

# 可定制的通用数据访问模型的设计与实现

夏满民<sup>1</sup>, 邓 芳<sup>2</sup>

(1. 北京邮电大学 网络教育学院, 北京 100088; 2. 北京邮电大学 计算机科学与技术学院, 北京 100876)

**摘 要:** 通过对元数据库结构定义和通用数据访问的设计, 给出了一个可定制的通用数据访问模型的设计及实现方法, 使能通过定制完成数据访问和展示, 降低数据库应用系统的开发难度。最后给出了模型在远程教学系统和器材管理的应用。

**关键词:** 定制; 数据访问模型; 元数据

中图法分类号: TP311.12

文献标识码: A

文章编号: 1001-3695(2006)02-0181-02

## Design and Implementation of Customized General Data Access Model

XIA Man-min<sup>1</sup>, DENG Fang<sup>2</sup>

(1. College of Network Education, Beijing University of Posts & Telecommunications, Beijing 100088, China; 2. College of Computer Science & Technology, Beijing University of Posts & Telecommunications, Beijing 100876, China)

**Abstract:** The design and implementation of customized general data access model is presented. Key components, general meta data tables and data access methods, are introduced. Database applications development can be completed only by custom\_building on this model. Long distance teaching and repertory management applications built on this model are also given.

**Key words:** Custom\_built; Data Access Model; Meta Data

在数据库应用系统的设计与开发中, 虽然用户高层业务需求千差万别, 但在具体系统实现时, 底层的数据库访问及结果展示则存在着共性。有许多关于通用数据访问模型或组件等方面的研究, 但它们主要是通过通用的数据访问方式来实现的。要使系统不依赖于某一种 DBMS, 利于开发多种 DBMS 应用系统, 主要的方法是: 通过将数据库访问有关的函数集中于一个数据库接口模块中, 即把该模块作为数据访问的中间件, 对上层应用程序提供访问数据库的支持。而本文给出一个可定制的通用数据访问模型, 不仅将共性的数据访问进行封装, 而且通过通用数据库结构的定义使开发人员和系统维护人员能根据需求对相应的访问和展示进行定制, 极大地提高开发效率和系统灵活性, 降低开发难度。

### 1 设计思想

可定制的通用数据访问模型是要能提供一个数据库应用系统的二次开发平台, 相关的应用和功能通过定制即可完成, 而不再需要写代码。模型的主要思想体现在两个方面: 开发和人员通过定制完成数据访问和结果展示功能开发; 对结果展示和数据访问用一个通用的数据访问模块和接口进行处理。其结构和主要工作过程见图 1。

从图 1 可以看出, 模型主要分三层: 用户接口层、数据处理层和数据存储层。用户接口层主要提供用户界面, 接收来自开发、维护人员及用户关于数据定制和访问的请求; 数据处理层主要完成数据定制处理和数据访问处理; 数据存储层是相关数据的存储。与通常的数据库系统相比, 这里的数据除了每个具

体的系统所应该有的具体数据外, 还有关于数据访问定制的元数据, 它来定义数据访问的具体内容和规则。

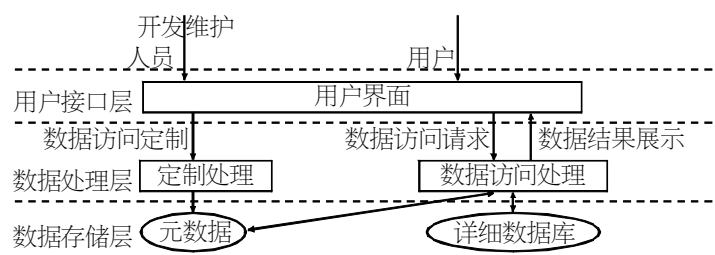


图 1 模型结构及主要工作过程

元数据库、定制处理以及数据访问处理构成了模型的核心。元数据库存储关于数据访问和结果展示相关信息数据的定义, 而详细数据是存储具体应用系统的具体数据。当需要进行具体数据库应用系统开发时, 不需要再编写数据访问的代码, 而只需要完成对数据访问相关元数据的定义。当用户需要进行数据访问时, 数据访问处理接收访问请求, 先在元数据库中查看其相应的定义, 根据定义的规则访问详细数据库中具体数据, 并依据定制的格式将结果展示。

### 2 核心部分设计

#### 2.1 元数据库结构设计

元数据库是关于数据访问内容和规则定义的存储, 是实现可定制模型的关键。在我们的设计中主要有以下两张关键表:

(1) 基表查询元数据表。主要定义查询与对应基表或视图关系, 每一类查询在此表中有一条记录说明, 用唯一的查询标志进行描述, 描述的内容还包括查询所需要访问的基表或视图, 及这类查询可能涉及到的某些隐含的条件, 如访问请求用户所能看到的数据范围等。其表结构主要包含的字段包括:

查询标志、查询表名、查询描述、隐含查询条件、隐含查询字段。

(2) 基表查询列信息数据表。主要定义查询与基表中各列之间的关系,每一类查询所涉及到的相应基表的每一列都在此表中有一条记录对应。其表结构主要包含的字段包括:

查询标志、列名、列序号、列描述、列数据类型、列长度、列小数位、取值范围下界、取值范围上界、显示宽度、浏览标志、修改标志、列类型、查找表主列、查找表名、查找表结果列、查询条件列标志、超级链接类型、超级链接查询标志、超级链接查询主键、超级链接查询值列、初值标志、初值、查询条件初值、查询条件初值 2、显示对齐方式、显示颜色等级。

在表中除了定义列的一般性信息如数据类型、长度、取值范围等外,需要定义的还有关于此列的两类信息:

查询条件标志。其中查询条件列标志用于描述这列相关的条件约束如等于、某一范围或是包含等,超级链接类型和超级链接查询标志定义是否需要再根据此列链接相关内容的深入查询,如果需要则定义其查询标志及查询条件(定义相应的超级链接查询主键和超级链接查询值列)。

定义与结果显示相关的信息。其中列类型、查找表主列、查找表名、查找表结果列定义:当在屏幕上此列位置显示的结果不是这列存储的真正信息时,需要进行信息查询和转换,比如省份在数据库表中如果以数字代码存储,而显示时需要实际省份名称,就需要将代码表的相应信息读出显示,就需要定义这种列类型,相关的数据如代码表的表名,查询表的主键和显示列的内容就需要事先定制;浏览标志定义此列是否需要在展示中出现;初值、查询条件初值、查询条件初值 2,分别在查询页面和更新页面中所应显示出的初值;显示对齐方式,显示颜色等级为此列在显示时的一些属性定义。

这两张元数据库表定义了进行统一数据处理时所需要的数据信息,包括控制信息和显示信息。开发时的定制也是对这两张表的数据进行定义,满足通用模型中统一数据处理所需要的数据要求。

### 2.2 数据处理设计

数据处理包含两部分,即定制处理和数据库访问处理。定制处理是开发和维护人员在开发和维护时进行定制时使用的,数据库访问处理是具体的应用系统在运行时完成的显示、查询、更新等数据处理功能。

#### (1) 定制处理

定制处理的功能主要是完成各种功能的相关数据的定义,是元数据的定义过程。定制处理就将相应的元数据写入元数据库表中。

定制处理根据数据处理所需要的相关信息提供相应的定制界面。在定制界面上用户对需要操作的数据库表进行定义,如操作的条件,对表中每一列的说明,是否需要显示,是否需要链接,链接的内容,显示的方式等,系统将为其定义的给一个唯一的查询标志,相应的数据将通过定制处理写入元数据库表中。

定制过程也可以看成是用此通用模型进行应用系统的开发过程和需求变更后的维护过程。

#### (2) 数据库访问处理

数据库访问处理主要分为两部分,即数据处理和数据库访问,如图 2 所示。

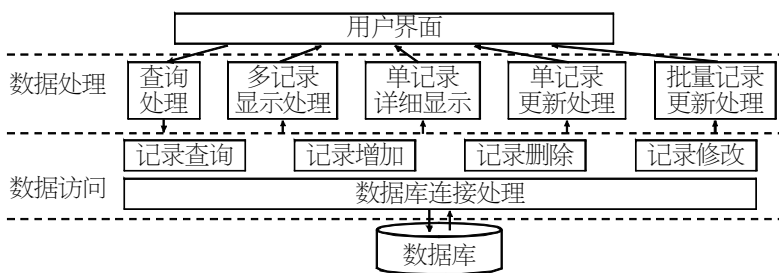


图 2 数据库访问处理结构

数据库访问是数据库访问处理的底层,与许多通用数据库访问接口类似,完成数据库连接、记录查询获取、基本的增、删、改操作,对上层的数据处理提供访问数据库的支持。

数据处理中主要完成的功能是查询处理和显示处理。查询处理的主要功能如下:

根据用户选择菜单所对应的功能获取查询标志;

按查询标志在元数据库表中读取相应元数据,给出查询界面,并获取在元数据库表中存储的某些隐含查询条件;

接收用户填写的查询条件后,与隐含查询条件一起形成完整的查询条件,交由数据库访问处理。

其中 是关键步骤,也是根据元数据完成定制和通用处理的主要一步。

显示处理主要包括这四个大的公共处理模块:多记录结果显示处理、批量记录数据更新处理、单记录详细显示处理、单记录数据更新处理。当查询处理完了以后,需要调用相应的数据显示模块进行结果展示。在调用时,除了传给它查询的数据外,还需要将在查询处理时访问到的有关显示的元数据信息一起传送,这些元数据信息就是主要在元数据库表中存储的关于列显示的信息,如显示的类型、初值、宽度、是否有链接等的数据库内容;还需要将链接的相关控制信息传送,如链接的表、链接查询的条件等。

在显示时分了单条记录的详细显示处理和多条记录的显示处理,这在数据库应用系统是很常见的,显示和处理上不同。在更新的时候也作了类似的处理。

### 3 实现及应用

目前模型已经在以下的平台上实现:数据库采用了 ORACLE9i,中间的数据处理层利用了 WebLogic,采用 J2EE 的结构将中间层的数据各处理封装成 JavaBean。

我们完成了应用模型两个具体的数据库应用系统:远程教学的教务管理系统。主要应用模型通过定制,完成了教务管理系统中的大部分功能,即招生管理、学籍管理、考务管理和教务管理等。器材库存管理系统。通过定制,完成了系统中的入库、出库、查询统计等功能。两个系统均已投入使用。在实际应用开发中,开发人员在熟悉了模型后,只需要进行功能的分析,然后进行定制,完成相关元数据的定义即可,而对大部分功能无需再写代码。

实践证明模型在灵活性的保证、开发效率的提高、开发复杂度的降低等方面有很大的优势。

(下转第 185 页)