合排序上给出了更优的结果。

5　结语

元搜索引擎在覆盖率和查准率上均高于Web搜索引擎。如何能够提高查询效率及更好地组织从不同搜索引擎返回的结果,是一个好的元搜索引擎主要要解决的问题。在提高效率方面,本系统通过 Web Agent 的设计,同时对多个搜索源进行自主的查询,在有新的搜索源被发现的时候,只需加入一个新的对应的 Web Agent 即可,这样提高了系统的灵活性和并行性。同时,为了避免选择所有的搜索引擎耗费过多的时间,采用决策树和遗传算法相结合的思想,只调度最佳的搜索引擎,进一步提高了查询效率。用决策树和遗传算法决定哪些是最佳的搜索引擎,为避免陷入总是调度同样的搜索引擎,遗传算法的变异,加入了不常被调度的搜索引擎。在最终的结果表现上,使用模糊积分融合的方法对所有返回结果进行了重新排序。取得较好结果的原因是模糊积分是一个单调函数,通过合并各信息源的模糊度量值得到总的评价。模糊积分在融合时不简单地忽略较为"弱小"的声音,而是将这些建议累加参与最后的评价。也就是说,如果一篇文档被一个权重很高的搜索引擎评价为重要,那么这篇文档很可能重要,如果一篇文档被多个权重较低的搜索引擎评价为重要,那么这篇文档也很可能重要。

参考文献:

[1] LAWRENCE S, GILES CL. Inquirus, the NECI meta search engine[J]. Computer networks and ISDN systems, 1998, 30(1-7): 95-105.

[2] POWELL AL, FRENCH JC. Comparing the performance of collection selection algorithms[J]. ACM Transactions on Information Systems, 2003, 21(4): 412-456.

[3] LIN SD, KNOBLOCK C. Exploiting a search engine to develop more flexible web agent[A]. Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence[C]. CA, USA: IEEE Computer Society Press, 2003. 54-60.

[4] SUGENO M. Fuzzy mesures and fuzzy integrals: A survey[A]. GUPTA MM, SARIDIS GN, GAINES BR, ed. Fuzzy Automata and Decision Processes[C]. Amsterdam: North-Holland, 1977. 89-102.

[5] CUI SN, FENG BQ. A fuzzy integral method to merge search engine results on web[A]. HAO Y, ed. Computational Intelligence and Security, LNCS 3802[C]. Xi'an, China, 2005, PartII. 731-736.

[6] LEWIS DD. The TREC-4 iiltering track[A]. HARMAN D, ed. The Third Text Retrieual Conference (TREC-4)[C]. Washington DC, US: Department of Commerce, 1996. 165-180.

(上接第 576 页)

该方法应用在某小型企业财务信息集成项目中,满足了企业的需求,得到了用户的肯定。底层各子系统异构数据的整合通过 Excel 及 DTS 实现;财务部在全局数据库系统层按业务流收集整合数据并导入 Excel,实现跨应用的报表格式转换。执行界面如图 4 所示。

现今,越来越多的 SOA 开发商将异构数据库间的研究放在了一定高度,尽量减少了元数据间的差异;开发工具的研究也加入了几乎全部的 SQL 数据类型。这就对 DTS 提供了便利的开发环境。但并不是说 Excel 在数据转换方面的作用开始淡化,Excel 是现今最强大的报表数据中间件。实践证明,本方法特别适合于网络架构层次不是很复杂的中小企业的异构数据集成。

参考文献:

[1] 陈华智,郑宁,葛瀛龙. 数据交换的一种设计方案与应用实现[J]. 计算机工程与应用,2005,41(11):186-189.

[2] 章立民. SQL Server 2000 完全实战——数据转换服务(DTS)[M]. 北京:中国铁道出版社, 2002.

[3] MARTIN SO. Database and Application Security[M]. Kluwer Academic Publishers, 2002. 61-182.

[4] BEA. BEA WHITE PAPER - IT TRANSFORMATION TO SERVICE-ORIENTED ARCHITECTURE[EB/OL]. http://www.bea.com, 2006-07-24.

[5] 余腊生, 李徐. 基于 Web 服务的跨网络异构数据交换技术[J]. 计算机应用, 2005,25(z1):9-11.

[6] MUTH P. Application Specific Transaction Management in Multidatabase Systems[J]. Distributed and Parallel Databases, 1997,5(4):357-403.

[7] VASWANI V. MySQL 完全手册[M]. 北京:电子工业出版社, 2004. 20-82.

[8] 杨健, 李育龙. IBM DB2 应用开发指南[M]. 北京:电子工业出版社, 2004.279-302.