# 基于 XML 的虚拟法异构数据集成方法研究 *

袁景凌，徐丽丽，苗连超
（武汉理工大学 计算机科学与技术学院，武汉 430070）

**摘　要**：随着信息技术的发展，Web 上的数据日趋成为当今数据的主流，但是 Web 上的数据多是异构的，而越来越多的人需要访问各种异构数据，为了满足这种需求，必须有一种系统能够支持异构数据集成。异构数据集成的方法有很多，给出一个基于 XML 的虚拟法异构数据集成系统体系结构，引入虚拟法，提出用虚拟法进行异构数据集成；最后引入数据清洗技术，能够较好地解决异构数据的集成。

**关键词**：虚拟法；包装器；查询引擎；数据清洗
**中图分类号**：TP311.13　　**文献标志码**：A　　**文章编号**：1001-3695(2009)01-0172-03

# Research on virtual approach about heterogeneous data integration based on XML

YUAN Jing-ling, XU Li-li, MIAO Lian-chao
(*School of Computer Science & Technology*, *Wuhan University of Technology*, *Wuhan* 430070, *China*)

**Abstract**: With the development of information technology, Web-based data becomes the mainstream of today's data, but data on Web are mostly heterogeneous. More and more people need to access all kinds of heterogeneous data, so there must be a kind of system which can support heterogeneous data integration. There are many kinds of methods that can carry out heterogeneous data integration, but this paper introduced a kind of architecture of the virtual approach for heterogeneous data integration based on XML in order to solve the problem of heterogeneous data integration on Web. It proposed to use the virtual approach to carry out heterogeneous data integration, and finally introduced the technology of data cleaning which could solve the integration of heterogeneous data better.

**Key words**: virtual approach; wrapper; search engine; data cleaning

## 0　引言

随着计算机技术，特别是 Internet 的迅速发展，可共享的资源越来越多，遍布于 Internet 中的 Web 服务器将各种异构数据集成在一起，形成一个全球性的信息共享环境，成为人们获取信息的主要渠道。Web 上的数据是数据集成系统中最大的数据来源，具有全面、及时、多样等特点。

异构数据不仅是指数据库之间的异构，而且还包括不同结构数据之间的异构性。随着信息化建设的不断推进，在选择数据处理方式时存在着时间和空间上的差异，使得对数据的描述千差万别，数据环境的差异也越来越大，主要表现在以下几个方面：

a) 系统异构，即数据源所依赖的业务应用系统、数据库管理系统乃至操作系统间的不同构成了系统异构。

b) 模式异构，即数据源在存储模式上的不同。存储模式包括关系模式、对象模式、对象关系模式和文档嵌套模式等几种，其中关系模式为主流存储模式。

c) 数据异构，不同数据源具有不同的数据类型，包括结构化数据（数据库）、半结构化数据（HTML、XML）、非结构化数据（如文本、图片）。

d) 语义异构，表现在相同的数据形式表示不同的语义或同一语义由不同形式的数据表示[1]。

异构数据集成是将互相关联的异构数据源集成到一起，使用户能以透明的方式访问这些数据源，并能够低代价、高效率地使用这些异构数据。如何对各种异构数据源中的数据进行有效集成，给用户提供一个统一和透明的访问界面，同时还要保持局部系统的自治性是异构数据集成研究的主要问题。

## 1　系统设计方案

目前，关于数据集成方案的研究很多，各不相同，但就其基本方法来说，大体上可归为如下两类：

a) 仓库法（warehousing approach）。建立一个数据仓库，将参加集成的各数据源的数据副本存储到数据仓库中；同时，系统将提供对该数据仓库的查询机制，此后所有的操作都直接在数据仓库上进行。这种方式的优点是数据仓库是集成的、面向主题的、随时间变化的持久的数据集合，无须访问数据源，因此访问效率高，既可用于数据集成，又可用于决策支持；缺点是数据重复存储，难以更新，比较适用于简单系统中。使用的技术主要涉及如何有效地加载数据、增量维护数据等。

b) 虚拟法（virtual approach）[2]。也称为中间件法（wrapper/mediator）。在这种方式下，数据仍然存储在各自独立的数据源，数据之间通过 wrapper 按全局模式进行转换，用户针对全局模式提出查询，由查询引擎对用户的全局查询进行查询解析、重写和分解；然后各数据源根据查询要求自行独立处理，由 mediator 把结果提取出来集成后返回给用户。它实际上是一种软件组件，支持虚拟数据库。该方式无须重复存储大量数

**收稿日期**：2008-03-25；**修回日期**：2008-06-28　　**基金项目**：国家教育部规划基金资助项目（07JA630028）

**作者简介**：袁景凌（1975-），女，湖北武汉人，副教授，博士，主要研究方向为智能方法、网络数据库等；徐丽丽，女，硕士研究生，主要研究方向为现代数据库技术、异构数据集成等（xulili1402043@126.com）；苗连超，男，硕士研究生，主要研究方向为网络存储技术、现代数据库技术等.

据,数据更新及时,比较适合于高度自治、集成数量多且更新变化快的数据集成系统,是目前主要的研究方向。虚拟法体系结构中使用的技术与传统数据服务器中的技术有着很大的不同。前者采用的技术更多地涉及到查询上的代数操作;后者采用的技术大部分集中在查询计算以及与存储有关的问题上(如数据的聚类、索引等)。

根据对以上两种集成方案的分析,综合考虑数据仓库和虚拟法方式的优缺点,结合目前状况和需求,本文采用虚拟法进行相应的设计开发。虚拟法架构的系统安全性更好,灵活度更高,系统性能更稳定,对系统资源的要求也更低,能够从根本上解决其他两种方式在实际应用中的不足,有较高的实用价值。一方面,集成系统通过 XML 将汇集分在各个异构数据源的信息进行数据清洗、转换和加载;另一方面,用户通过 Internet 连接到集成系统中,进行查询操作。

集成系统包括三层:用户应用层、中介层和信息抽取层。用户层主要是用户提出查询;中介层包括查询引擎、数据清洗、中间数据库和 XML 文档以及 wrapper;信息抽取层包括各异构数据源。系统实现的基本框架如图 1 所示。

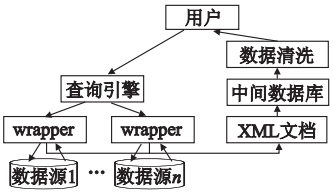


图1　异构数据集成系统框架图

## 2　系统实现关键技术

### 2.1　XML 的特点

XML 是一种半结构化的数据模型。其可扩展性、灵活性及自描述性等诸多特性,使得它可以描述不规则的数据,能够集成来自不同数据源的数据,可以将多个应用程序所生成的数据纳入同一个 XML 文件。因此,把 XML 作为集成系统中集成层的数据描述工具和转换工具是必然的选择。XML 具有内容的自描述性、跨平台性、内容与显示分离、可扩展性等特点,为数据访问和交换提供了一种新的模式,使描述各种格式的数据成为可能。基于 XML 的各种技术不断涌现,XML 将成为一种新的数据互操作的标准。

### 2.2　中介层模型

#### 2.2.1　查询引擎

查询引擎模块主要包含:

a)查询解析。用户提出的查询请求是 Xquery 形式,对用户提出的请求进行简单的语法分析,把不合理的查询进行修改或直接返回给用户,把初步处理过的查询进行重写。

b)查询重写。给定数据库 D 和在数据库 D 上定义的视图集合 $V\{V_1,V_2,\cdots,V_n\}$。对数据库 D 的查询 $q$,如果存在查询 $q_1$,其中 $q$ 至少查询了视图集合 $V$ 中一个视图,而且 $q_1$ 的查询结果与 $q$ 在数据库中的查询结果一致,则称 $q_1$ 是 $q$ 的查询重写。如果 $q_1$ 只对视图集合 $V$ 中的视图进行查询,则称上述查询重写 $q_1$ 为 $q$ 的完全重写。

c)查询分解。对重写后的查询进行分解,分解为对各个数据源的查询请求。

#### 2.2.2　Wrapper 中的数据映射

包装器主要完成三个功能:a)解析中介器提交过来的子查询请求;b)根据请求内容对数据进行读取和查询;c)将查询结果转换成统一的数据格式,返回给中介器。Wrapper 组件由两部分组成,即 wrapper 生成器和 wrapper 实例。每个 wrapper 都要为某个特定的数据源制定相应的接口,这个功能由 wrapper 生成器完成。各个数据源的 schema 通过相应的 wrapper 溶入到全局 schema。这个过程中,wrapper 将数据源中的数据转换为一个 DOM 对象[3]。

例如,有 student 和 course 两个关系表,它们与 XML DTD 之间的映射如下:

```
Create table student
(StuNo char(7) not null,stuName char(10),Gender char(2))
Create table course
(CNo char(5) not null,CName char(10),CTeacher char(10))
  〈! ELEMENT Student(StuName,Gender)〉
  〈! ATTLIST StuNo IDREF #REQUIRED〉
  〈! ELEMENT StuName(#PCDATA)〉
  〈! ELEMENT Gender(#PCDATA)〉
  〈! ELEMENT Course(CName,CTeacher)〉
  〈! ATTLIST CNo IDREF #REQUIRED〉
  〈! ELEMENT CName(#PCDATA)〉
  〈! ELEMENT CTeacher(#PCDATA)〉
```

#### 2.2.3　XML 与中间数据库的映射

XML 与数据库之间的映射可以分为两种:模板驱动和模型驱动。

基于模板驱动没有预先定义文档结构与数据库结构之间的映射关系,只能应用于关系数据库与 XML 文档之间传递数据,使用时将带参数的 SQL 命令内嵌于模板中,并用中间件等软件来进行处理。采用这种方法进行集成不需要一个明确的数据模式。

模板驱动的映射优点是可以相当灵活;缺点是目前只支持将关系数据库转换为 XML 文档。

基于模型驱动的转换把数据从数据库传送到 XML 文档,或从 XML 文档传送到数据库时不仅仅依赖于内嵌的 SQL 命令,要用一个具体的模型。需要构建一个统一的描述方法来描述数据,通过这种描述来访问全局信息系统,并采用合适的访问机制来访问数据源或者数据视图。

模型驱动的优点是简单易用;缺点是灵活性不够。

下面是 Java 编写的用 DOM API 解析上面的 XML 文档:

```
public class DomParse
{
  public class DomParse()
  {
    DocumentBuilderFactory df = DocumentBuilderFactory. newIn-
stance();
    try
    {
    DocumentBuilder db = df. newDocumentBuilder();
    Document doc = db. parse(new File("student. xml"));
    Element root = doc. getDocumentElement();
    NodeList students = root. getChildNodes();
    if(students! = NULL)
      {for cint i = 0; i < students. length; i + +
    {
        Node student = students. getChildNodes(). item(i);
        System. out. println(student. getNodeValue());
      } //end for
      } //end if
    } //end try
  catch(ParserConfigurationException e)
```

```
    {
      e. printstacktrace( ) ;
      }
      catch( IOEception e)
      {
        e. printstacktrace( ) ;
      }
      }
      public static void main( String args[ ] )
      {
      new DomParse( ) ;
    }
  }
```

#### 2.2.4　对中间数据库的数据清洗

数据质量是评价一个数据集成系统优劣的关键。面对现阶段巨大而零乱的数据,人工处理非常困难,数据质量问题因此成为制约数据应用的瓶颈之一。数据清洗是解决数据质量问题的一个关键步骤,是保证和提高数据质量的关键,重要性是不言而喻的。

数据清洗(data cleaning)是一个减少错误和不一致性,解决对象识别的过程。随着计算机科学技术的发展与普及,以及人们对信息的需求越来越广泛,数据的共享已经不局限于一个部门内数据库的相互访问,还涉及到部门间的数据共享。对于实际运行的系统来说,很多因素都可能产生数据库系统间的差异性,因此在异构数据集成中存在大量的“脏数据”。所谓“脏数据”是指数据不在给定的范围内或对于实际业务毫无意义,或是数据格式非法以及错误操作产生等。这些“脏数据”会扭曲事实,影响信息系统的运行效果。

Web 异构数据源中的数据提取工作的完成并不意味着结束,如果仅仅是将这些信息抽取出来不加以利用,则这些信息将毫无价值。异构数据集成系统中数据的清洗一般分为七个步骤:元素化(将非标准的数据统一格式化成数据元素)、标准化(将元素标准化)、校验(对标准化的元素进行校验分析,即在内容、规则上验证字段值的正确性,若发现错误则加以改正)、匹配(在其他记录中寻找相似的记录,发现重复异常)、消除重复记录(根据匹配结果进行处理,删除部分记录或把多个记录合并为一个信息更完整的记录)、去除无效记录(依据庞大的参考数据库或者通过电话访问核实,找出无效的记录或将这些记录更正补全)、档案化、系统化(将结果写入新的数据库中,这样既可使用清理过的数据,又可保留对原数据的操作记录,为今后的进一步分析或使用留下必要的记录信息)。

### 2.3　信息抽取层

信息抽取层位于系统的底层,是系统数据提供者,主要功能是抽取和集成分布在多个异构数据源上的信息。Web 上的数据经过抽取并打包到一个 XML 文档,最后将该文档返回给中介层。

## 3　集成系统的特点分析

本文提出的三层架构的虚拟法异构数据集成系统具有以下几个特点:

a)数据仍然存储在各自独立的数据源,各数据源根据查询要求自行独立处理。因此,数据更新及时,避免了数据大量重复存储。

b)用户针对全局模式提出查询,由查询引擎对用户的全局查询进行查询解析、重写和分解。

c)采用数据清理机制消除数据的不一致性、重复性,使得从 Web 数据源中抽取出的信息可以更有效地集成到此系统中,实现信息的再利用。

d)采用 XML 到中间数据库的映射机制,完成数据库与 XML 文档之间的数据传递。

e)利用 wrapper 技术抽取和集成分布在多个异构数据源上的信息,并将文档返回。

## 4　结束语

企业内部或企业之间的分布式异构系统的数据交互和集成对于提高企业办事效率、节省成本都有着重要的意义。数据集成是企业业务集成和门户集成的基础。本文从实际出发,目的旨在解决 Web 上的异构数据集成问题,在所设计的系统中利用了虚拟法结合时下比较流行的 XML 技术、数据清洗技术,从而很方便地实现了异构数据集成问题。

**参考文献:**


[1] 景娅萍. 基于 Web 异构数据源的整合与集成[J]. 贵州教育学院学报,2004,15(4):72-73,79.

[2] 洪小伟,刘亚军. 基于 XML 异构数据集成的研究[D]. 南京:东南大学,2004.

[3] 李军怀,周明全,耿国华,等. XML 在异构数据集成中的应用研究[J]. 计算机应用,2002,22(9):10-12.

[4] ALBERT J. Data integration in the RODIN multidatabase system [C]//Proc of the 1st International Conference on Cooperative Information Systems. 1996:48-57.

[5] 刘华,胡燕. Web 信息集成中数据清洗的研究[D]. 武汉:武汉理工大学,2007.

[6] 顾天竺,沈洁,陈晓红,等. 基于 XML 的异构数据集成模式的研究[J]. 计算机应用研究,2007,24(4):94-96.


(上接第 142 页)

## 6　结束语

本文提出的针对安全苛求软件的测试用例生成策略,应用安全覆盖准则自动生成测试用例。它是一种自动黑盒测试方法,它使从测试规格说明产生测试用例的过程自动化,减少了软件测试的复杂度。实验表明,它对于安全苛求软件具有更好的安全性保证。今后的研究工作将放在开发出一个高效、实用的测试用例自动生成工具上。

**参考文献:**


[1] ISAKSEN U, BOWEN J P, NISSANKE N. System and software safety in critical systems[EB/OL]. (1996)[2008-05-27]. http://citeseer. ist. psu. edu/isaksen96system. html.

[2] 年晓玲. 基于扩展有限状态机软件测试用例自动生成的研究[D]. 成都:西南交通大学, 2005.

[3] 占学德. 基于 UML statecharts 测试方法的研究[D]. 上海:上海大学,2005.

[4] JENSEN K. Colored Petri nets:basic concepts, analysis methods and practical use[M]. 2rd ed. Berlin, New York: Springer-Verlag, 1995.

[5] DICK J, FAIVRE A. Automating the generation and sequencing of test cases from model-based specification[C]//Proc of the 1st International Symposium of Formal Methods Europe on Industrial-Strength Formal Methods. London:Springer-Verlag, 1993:268-284.