

文章编号: 1006-4729(2007)01-0041-04

电子政务数据集成系统中数据模式的研究^{*}

邓莎莎¹, 杜明干²

(1. 上海电力学院 计算机与信息工程学院, 上海 200090

2. 柏克德(中国)工程有限公司, 上海 200050)

摘要: 分析了数据模式提取的基本原则和电子政务集成系统对数据模式的管理. 研究了结构化、半结构化以及网页数据的 XML 数据模型, 该模型适合异构数据之间转换.

关键词: 数据模型; 电子政务; 数据集成系统

中图分类号: N37; TP274⁺. 2 文献标识码: A

Research on Data Model of Data Integration System in E-Government

DENG Sha-sha, DU Ming-gan

(1. School of Computer and Information Engineering, Shanghai University of Electric Power

Shanghai 200090 China; 2. Becht (China) Engineering & Construction Co Ltd, Shanghai 200050 China)

Abstract: Basic principles of extracting data model and management of data model for integration system in E-Government are presented. Study on XML data model of structured and half structured and Web data shows that this model is fit for transition of heterogeneous data.

Key words: data model; E-Government; data integration system

随着全球政治经济一体化的日益明显, 以电子政务为代表的政府管理服务职能的无纸化、自动化已在许多国家尤其是发达国家中迅速发展^[1]. 然而, 众多电子政务系统的涌现很容易造成局面的混乱, 给使用电子政务的公众带来极大的麻烦. 因此迫切需要一个能有效地将各种电子政务资源整合到一起的平台.

但是, 目前我国还没有一个关于电子政务数据集成平台. 这是因为这些数据往往是跨行政边界的, 数据结构多种多样, 隶属关系纷繁复杂, 要协调建立一个数据集成平台将会遇到各种各样的困难^[1, 2]. 如何建立一个共享共用的数据集成平台是目前需要迫切研究的课题.

一个数据集成系统的基本单位是数据模式^[2], 它通常表现为本地数据源的一个简单查询、单个实体的数据结构或多个紧密关联的实体集. 数据模式是源节点数据之间数据交互与共享的基础, 是源节点数据对集成系统提供的共享数据接口.

1 数据模式的提取

数据模式的选取应根据节点数据库共享数据的需求而定. 数据模型为某一个数据源的数据视图^[3]. 就结构化数据而言, 数据视图可以是一个查询、某一个实体的描述或者某一个实体间关系的描述等. 这些数据视图可以直接描述为数据模

式.半结构化数据和 Web数据则相对复杂一些,它是通过与用户交互产生一组正则表达式以完成数据模式的抽取.

2 数据模式的管理

每个源节点要向数据集成系统提供数据访问时,必须先将自己的数据模式发布出来,数据集成系统的其他节点才能根据发布的数据模式访问该节点的数据.

某一个源节点在发布和共享自己的数据模式的同时,还要针对该数据模式为数据集成系统提供数据访问接口.因此,发布出去的数据模式与该节点的数据访问接口是绑定的.

为了便于对数据模式和数据访问接口的统一管理,数据集成系统采用注册服务中心的方式管理已经发布共享的数据模式和数据访问接口.注册中心存储了各个节点所发布的共享数据模式,并按照提供者和提供类型进行分类,形成一个松散的全局数据模式.注册服务中心可以配置数据模式的访问权限,为参与集成的每个源节点提供数据模式和数据访问接口的查询及修改服务.

3 数据模型的 XML表示

可扩展标记语言 XML(Extensible Markup Language)是一种面向 Internet应用的标记语言,它是由 W3C组织于 1998年 2月制定的一种通用语言规范.XML具备强大数据描述能力,将数据格式与数据内容巧妙地结合在一起,便于机器对 XML数据的理解^[4,5].XML能够很好地处理异构系统语义歧义的问题,异构数据源间可以利用 XML文档转换技术进行有效的数据交换.正是由于 XML本身具有的这些特征,使其成为集成系统中数据模型的重要表示方法.

3.1 结构化数据模式的 XML表示

一般所集成的结构化数据大多采用关系数据模型存储数据.这是因为关系数据库理论体系较为完善,访问性能好,数据冗余小,一致性好,数据间联系紧密,适合于本地数据的存储.因此需要用 XML有效地表示关系数据库中的关系型数据,让其既尽可能保持 XML数据的关系模型,又方便 XML数据与关系数据库中数据的相互转换.

XML文档是一种树状、层次型的表达结构,

每个元素都可以被多个属性和子元素来修饰和扩充.该特性与信息世界所抽象出的实体集、实体、属性和键的层次结构特点相对应.同时,XML还有数据格式的描述标准 XML DTD和 XML Schema,以限制数据在 XML文档中的表现形式.以 XML DTD方式对数据模式的表示如下:

```
<MODEL ID="Model">
  <ENTITY ID="法人">
    <FIELD ID="公司编号" TYPE=
      "number" LENGTH="4" NULABLE
      ="NO"/>
    <FIELD ID="姓名" TYPE=
      "string" LENGTH="10" NULABLE
      ="NO"/>
    <FIELD ID="年龄" TYPE=
      "number" LENGTH="2" NULABLE
      ="NO"/>
    <FIELD ID="性别" TYPE=
      "string" VALUE="男 女" LENGTH
      ="1"/>
    <PRIMARYKEY>
      <IFDID ID="公司编号"/>
    </PRIMARYKEY>
  </ENTITY>
  <ENTITY ID="税务">
    <FIELD ID="税收项目编号" TYPE=
      "number" LENGTH="8" NULABLE
      ="NO"/>
    <FIELD ID="税收项目名称" TYPE=
      "string" LENGTH="10" NULABLE
      ="NO"/>
    <PRIMARYKEY>
      <IFDID ID="税收项目编号"/>
    </PRIMARYKEY>
  </ENTITY>
  <ENTITY ID="公司税收">
    <FIELD ID="公司编号" TYPE=
      "number" LENGTH="4" NULABLE
      ="NO"/>
    <FIELD ID="税收项目编号" TYPE=
      "string" LENGTH="8" NULABLE=
      "NO"/>
    <FIELD ID="税收收入" TYPE=
      "number" LENGTH="20"
      NULABLE="NO"/>
    <PRIMARYKEY>
      <IFDID ID="公司编号"/>
      <IFDID ID="税收项目编号"/>
    </PRIMARYKEY>
  </ENTITY>
```

```

< FOREIGNKEY ID= " FRkey1" FROM _
  ENTITY=
    "公司税收" TO ENTITY="法人">
  < lFORM FIELD="公司编号"
    TO FIELD="公司编号" />
< /FOREIGNKEY>
< FOREIGNKEY ID= " FRkey2" FROM _
  ENTITY="公司税收" TO ENTITY="税
  务">
  < lFORM FIELD="税收项目编号"
    TO FIELD="税收项目编号" />
< /FOREIGNKEY>
< /MODEL>

```

由此可见, 该定义巧妙地利用了 XML DTD 的特点, 将数据模式单独放在 DTD 中定义, 将数据模式和数据内容分开. 这样, 既清晰地表达了数据模式, 又在很大程度上降低了数据内容部分的冗余信息.

同时, 数据模式与数据内容又在同一个 XML 文档中, 采用 DTD 的内容使用方式, 保证了 XML 数据在互联网上交流时数据、语义和结构信息的共存. 在 DTD 的约束下, XML 中的数据在被处理过程中会符合最初的信息模式.

3.2 半结构化数据模式的 XML 表示

在电子政务数据集成系统中, 半结构化数据主要是异构的文件系统, 其数据 and 应用程序紧密相连. 异构性主要表现在环境的异构性、数据结构的异构和数据操作的异构等方面, 且文件系统不具备传统数据库系统的一些特点: 模式定义、查询语句、事务的 ACID 特征等.

当有新的文件系统作为数据源加入到系统中时, 需要向服务中心申请注册. 经过注册后的数据源才可以为系统所用. 系统需要访问文件系统数据时, 不同类型的文件系统将会调用不同的数据访问组件. 下面列出的是半结构化数据的元数据模型, 它记录了文件系统数据源的特征.

```

< MODEL ID= " Model1">
  < ENTITY ID= " ExceRegister">
    < FIELD ID= " DBNum" TYPE=
      " number LENGTH= " 4" NULABLE=
      " NO" />
    < FIELD ID= " FilePath" TYPE=
      " string LENGTH= " 100" NULABLE=
      " NO" />
    < FIELD ID= " SheetName" TYPE=

```

```

      " string" LENGTH= " 20" NULABLE=
      " NO" />
    < FIELD ID= " TargetTableName" TYPE=
      " string" LENGTH= " 20" NULABLE=
      " NO" />
    < FIELD ID= " TargetHeadName" TYPE=
      " number LENGTH= " 4" NULABLE=
      " NO" />
    < FIELD ID= " isColumnName" TYPE=
      " number LENGTH= " 1" NULABLE=
      " NO" />
    < PRIMARYKEY>
      < lFIELD ID= " TargetTableName" />
    < /PRIMARYKEY>
  < /ENTITY>
  < ENTITY ID= " TxRegister">
    < FIELD ID= " FilePath" TYPE=
      " string LENGTH= " 100" NULABLE=
      " NO" />
    < FIELD ID= " Separator" TYPE=
      " string LENGTH= " 3" NULABLE=
      " NO" />
    < FIELD ID= " SheetName" TYPE=
      " string LENGTH= " 20" NULABLE=
      " NO" />
    < FIELD ID= " TargetTableName" TYPE=
      " string" LENGTH= " 20" NULABLE=
      " NO" />
    < FIELD ID= " TargetHeadName" TYPE=
      " number LENGTH= " 4" NULABLE=
      " NO" />
    < FIELD ID= " isColumnName" TYPE=
      " number LENGTH= " 1" NULABLE=
      " NO" />
    < FIELD ID= " Type1" TYPE=
      " string LENGTH= " 20" NULABLE=
      " NO" />
    ...
    < FIELD ID= " Type n" TYPE=
      " string LENGTH= " 20" NULABLE=
      " NO" />
    < PRIMARYKEY>
      < lFIELD ID= " FilePath" />
    < /PRIMARYKEY>
  < /ENTITY>
< /MODEL>

```

用户完成文件系统数据源登记后, 系统将根据元数据模式定义存储相关数据. ExceRegister 中主键为 TargetTableName. 由于 Exce 文件中会有多个 Sheet, 所以在模块中定义, 一个 sheet 对应

一张表, 而一个 Excel文件只对应一个 ID号. SheetName是在程序读取源文件的 Sheet名称时临时生成的. 由于模块对 Excel文件的操作是以单元格为单位的, 因此必须知道文件数据的起始单元数.

同样, 文本文件的信息存储在 textRegister中, 主键为 filePath它是文本文件的存储路径, 具有唯一性. Separator是根据用户提供的信息填入的. 由于一个文本文件中会有多个分隔符, 分隔符和分隔符之间用“!”分开. Type1 ~ TypeN也由用户提供, 由于文本文件是半结构化文件, 因此需要用户提供每个数据项的详细信息.

3.3 网页数据模式的 XML表示

政府机构很多统计数据都在其网站中以表格的形式公布. 因此, 本系统对于网页数据的处理着重于对网页中表格数据的处理. 网页中表格多种多样, 主要有以下 4种形式: 一维横向表格、一维纵向表格、二维横向表格、二维纵向表格. 由于网页数据的不规则性, 因此在获取网页数据模式时需要人机交互, 这样才能有效避免垃圾数据和无用数据. 网页数据模式的表示形式如下:

```
< MODEL ID= "Model3">
  < ENTITY ID= "InternetData">
    < FIELD ID= "IPaddress" TYPE=
      " string" LENGTH= "400" NULABLE=
      " NO" >
    < FIELD ID= "TableFormat" TYPE=
      " string" LENGTH= "100" NULABLE=
      " NO" >
    < FIELD ID= "TableName" TYPE=
      " string" LENGTH= "50" NULABLE=
      " NO" >
    < PRIMARYKEY>
      < IFDID ID= "TableName" >
    < /PRIMARYKEY>
  < /ENTITY>
< /MODEL>
```

该模型中的 IPaddress为所需数据所在的网

址, 通过该网址可以获取网页数据, 同时可以在网页中人工选取所需数据所在的表格. TableFormat是由人工选择该表格的类型. 针对不同的类型, 系统的处理方式各不相同. TableName指的是在系统中心数据库中保存的表名. 在该系统中表名是唯一的. 同一个网址可以有多个所需的表格数据, 通过人工交互最终保存在中心数据库中可以是多个或者一个数据表.

4 结束语

数据模式是电子政务数据集成的基本单元. 数据模式定义了数据的格式、数据的约束、子数据项之间的关系, 以及数据的操作方式等, 为程序自动处理基于该数据模式的数据提供了参考. 数据模式的选取对集成系统影响很大, 直接关系到数据的基本操作. 提取的数据模式需要按照一定的方式管理, 便于集成系统的检索和使用. 一般而言, 数据类型有结构化数据和半结构化数据, 采用 XML数据模式进行数据交换, 快捷而又方便. 数据模型的异构性比较复杂, 本文只从简单的异构性分析入手, 逐步解决多种类型的数据模式异构性.

参考文献:

[1] 王长胜. 电子政务蓝皮书: 中国电子政务发展报告[M]. 北京: 社会科学文献出版社, 2003 1-137

[2] Glassy Q. Developing a One-step Government Data Model[J]. Government Information Quarterly, 2004 21: 156 ~ 169.

[3] 陈跃国, 王京春. 基于 XML的关系数据模型研究[J]. 中南大学学报, 2003 34(2): 307-310

[4] Christophides V, Chiet S, Simeon J. On Wrapping Query Languages and Efficient XML Integration[C]. Proceedings of the ACM SIGMOD Conference on Management of Data, Dallas, USA, 1998 141-152

[5] Chamberlin D, Robje J, Florescu D, Quilc. An XML Language for Heterogeneous Data Sources[C]. Proceedings of the International Workshop on the Web and Databases (WebDB), Dallas, USA, 2000 53-62