

# 数据库中间件及其在 AES 中的应用研究

宋海生, 李 淼, 张 建

(中国科学院 合肥智能机械研究所, 安徽 合肥 230031)

**摘 要:**数据库是计算机应用的热点之一,在软件开发中利用数据库中间件技术可提高软件质量和软件开发效率。文中对当前两种主流的数据库中间件 ODBC 和 JDBC 技术规范作了深入的阐述,把该技术分别应用于农业专家系统与数据库的通讯接口中,知识库以统一的格式来调用不同的数据库,解决了以往调用不同数据库所带来的不一致性问题,扩大了农业专家系统的知识获取范围。并对这两种数据库中间件的异同做了比较。

**关键词:**数据库中间件;农业专家系统;Java 数据库连接;开放式数据库连接

**中图分类号:** TP311.138

**文献标识码:** A

**文章编号:** 1005-3751(2004)06-0052-03

## Research on Database Middleware and Its Application in AES

SONG Hai-sheng, LI Miao, ZHANG Jian

(Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China)

**Abstract:** Database technology is a hotspot in the field of computer applications. Using database middleware technology in software development can improve software quality and efficiency. This paper introduces two database middleware technology ODBC and JDBC specifications, and puts these technologies to use in communication interface between agriculture expert system(AES) and database respectively. Knowledge base calls different databases in the same way, it solves the problem of inconsistency and enlarges the scope of Knowledge acquisition. Finally makes a comparison between them.

**Key words:** database middleware; agriculture expert system; JDBC; ODBC

### 0 引 言

数据库是计算机应用中最活跃的领域之一,数据库技术的发展非常迅猛,从单层的数据库到 C/S 双层数据库再到新兴的三层乃至多层数据库结构,数据库技术日益成熟,数据库的应用也日益广泛。人们对数据库存取访问和管理不仅仅通过数据库开发商提供的各式各样的数据库管理系统(DBMS),而且也使用各种面向对象的可视化开发语言,这些开发语言通过数据库中间件和 API 接口函数来存取管理数据库,实现 DBMS 所能实现的功能<sup>[1]</sup>。在各种提供数据库连接的数据库中间件技术中,ODBC 和 JDBC 这两种技术应用最广泛,也最为成熟。农业专家系统(AES)是当前人工智能研究中最活跃的分支之一,它主要由农业知识库(包括规则库和事实库)、推理机、解释机、界面和接口组成。规则库中存放产生式规则的集合;事实库中存放事实的集合,包括输入的事实和中间结果、最终结果以及各种数据库表。推理是指按照某种规则从已知事实出发推导出中间结果和最终结果的过程,推理方法有

三种:正向推理、反向推理、正反向混合推理,推理机就是执行推理功能的程序。解释机负责向用户解释专家系统的行为,包括解释推理的正确性和系统输出其它候选的原因等。界面负责向用户显示专家咨询的最终结果。推理机、解释机与知识库之间的通讯则是接口通过数据库中间件来完成的。

### 1 数据库中间件技术概述

#### 1.1 ODBC 基本概念与组成原理

ODBC(Open DataBase Connectivity, 开放式数据库互连)是微软推出的符合 WOSA(Windows Open Services Architecture, 视窗开放服务体系结构)规范的工业标准,它提供了一组对数据库访问的标准应用程序编程接口(API)<sup>[2]</sup>。目前绝大多数数据库厂商、应用软件和工具软件厂商都为自己的产品提供了 ODBC 接口或提供了 ODBC 支持,这其中包括常用的 SQL SERVER, ORACLE, INFORMIX, SYBASE, FOXPRO 等。

ODBC 是应用程序和数据库系统之间的中间件,它通过使用相应平台上和所需数据库对应的驱动程序与应用程序的交互来实现对数据库的操作,避免了在应用程序中直接调用与数据库相关的操作,从而提供了数据库的独立性。一个基于 ODBC 的应用程序对数据库的操作不依赖

收稿日期:2003-10-20

基金项目:国家“八六三”高技术研究发展计划(2001AA115180)

作者简介:宋海生(1975—),男,江西会昌人,硕士研究生,研究方向为数据库与专家系统。

于任何 DBMS,不直接与 DBMS 打交道,所有的数据库操作由对应的 ODBC 驱动程序完成。也就是说,不论是 FOXPRO, ACCESS 还是 ORACLE 数据库,均可以用 ODBC API 进行访问,这也是 ODBC 一经推出就获得巨大成功的原因。

ODBC 以 X/Open 和 ISO/IEC 的调用级接口(CLI)规范为基础,并使用结构化查询语言(SQL)作为其数据库访问语言,ODBC 总体结构主要有四个部件:

\* 应用程序:执行处理并调用 ODBC API 函数,以提交 SQL 语句并检索结果。

\* 驱动程序管理器:根据应用程序需要加载或卸载驱动程序,处理 ODBC 函数调用或把它们传送到驱动程序。

\* ODBC 驱动程序:处理 ODBC 函数调用,提交 SQL 请求到一个指定的数据源,并把结果返回到应用程序。

\* 数据源:数据源包含了数据库位置和数据库类型等信息,实际上是一种数据连接的抽象。

各组成部件之间的关系如图 1 所示。

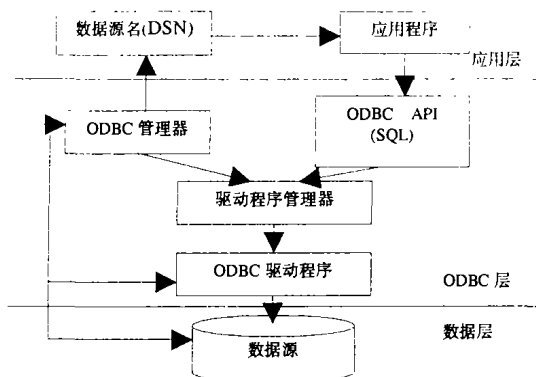


图 1 ODBC 各组成部件关系示意图

应用程序要访问一个数据库,首先必须用 ODBC 管理器注册一个数据源,管理器根据数据源提供的数据库位置、数据库类型及 ODBC 驱动程序等信息,建立起 ODBC 与具体数据库的联系,然后应用程序将数据源名提供给 ODBC,ODBC 才能建立起与相应数据库的连接。ODBC API 并不能直接访问数据库,必须通过驱动程序管理器与数据库交换信息,驱动程序管理器负责将应用程序对 ODBC API 的调用传给正确的驱动程序,驱动程序在执行完相应的操作后,将结果通过驱动程序管理器返回给应用程序,从而完成了对数据库的存取访问,达到应用程序操作数据库的目的。

## 1.2 JDBC 基本概念与组成原理

自从 Java 语言于 1995 年 5 月正式公布以来,Java 风靡全球。涌现了大量的用 Java 语言开发的应用系统,其中也包括数据库应用系统。由于没有一个 Java 语言的 API,编程人员不得不在 Java 程序中加入 C 语言的 ODBC 函数调用。这就使很多 Java 的优秀特性无法充分发挥,比如平台无关性、面向对象特性等。随着越来越多的编程人员对 Java 语言的日益喜爱,越来越多的公司在 Java 程

序开发上投入的精力日益增加,用 Java 语言访问数据库的 API 接口的要求越来越强烈,在这种情况下,SUN 公司决定开发一种基于 Java 语言的数据库应用程序接口,由此便产生了 JDBC。

JDBC(Java DataBase Connectivity)是 Java 与数据库的接口规范,它定义了一个支持标准 SQL 功能的通用底层的应用程序编程接口 API,它是由 Java 语言编写的类和接口组成,旨在让各数据库开发商为 Java 程序员提供标准的数据库 API<sup>[3]</sup>。JDBC API 定义了若干 Java 中的类,表示数据库连接 SQL 指令、结果集、数据库元数据等,允许 Java 程序员发送 SQL 指令并处理结果。通过驱动程序管理器,JDBC API 可利用不同的驱动程序连接不同的数据库系统。

JDBC 的总体结构类似于 ODBC,也主要有 4 个部件:应用程序、驱动程序管理器、驱动程序和数据源。各组成部分关系如图 2 所示。

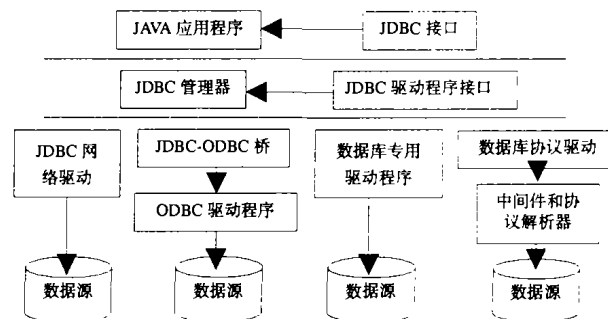


图 2 JDBC 各部件关系图

JDBC Driver API 分为 4 类:

\* JDBC 网络驱动:该方式将 JDBC 调用直接转换成数据库使用的网络协议,用户可以直接访问数据库系统。

\* JDBC-ODBC 桥接驱动程序:该方式通过 JDBC-ODBC 桥使用 ODBC 来访问数据库,因此,所有支持 ODBC 的数据库可不加任何修改而直接与 JDBC 协同工作。

\* 本地数据库专用驱动程序:该方式将用户应用程序的 JDBC 调用转换成对 Oracle, Sybase, Informix 等 DBMS 系统的调用。

\* 数据库协议驱动程序:该方式将 JDBC 与一种通用数据库协议驱动程序相连,利用中间件和协议解析器将该协议驱动程序与某种具体的数据库系统相连。

JDBC API 主要包括:Java. Sql. Connection, 完成某一指定的数据库的连接;Java. Sql. DriverManager, 用于处理装载驱动程序并为创建新的数据库连接提供支持;Java. Sql. ResultSet, 定义指定 SQL 语句执行的原始结果集,并提供对执行 SQL 语句后产生的结果的访问;Java. Sql. Statement, 在一个给定的连接中作为 SQL 执行的容器。其中 Java. Sql. Statement 又有两个子类型:Java. Sql. PreparedStatement, 用于执行预编译的 SQL 语句;Java. Sql. CallableStatement, 用于执行对一个数据库内嵌过程的调

用。

## 2 数据库中间件技术在 AES 中的应用

知识库在农业专家系统中占有相当重要的地位,从某种意义上说,决定了应用系统质量的优劣。因事实库(知识库)由输入的事实、中间和最终结果以及各种数据库表组成,故农业专家系统除了支持可直接访问\*.dbf 格式文件的 FoxBase 之外,还应支持诸如 Access, Oracle 等必须通过数据库中间件来访问的 DBMS,以提高 AES 的扩展性和包容性<sup>[4]</sup>。

AES 与数据库中间件相连时,首先必须建立 ODBC 数据源,通过运行 Windows 中的数据源(ODBC)来建立系统数据源;其次进行具体的数据库连接,其程序如下:

```
int handle, handle1;
handle = ODBCDSN_USE("数据源名"); // 先打开数据源,或者
handle = ODBCDSNPW_USE("数据源名", "用户名", "密码");
// 先打开有密码的数据源
handle1 = ODBCDB_USE(handle, "表名"); // 再打开数据源中的
具体的表
ODBCDB_BROW(handle1, "项名"); // 显示某一“项名”的所有
记录
X1 = ODBCDB_ITEM(handle1, "项名"); // 鼠标点中某一个记
录,赋值给变量 X1
ODBCSELECT handle1 ODBCWHERE ODBCDB_ITEM(handle1,
"项名 1") >= 条件 1 AND ODBCDB_ITEM(handle1, "项名 2")
= 条件 2 // 限定满足若干条件的所有记录
DBRES_BROW(handle1, "项名"); // 浏览限定满足若干条件的
所有记录
IF DBRES_NUM(handle) = 0 THEN {INFO "提示字符串"; EX-
IT;}; // 无记录提示并退出
ODBCDB_CLOSE(handle1); // 先关闭表
ODBCDSN_CLOSE(handle); // 再关闭数据源
```

以上 ODBC 提供的数据库连接只是适合 Windows 平台下的 AES,要实现 AES 在 Windows, Linux 平台下的平台无关性,其数据库连接必须借助于 JDBC 才能实现。JDBC 不仅能实现跨平台性,还无需配置,简化了最终用户的配置复杂度,实现对用户呈现透明的访问策略。其在 AES 中实现如下:

```
public Connection newConnection() // 新建连接
{
    Connection con = null;
    if (user == null) { con = DriverManager.getConnection
(URL); } // 无密码
    else { con = DriverManager.getConnection(URL, user, pass-
word); } // 有密码
}
public void DETItemBrowse() // 浏览
{
    DETJDBCContainer dc = new DETJDBCContainer(); // 创建
容器对象
    int h = dc.useDsn("ynnw"); // 打开数据源
```

```
int t = dc.useTable("sys"); // 打开表
String[] str = dc.browseDB(t, "zjmc"); // 取表字段的值
for(int i = 0; i < str.length; i++) System.out.println(str
[i]); // 显示该列
}
String[] browse(String field) throws DETException {
    ArrayList list = new ArrayList();
    Connection con = DETJDBCConnPool.getInstance().getCon-
nection("ynnw"); // 连接
    Statement stmt = con.createStatement(); // SQL 执行的容器
    ResultSet rs = stmt.executeQuery("SELECT" + field + "from"
+ tableName);
    while (rs.next()) { list.add(rs.getString(1)); }
    rs.close(); // 关闭结果集
    stmt.close(); // 关闭容器
    DETJDBCConnPool.getInstance().freeConnection("ynnw",
con); // 释放连接
}
```

推理机运行的结果如图 3 所示。

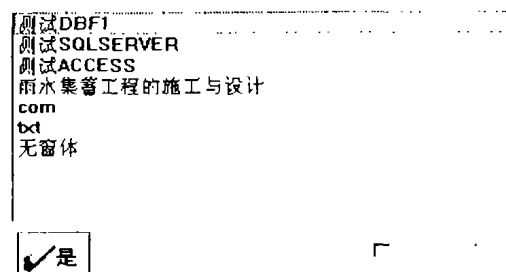


图 3 AES 推理机调用数据库的运行结果

## 3 ODBC 与 JDBC 的比较

### 3.1 相同点

ODBC 和 JDBC 技术都是以 X/Open 和 ISO/IEC 的调用级接口 (CLI) 规范为基础,并使用结构化查询语言 (SQL) 作为它们的数据库访问语言。JDBC 的设计在思想上沿袭了 ODBC,同时在其主要抽象和 SQL CLI 实现上也沿袭了 ODBC。即使是在总体结构上, JDBC 和 ODBC 也非常类似,都主要有四个部件组成:应用程序、驱动程序管理器、驱动程序和数据源,并且 JDBC 和 ODBC 都是独立于特定数据库的。

### 3.2 区别

JDBC 使用的是 Java 语言,而 ODBC 使用的是 C 语言,Java 语言天然所具有的平台无关、移植性强、稳定安全、分布式、面向对象等众多优点使得 JDBC 应用程序可以自然地实现跨平台特性,Java 和 JDBC 的结合可以让开发人员在开发数据库应用时真正实现“一次编写,到处运行”,更适用于 Internet 上异构环境的数据库应用。而 ODBC 只能适用于微软的 Windows 平台上。

JDBC 驱动管理器是内置的,驱动程序本身也可通过

(下转第 60 页)

共编程接口(如 JDBC)或专业数据库网关(Gateway)来访问它们,然后利用 XML Schema(或 DTD)把它们的数据模式表示出来<sup>[3,4]</sup>(由于 XML 具有良好的结构性,这个过程是比较容易实现的);而对于其它数据源,则要根据它们的具体情况,采用不同的访问接口(API),并设计出不同的 XML Schema(或 DTD)来表示它们的数据模式(比如对最简单的“标题+正文”型的普通文本文件,可以用<subject>和<content>两个元素来表示文件内容)。这样模式转换器就可以通过 DOM 或 SAX 接口来从包装器读入局部数据模式,并把它们按照用户指定的集成规则作适当转换后溶入到全局数据模式中。

VDB 是全局数据视图的表现形式,可以使用 XML 来实现它,为此只需把全局数据模式用 XML Schema(或 DTD)表示出来,并把集成规则存储在相应元素中即可,此时可以使用 XML 查询语句(如 XQuery)来访问 VDB。

另外也可以使用传统的关系数据库来实现 VDB,此时需要根据全局数据模式构造出一个关系数据库,此关系数据库同样不存储实际的数据,而只存储集成规则,此时可以使用标准的 SQL 语句向 VDB 发出查询请求。

访问转换器、结果合成器和模式转换器实质上都是一个 Java 包,里面包含各种不同功能的 API,它们根据不同的输入参数(如查询语句、集成规则)完成不同的功能。比如访问转换器由针对不同类型(如插入、删除)的查询语句(如 SQL 或 XQUERY)定制的 API 组成,这些 API 以应用层提交的查询语句为参数,根据集成规则把它们分解为针对具体局部数据源的子查询语句。

### 3 结 论

提出了一种使用虚拟集中方法实现的异构分布式数

据集成模型,并给出了基本的技术实现方法。采用该模型可以实现数据源的“即插即用”,即允许数据源的动态集成(增加一个新数据源时,只需在全局 XML 视图加入对新数据源的数据模式的 XML Schema 或 DTD 描述即可,另外集成规则可以随时调整),而且由于所有的数据访问是通过 VDB 对局部数据源实时进行的,所以也很好地保证了数据的一致性和实时性。

但由于所有的数据访问都要通过中央虚拟数据库作为中介落实到局部数据源上来进行,所以某些情况下(比如局部数据源分布在网络中不同的节点上,而网络传输速度较慢)查询速度可能会有些问题,对这个问题可以考虑通过增加一个查询结果的缓存库来解决;另外由于局部数据源可能会有很多,而且异构性也可能很大,所以针对不同数据源定制 wrapper 也是件繁琐和有一定难度的工作。

最后,为了使用的方便,可以考虑为系统设计一个友好的图形用户界面(GUI),可以让用户通过可视化的方法来灵活地完成局部数据源和 VDB 数据对象的选择、映射关系的建立等工作。以上这些都是需要进一步考虑和今后努力解决的问题。

#### 参考文献:

- [1] 高明,陈昕,李炜,等.基于 XML 实现异构数据源的联合使用[J].计算机科学,2002,29(3):83-84.
- [2] 戴青云,樊沛,李刚,等.Web-based 多数据源集成系统的研究[J].计算机科学,2002,29(9):119-封四.
- [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] Collins S R, Navathe S, Mark L. XML Schema Mappings For Heterogeneous Database Access[J]. Information and Software Technology, 2002, 44(4): 251-257.

(上接第 54 页)

Web 浏览器自动下载,无须安装、配置;而 ODBC 驱动程序管理器和 ODBC 驱动程序必须在每台客户机上分别安装、配置,这对于非专业人员来说,有些麻烦,因而在使用的便利方面远不如 JDBC。

ODBC 将简单特性和复杂特性混杂在一起,甚至对非常简单的查询都有复杂的选项,而 JDBC 刚好相反,它保持了简单事物的简单性,但又允许复杂的特性。

### 4 结束语

数据库中间件技术应用于 AES 中,知识库以统一的格式来调用不同的数据库,解决了调用不同数据库所带来的不一致性问题,并且采用 JDBC 便可简单地实现调用的跨平台性。数据库中间件技术的出现,使得软件开发人员可以避免繁琐的 DBMS,很方便地利用 ODBC 和 JDBC 提

供的 API 函数进行数据库连接的软件开发,从而减少了软件开发的工作量,缩短了开发周期,提高了软件开发效率和软件的可靠性。

#### 参考文献:

- [1] 周园春,李森,张建.中间件技术综述[J].计算机工程与应用,2002,38(15):80-82.
- [2] 齐亮.Linux/Unix 下 ODBC 的安装、配置与编程[EB/OL]. <http://www-900.ibm.com/developerWorks/cn/linux/database/odbc/index.shtml>, 2002-07.
- [3] Geof C C, Keeton F B. Java 完全探索[M]. 北京:中国青年出版社,2001.
- [4] 中科院智能所.“八六三”项目“开放式农业专家系统开发与信息处理平台”技术报告[R]. 北京:中科院智能所, 2003.