

实现企业信息集成的一种有效方法

杨北约 (张家口市农业机械工程学校 075000)

摘要:本文介绍一种利用 OO 技术及开放数据库互联(ODBC)标准实现的企业信息集成的思想、方法、结构及其关键技术。

关键词:OO、ODBC、DBMS 分布式处理 Client/Server

一、引言

企业的信息集成包括多种设备间网络互联的物理集成;异构数据库间数据互操作的信息集成;不同部门之间的功能集成;多种专业交叉渗透的相关学科的集成以及不同阶段投资的价值集成。其中,物理集成是信息集成的必要基础,功能集成、相关学科集成以及价值集成是信息集成的延伸及必然结果。

实现企业信息集成必须具备两个条件,其一是从企业的全局角度出发,运用系统工程的观点和方法,建立企业全局信息模型;其二是利用计算机硬件、软件及网络技术建立信息集成环境。信息集成环境是全局信息模型的支撑环境和实现手段。只有建立符合企业需求的信息集成环境,才能借助于全局信息模型,实现企业的信息集成。

本文从企业的实际需求出发,给出了企业信息集成环境的结构,简述了该集成环境的基本功能,并介绍了利用 OO 技术和 ODBC 实现该集成环境的关键技术。

二、企业信息集成环境的结构

建立企业信息集成环境,目的在于为 CIMS 种类繁多,数量庞大的信息提供一整套有效而合理的组织手段,使不同载体不同形式的信息得以充分共享,并取得最佳的使用效率,从而为实现 CIMS 信息集成提供有力保障。

在 CIMS 环境下,信息的存储及其处理具有环境的异构性及处理的分布性等特点。它的异构性不仅体现在不同硬件平台和网络环境,还体现在不同操作系统和不同数据存储环境(DBMS)上,甚至有的数据还直接存储在文件系统中。不仅如此,各种数据的生成场所,它们的生成期与生命周期也各不相同。这些异构和分布的数据也有着不同的访问方式与规则,如逐条逐项的数据访问

方式和批处理、事务型的访问方式。总而言之,一个成功的信息集成环境应该至少具备这样的能力,即无论数据存放在什么地方或以什么形式存储,应用都能够以一种统一的方式对其进行访问和更新,并能对分布在不同计算机上的各类事务进行协调与控制,以便方便、快捷、实用地处理复杂数据对象。

为满足 CIMS 信息集成的要求,实现信息在异构分布环境下的处理与共享,必须采用开放式体系结构。近年来,分布式开放体系结构中最引人注目是客户/服务器(Client/Server)模式这种结构与主机/终端结构和局域网分布式文件系统结构相比,能以一种更为协调和有效的方式进行信息管理与操作,可以说信息集成环境建立在 Client/Server 体系结构之上更能出色地发挥集成环境的优势和潜力。

然而,无论采用何种方式组建信息集成环境,都不可避免地会遇到异种操作系统下的异种 DBMS 互联的问题,如何以一种有效的方法来解决这个问题,而又不牺牲集成环境的分布性和开放性呢?

美国微软公司推出的基于 Windows 的开放服务体系(WOSA)的数据库部分,定义了一种 ODBC(Open DataBase Connectivity)的接口标准,它允许 Windows 的应用可以不加修改地与多平台的异种数据库进行互联。目前这一接口标准已受到大多数数据库厂家的认可,并提供了如 Oracle, Sybase, Ingres, DB2, Informix, DBASE, Foxpro, Access 等名类数据库管理系统(DBMS)的驱动(Driver)。ODBC 技术在信息行业日益受到重视与关注,各软件厂商纷纷推出基于 ODBC 的数据库产品,其中包括以信息系统前端开发工具而闻名的 PowerSoft 公司的产品之一 PowerBuilder。

ODBC 接口支持通过标准 SQL(Structure Query Language)来访问 DBMS 中的数据,并且提供最大限度的可

交互性(interoperability)——一个应用可同时访问不同的DBMS,而且用户可以动态追加各种DBMS的驱动模块(Driver)来增加访问的范围与灵活性。ODBC接口与嵌入式SQL(Embedded SQL)不同,当应用在不同硬件平台或操作系统之间移植时,嵌入SQL的源码将必须重新在新环境下进行编译。而ODBC的应用则只需挂接新的Driver就可以不加修改地在新环境下运行了。

众多独立的DBMS驱动模块(Driver),隐藏了不同DBMS的数据操作细节,按照ODBC定义的标准提供统一的用户接口。这种方法很类似Windows的打印功能,众多的Printer Drivers赋予Windows的打印设备无关性,同样众多的ODBC Driver也使基于ODBC的应用具备了数据操作的DBMS无关性。

结合ODBC的上述特点组建的信息集成环境具有以下层次结构。最上层的应用具有统一的数据定义和操作界面,基于ODBC接口标准的信息集成环境提供数据定义和数据操作的功能,为上层应用服务。ODBC层不仅屏蔽了其下层网络协议与实现细节,而且为最底层异种操作系统及其上层的异种DBMS提供了有效的互联方式。

基于ODBC技术的信息集成环境具有更好的分布性与开放性。它的分布性表现在从体系结构上实现了真正的分布式计算和分布式数据存储。ODBC接口标准使得应用程序以统一的方式对不同结点上的数据进行透明操作。同时,在基于ODBC的信息集成环境的支撑下,信息的组织具有更大的灵活性,例如,有些局部信息生成周期短,使用率高,并且对实时性要求较强,可以将这些信息在客户机一端以临时数据的方式进行存储和处理,在服务器上存储某些全局共享信息,使服务器端集中完成信息的全局管理和调度。上述数据组织方式一方面可以均衡系统中各部分的负荷,大大减缓对服务器处理能力的依赖,解决了系统信息处理的瓶颈,提高处理效率,另一方面可以实现全局与局部数据的合理分布,减轻网络负载,提高信息的传输率与使用率。

由于ODBC技术比较成功地解决了异种操作系统与异种数据库互联的困难,从而使系统更具开放性。应用对操作系统和DBMS的选择更加自由,如服务器端可以采用Windows NT或SCO Unix等操作系统,SQL Server、Oracle、Sybase、Ingres等大型DBMS进行多任务的事务管理;在客户机端采用Windows 3.1或Windows 95,并配置Access、Foxpro等小型而高效的DBMS,完成局部信息处

理,并提供友好的用户界面。

三、信息集成环境的基本功能

运用ODBC技术,结合Client/Server体系结构实现的信息集成环境,具备以下四项基本功能。

1. 数据定义功能

完成数据项的定义,表的定义,视图、索引的定义以及表的关系定义,并提供相应的维护功能。

2. 数据操作功能

包括数据的录入、修改、查询与删除操作,并提供一致性的维护功能。由于ODBC具有一致性接口标准,使得数据具有一致的操作方法与界面。

3. 数据库互联功能

各子系统分别登录与之相关的数据库(以数据源名的方式),并建立联接。联接建立以后,数据源名(DSN)唯一标识数据来源,屏蔽底层通信协议及数据格式。

4. 数据通信

完成各子系统之间或子系统与全局数据管理系统之间的数据交换与数据转储,并提供分布事务管理机制来保证数据非同步更新的数据一致性。

四、实现的关键技术

下面着重介绍采用ODBC技术来实现信息集成环境的思路、方法及主要过程。

1. ODBC的接口定义与组成部分

ODBC的接口定义如下:

- * ODBC函数库提供应用与DBMS之间的联接手段,处理并执行SQL命令,返回结果。

- * 基于X/Open和SAG.(SQL Access Group)的SQL CAE draft Specification (1991)中规定的SQL语法集。

- * 标准错误码集

- * 标准的DBMS联接登录方式

- * 标准数据类型表示

其接口的柔性表现为:

- * 包含SQL语句的字符串可以显式地出现在源代码中或在运行时刻(Run-time)构造。
- * 同样的目标代码可运行在不同DBMS之上。

- * 应用与底层数据通信协议是隔离的,可以不予考虑。

- * 数据可按应用定义的格式发送或获取。

ODBC 的组成如下,共包括五个部分:

- * ODBC API

包括一个函数调用库,一套错误代码,用于 DBMS 数据访问的标准 SQL 语法。

- * ODBC Driver Manager

名为 ODBC.DLL 的动态链接库,对应用是透明的,根据应用的需要加载 ODBC 的数据库 Drivers。

- * ODBC Database Drivers

一个或多个动态库,分别针对特定的 DBMS 处理 ODBC 的函数调用(也包括针对文件系统的 Drivers)。

- * ODBC Cursor Library

名为 ODBC.CURS.DLL 的动态库,用以在 ODBC Driver Manager 与 Drivers 之间实现数据滚动。* ODBC Administrator

登录数据源的工具。

2. 利用面向对象(OO)方法实现 ODBC 数据访问机制

为实现信息集成环境的前述功能,ODBC 数据访问机制应该具备较强的适应性和灵活性。ODBC (ver 2.01)共提供了 54 个 API 函数,足以完成 ODBC 涉及的所有数据访问功能,然而不同 DBMS 的 Driver 并不完全支持这 54 个 API 函数,因为 Driver 的划分等级通常有两种途径,分别按对 SQL 语法和对 API 函数的支持程度来定。ODBC 提供的全部 API 函数被划分为三个等级:核心级(Core),扩展一级(Extension Level 1),扩展二级(Extension Level 2)。大部分 Driver 均支持核心级的函数调用,但支持全部扩展二级函数的 Driver 很少见。具体到某个 API 是否被支持,需要调用 SQLGetInfo 或 SQLGet-Functions 函数来确定。另一方面,不同 Driver 实现的 ODBC 访问方式可能各自有些使用上的限制,如某些 Driver 不支持 Cursor 定位的更新与删除(例如:Delete Where Current of Curname),某些 Driver 只支持唯一的活动联接(Active Connection)或活动语句(Active Statement),甚至某些 Driver 只支持单向只读的滚动方式等等。因此,在实现 ODBC 访问机制时,必须尽可能全面地考虑各种情况。

可以采取的策略包括,第一,尽量采用核心级 API

函数和部分扩展一级函数,对于扩展一级和二级函数最好事先判定是否被当前版本 Driver 支持。第二,在某个函数不受支持的情况下,采用其他途径或其他函数来替代完成该函数的功能。第三,在不清楚某种访问方式是否被支持时(例如通过 API 函数不能确定的情况),可以此种访问方式试探性执行,再考查其返回值,同时屏蔽掉错误信息显示;若不被支持,紧接着调用备用访问方式。

可见为增强信息集成环境的适应性和灵活性,实现过程是比较复杂的,而且考虑到必须的可靠性与实用性,采用“松散”的编程方法是不可取的。当前,面向对象(OO)技术正受到广泛的重视与应用。在 OO 方法中,基本元素是对象。所谓对象即数据以及可以在其上施行的操作所构成的独立单元。对象将数据(属性)与数据操作(方法)封装在一起,利用继承性和多态性将对象之间的共性与个性以一种结构化的层次体系描述出来。而面向对象技术运用在编程中,通常采用 C++ 语言,以类作为对象的原型,使应用程序具有很大的灵活性和可复用性。将 ODBC 的访问功能以类的形式封装起来,可以隐藏复杂的实现细节,使用户将注意力集中在如何使用上,增加了系统的易用性。由于类体系具有较好的层次结构,大大减轻了维护与修改的困难程度,拥有良好的可扩展性。

目前能够完成 ODBC 编程的开发工具只有两种:Microsoft 公司的 Visual C++ 1.5(或更高版本)和 Visual Basic Professional Edition 3.0(或更高版本)。这两种开发工具都提供了面向对象的类体系结构。完全可以利用 Visual C++ 1.5 中 MFC 基本类库 2.5 版的数据库类为原型,开发出适应信息集成环境的 ODBC 数据访问机制。

五、结语

开放数据库互联(ODBC)技术是一门极为实用的技术。利用这种技术,结合 Client/Server 模式是实现 CIMS 信息集成的一种有效方法。我们在国家 863/CIMS 主题支持项目——快速应用系统开发与集成支撑系统(RADISS)的研制过程中,采用并细化了这种方法,建立了一整套基于 ODBC 的信息集成环境的体系结构,证明了它的实用性和有效性。可以预料,随着 ODBC 技术的推广和应用,这种方法必将在信息集成领域形成深远的影响。

(来稿时间:1996年11月)