

基于 XML 的数据库访问中间件设计与实现

智永锋, 张 骏

(西北工业大学 自动化学院, 陕西 西安 710072)

摘 要: 为了完全地对程序开发人员屏蔽 SQL 语言, 提出了基于 XML 的数据库访问中间件模型。该模型利用 XML 脚本为操作不同数据库中的数据提供了一个跨平台的运行环境, 解决了对数据库操作的重复编码, 并且提供了数据连接池技术, 提高了对数据库访问的效率。

关键词: XML; SQL; JDBC; ODBC; 脚本

中图法分类号: TP311.52

文献标识码: A

文章编号: 1001-3695(2005)11-0087-02

Design and Implementation of Database Access Middleware Based on XML

ZHI Yong-feng, ZHANG Jun

(College of Automation, Northwestern Polytechnical University, Xi'an Shanxi 710072, China)

Abstract: A design of database access middleware based on XML is proposed to shield the SQL language completely. It provides the uniform platform to operate different database in different environments by XML script, resolve reduplicate coding of operation database. It also offers the technique of DBCP to improve efficiency of accessing databases.

Key words: XML; SQL; JDBC; ODBC; Script

随着计算机技术的发展, 许多应用程序需在网络环境的异构平台上运行。在这种分布异构环境中, 通常存在多种硬件系统平台, 在这些硬件系统平台上又存在各种各样的操作系统、数据库。这造成程序开发人员经常要编写重复的操作数据库的代码, 造成大量人力物力的浪费。如果把对数据库访问的程序做成中间件, 将使程序开发人员面对一个简单而统一的开发环境, 减少程序设计的复杂性, 可以将注意力集中在自己的业务上, 不必再为程序在不同系统软件上的移植而重复工作, 从而大大减少了技术上的负担。

数据库访问中间件一直是在中间件技术的边缘上, 独立的产品不多, 大都由数据库厂商随数据库产品一起提供。目前国内数据库访问中间件的发展还不是很成熟, 还没有完全做到对程序开发人员完全屏蔽 SQL 语言, 并且不支持数据连接池工作。

将提出一种基于 XML 数据库访问中间件模型, 它能够使应用系统透明、高效、统一的访问位于不同硬件平台、操作系统中不同数据库中的数据。它不但支持 JDBC, ODBC 等接口, 而且还支持数据连接池, 提高对数据库访问的效率。系统根据客户端要求, 自动完成配置 XML 脚本工作, 从而完全对程序开发人员屏蔽 SQL 语言。

1 数据库访问中间件(DBAM) 构架

1.1 系统组成

DBAM 主要由四部分组成, 它们是: XML 脚本信息提取器、客户数据源封装器、SQL 执行器、数据库数据源封装器。此外还有用于配置 DBAM 的部件, 它是驱动配置器和 XML 脚本

自动生成器, 这两部分是使 DBAM 正常工作必不可少的部件。这六部分的主要功能如下:

(1) 驱动配置器。在使用 DBAM 和 XML 脚本自动生成器前的一些预配置, 包括使用什么类型数据库, 采用 JDBC, ODBC 等类型接口, 还是利用数据连接池技术。如果采用数据池连接技术, 在驱动配置器上还要进行一些数据连接池的预处理配置。

(2) XML 脚本自动生成器。它根据客户端要进行的操作自动生成 XML 脚本, 其中客户端要包含一个隐含的字段, 使它与 XML 脚本关联起来。XML 脚本中包含了对数据库操作的 SQL 语言等所有的信息。

(3) XML 脚本信息提取器。它从生成的 XML 脚本中提取对数据库操作的信息, 然后把这些信息传递给 SQL 执行器。

(4) 客户数据源封装器。负责把客户端传递过来的参数和隐含字段按照协商好的协议打包, 使 SQL 执行器能够明白不同客户端传递过来的信息。

(5) SQL 执行器。从客户数据源封装器传递过来的信息中获得隐含的字段, 根据隐含的字段查到 XML 脚本, 再由 XML 脚本信息提取器获得必要的对数据库操作的信息, 然后由 SQL 执行器真正地进行对数据库操作。

(6) 数据库数据源封装器。它把 XML 执行器执行的所有可能的结果用打包的方式传递回去, 让应用服务器根据此信息进行必要的操作。

1.2 总体构架分析

驱动配置器和 XML 脚本自动生成器在 Web 客户端给用户提供了一个操作接口。通过把 Web 端的操作提交到 Web 应用服务器(服务程序), 由 Web 应用服务器完成对数据处理服务器的控制, 再由数据处理器完成对数据库的操作, 同时服务

器要反馈给客户端必要的信息, 以便于客户端进行有效的操作。整个结构可分为三个层次: Web 客户端、Web 应用服务器、数据处理服务器, 如图 1 所示。

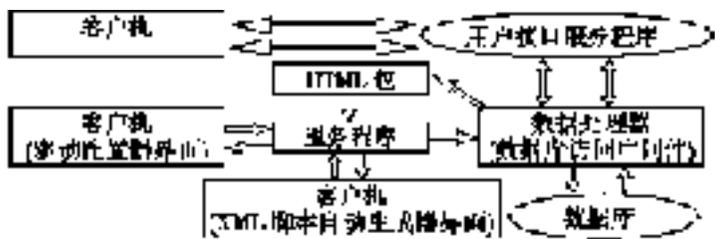


图 1 总体构架

驱动和 XML 脚本配置完成后, 就能够使用数据库访问中间件了, 它对程序开发人员提供了一个唯一的接口。当需要操作数据库时, 直接调用它就可以了。如: ResponseEnv rEnv = dbOp. DBhandle(env), 其中 ResponseEnv 为数据库数据源封装器中封装的对象, dbOp 为数据处理器中定义的变量, DBhandle 为数据处理器提供给程序开发人员调用的函数, env 为客户数据源封装器中封装的对象定义的变量, 使用非常方便, 完全屏蔽了 SQL 语言, 给程序开发人员减少了大量的时间。

1.3 DBAM 构架分析

DBAM 是整个系统的核心, 它是对数据库操作的中间件, 其他两部分只是起辅助作用。它的整体结构如图 2 所示。



图 2 DBAM 系统结构

客户数据源封装器接收到客户端发送过来的信息后, 按照定义好的协议进行封装, 然后把它发送给 SQL 执行器。SQL 执行器再按协商好的协议获得客户端发送过来的信息, 利用解析出来的隐含的字段从数据库查找 XML 脚本, 获得后再传送给 XML 脚本信息提取器。随后由 XML 脚本信息提取器分析 XML 脚本, 把获得的信息再传送给 SQL 执行器。SQL 执行器再把从 XML 脚本解析出来的 SQL 语句中的未知参数替换成解析出来的操作参数, 然后 SQL 语句就能真正地执行对数据库的操作, 执行完后把操作的结果发送给数据库数据源封装器。数据库数据源封装器接收到 SQL 执行器发送过来的信息后, 按照规定好的协议进行打包处理发送给应用服务器。这样一个完整的流程就完成了。

2 DBAM 实现关键问题

(1) XML 脚本

XML 脚本是实现 DBAM 的关键, 它包含了对数据库进行操作的所有信息。它是由使用 DBAM 的程序设计人员按照客户端将要进行的操作利用 XML 脚本自动生成器生成的。每段 XML 脚本对应一个唯一的隐含字段, 以便于从数据库中查找。它由 TableName, Auto, ParaNum, Operate, TwoSelect, SQL 六项标注组成。TableName 的值表示对数据库进行操作时操作的数据库中的那个表; Auto 的值表示对 TableName 的值进行 Insert 操作时, 此表的关键值是要要求数据库维护, 还是要求用户自己维护; ParaNum 的值表示客户端传递过来的参数个数; Operate

的值表示对数据库进行的是什么操作, 共有四个可选项, 它们是: Select, Insert, Update, Delete; TwoSelect 的值表示是否进行嵌套查询; SQL 的值表示对数据库进行操作的 SQL 语言, 但在这里的 SQL 语言中有包含的未知参数, 需要 SQL 执行器把这些未知参数替换后再执行操作数据库的动作。

(2) 客户数据源的封装

因为整个系统构架是基于三层或者多层结构来对数据库进行操作的, 因此客户端要让数据库访问中间件明白将要进行什么样的操作。如果采用传统的方法, DBAM 将无法明白客户端要让自己进行什么样的操作, 并且无法识别客户端传递过来的参数信息。在这里采用的是在客户端的操作中加入了一个隐含的字段, 通过这个隐含的字段 DBAM 查找 XML 脚本, 然后通过解析, 将明白要进行什么操作。为了让数据库访问中间件能够明白客户端传送过来的参数, 客户数据源封装器把客户端传递过来的参数名字按照给定的协议重新命名, 这样可以让 DBAM 解析出客户端传递过来的参数信息。比如客户端传递过来的第一个参数名字都命名为 #1, 传递过来的第二个参数名字都命名为 #2, 依此类推。通过这个协议, 客户机通过应用服务器程序把隐含字段和参数信息递给 DBAM, DBAM 就会明白传递过来的参数信息。

(3) 数据库数据源封装

传统数据处理器对客户机响应时, 客户机根据请求的信息已经知道反馈回来的将是什么类型信息。但对于数据库访问中间件, 传统方法不再适用, 因为它需要把各方面可能的反馈信息都要考虑进去。在这里是通过把所有可能的反馈信息 (进行了什么操作, 操作是否成功, 查询一维结果集, 查询二维结果集, 影响行数) 通过打包方式传递给应用服务器的。客户机接收到这个包后经过解析, 将明白进行的是什么操作, 操作是否成功, 操作的结果等。

(4) SQL 执行器对数据库的操作

在传统的方法中, 对数据库的一种操作就对应一个方法, 在每个方法中说明了一个具体的操作。对于数据库访问中间件, 应用服务器要求操作数据库时, 数据库访问中间件对应用服务器只有一个接口, 应用服务器只能调用这个唯一的方法。通过打包把要进行的操作和参数信息传递给 SQL 执行器, 再由 SQL 执行器根据这些信息来决定自己对数据库进行什么操作。在这里是通过 XML 脚本来实现的, SQL 执行器接收到对数据库操作的数据包后, 通过分析获得需要进行什么操作隐含字段, 通过这个隐含字段查找进行操作的 XML 脚本, 获得后再由 XML 脚本信息提取器分解这个脚本, 利用 SQL 执行器获得的操作参数替换掉解析出来的 SQL 语句中的未知参数。这样得到了对数据库真正操作的所有信息, 就可以对数据库进行操作, 再把操作的结果传送给数据库数据源封装器进行封装。

(5) 数据池的连接

它是通过对 Tomcat 服务器下 [TOMCAT_HOME] \conf 目录中的 Server.xml 文件的修改配置完成的, 在 Server.xml 文件中包含了对数据库进行操作的用户名和密码。为了系统的安全, 是通过 Server.xml 文件进行信息隐藏实现的, 要使其文件显示, 必须输入用户名和密码, 其中在这里的用户名和密码都是通过不可逆单项散列函数加密的。

(6) 系统的安全和速度

(下转第 91 页)

性等问题。

在开发的工程流程上,通过对各流程的共性操作进行抽象,对其进行规范,我们借鉴了集成开发环境的思想,将各开发活动集成起来,形成了运行于通用计算机上的 IDE 环境。该 IDE 环境集成了应用程序的编辑、编译、链接、模拟运行/调试、数据(包括应用程序目标代码)下载、上传等各个流程,其体系结构如图 5 所示。



图 5 集成开发环境结构

相对于一般意义上的 IDE 环境而言,这个 IDE 环境的特殊之处在于其模拟运行/调试部分。在本文应用中,开发中应用程序的调试要下载到读卡机 Flash 中,通过调度该程序进行运行来调试。这个调试过程涉及到代码下载、在目标系统中运行、调试结果上传分析,操作十分复杂、烦琐,很容易出错,尤其是对开发人员经验和操作熟练程度的依赖性相当大。而在这个 IDE 环境中,通过在宿主系统上的仿真运行进行调试,较好地解决了这个问题。隐藏于这个 IDE 中的流程规范化过程我们称之为过程复用。

4 结束语

目前,嵌入式系统的应用开发还仍是一个百花齐放的局面,不像通用计算机操作系统或通用计算机系统的体系结构(Wintel 架构)已基本形成事实标准的格局。由于这是一项极具市场潜力的应用技术,我们在应用开发中,若能站在更高的层次上,用体系结构的眼光来构架应用系统将会达到事半功倍的效果。比如在研发力量许可的前提下用微内核模式的思想

来构架我们的应用,形成一种源码级的可裁剪、伸缩能力强的微系统,或许将会成为一种更高级的开发模式。

本文是对一个涉及软件体系结构、软件开发过程进行改进的平台项目的总结。该项目投入实际应用半年多来,从应用效果来看,显著提高了开发效率,取得了较好的效益。专业开发人员和领域开发人员对这个项目的开展反应积极,并不断就开发细节提出改进意见,希望能将这个项目完善下去,形成一个规范的开发平台,对开发技术和开发过程进行规范,以提高开发效率,让更多的非专业开发人员参与到应用开发中来。这说明,这个项目确实具有实际价值,在系统体系结构上、系统开发过程上的改进作用确实明显。本文介绍了基于嵌入式应用开发的宏观体系结构的设计和表达。所使用的方法是形式化的,还不太规范。这方面的研究和开发工作还有待进一步的努力,才能加以规范并推广使用。

项目使用的手持读写机型号为 TS-D8(配置为 MC68EZ328 + 2M Flash + 128K SRAM)。

参考文献:

- [1] Li, Zhiyan Wang, Haizan Zeng. Adaptive Color Document Images Binarization for Text Retrieval [C]. Proc. of SPIE, 2004. 35-44.
- [2] 李谊,王知衍,曾海赞. 采用混合神经网络高精度提取机票字符 [J]. 计算机工程与应用, 2004, 40(8): 209-211.
- [3] 万建成,卢雷. 软件体系结构的原理、组成与应用 [M]. 北京: 科学出版社, 2002.
- [4] Mary Show, David Garlan. 软件体系结构(影印版) [M]. 北京: 科学出版社, 2004.
- [5] Kirk Zurell. 嵌入式系统的 C 程序设计 [M]. 艾克武, 等. 北京: 机械工业出版社, 2001.

作者简介:

严和平(1969-),男,博士研究生,主要研究领域为图像分析、模式识别及相关成果的开发转化;邓荣佐(1969-),男,工程师,主要从事 3S 技术、网络数据库及工程地震等应用研究;王知衍(1949-),男,IEEE 会员,教授,博士生导师,主要研究领域为图形图像处理,现负责高速机票影像处理系统设计。

(上接第 88 页)

由于对每一个数据库操作都要查找它需要的脚本,通常情况下对数据库操作比对文件操作的速度要快,因此把 XML 脚本放到数据库中,使它和隐含的字段关联起来,并且这样可以很好地解决安全的问题。如果把 XML 脚本放到文件中,不但容易被别人盗取,而且容易出错。驱动配置器配置的信息为了使用的方便,把它们放到了文件中,但对其进行了隐藏。使用时通过定义静态变量的方法,把它们初始化后放到内存,以后就不再访问,加快程序运行的速度。

3 结束语

本文利用 XML 是一种定义表示语言,能够在一起定义大量信息的特性,把它作为操作数据库的脚本,实现了完全屏蔽 SQL 语言的操作数据库中间件,只要求程序开发人员按照一定的协议进行操作即可。并加入了对数据连接池的支持,实现了对关键文件 Server.xml 的信息隐藏。

参考文献:

- [1] [美] Tom Myers, Alexander Nakhimovsky. Java XML 编程指南 [M]. 北京: 电子工业出版社, 2001.
- [2] [美] John Camell, Bjarki Holm. Oracle 8i 应用高级编程——Java,

PL/SQL 和 XML 深入开发 [M]. 北京: 清华大学出版社, 2002.

- [3] [美] Microsoft Corporation. SQL Server 2000 数据库开发手册 [M]. 北京: 科学出版社, 2001.
- [4] [美] Leon Atkinson. MySQL 核心编程——高级开发指南 [M]. 北京: 清华大学出版社, 2003.
- [5] [美] Craig S Mullins. DB2 开发人员指南 [M]. 北京: 机械工业出版社, 2002.
- [6] 邵佩英,杨孝如. Sybase 数据库系统基础知识 [M]. 北京: 中国水利水电出版社, 1997.
- [7] 严超,谈金泉,张舒怡. 基于 XML 的分布式系统集成方案 [J]. 计算机工程, 2003, 29(21): 114-115, 119.
- [8] 朱跃龙,郑光迎,张萍. 基于 CORBA 的三峡数据存取中间件的设计与实现 [J]. 计算机工程, 2003, 29(21): 107-109.
- [9] 范国闯,陈宁江,钟华. Web 应用服务器: 新一代中间件 [J]. 计算机科学, 2004, 31(1): 1-4.
- [10] 杜小丹,张凤荔,周明天,等. 中间件技术在 Web 数据库访问中应用研究 [J]. 计算机应用研究, 2004, 21(10): 56-57, 99.
- [11] 谢天保,雷西玲,张景. 基于 Web 的数据库访问模式的研究 [J]. 计算机应用研究, 2002, 19(8): 75-76, 110.
- [12] 徐琨,刘志镜,来琳涵. Web 数据库访问的中间件技术分析和研究 [J]. 计算机工程与科学, 2002, 24(4): 54-56.

作者简介:

智永锋(1979-),男,河北元氏人,硕士研究生,研究方向为网络与信息安全技术;张骏(1964-),男,山东枣庄人,研究生院副院长,教授,博士,研究方向为网络与信息安全技术。