

UML 模型向关系数据库的映射方法初探

张念春¹ 杨寿保²

¹ (江苏淮海工学院计算中心, 连云港 222005)

² (中国科学技术大学计算机科学与技术系, 合肥 230026)

E-mail: zhangnc@china.com

摘要 关系数据库的数据模型变得越来越复杂, 用传统方法分析和描述变得越来越困难, 在数据密集型的应用系统中, 应用逻辑与数据存储存在差异, 不利于开发人员的交流和开发过程的管理, 这是文章考虑把对象技术引入关系数据库逻辑设计的出发点。统一建模语言 UML 是对象建模技术的发展结晶, 可用于数据库的数据模型分析。通过技巧性的方法, 可把 UML 模型映射到关系数据库, 从而在数据库领域得到对象技术的诸多益处。

关键词 统一建模语言 对象建模技术 关系数据库 映射

文章编号 1002-8331- (2002) 19-0212-03 文献标识码 A 中图分类号 TP311.13

The Primary Discussing of the Method of Mapping UML Model to Relationship Database

Zhang Nianchun¹ Yang Shoubao²

¹ (Computer Center, Huaihai Institute, Lianyungang 222005)

² (Department of Computer Science and Technology, USTC, Hefei 230026)

Abstract: The data model of relationship database is becoming more and more complex so more and more difficult to be analyzed and described using traditional method. Mean while, there are differences between application logic and the data store in data-dense applications, these go against the management of development process and the communication between developers. This is the beginning point for considering to introduce the object-oriented technology to the designing of the database logic. UML is the consequence of the development of the object-oriented modeling technology, and can be used to analyze the data model of database. Through skill method, people can map the UML model to the relationship database and get the benefit from the object-oriented technology.

Keywords: Unified Modeling Language (UML), Object-Oriented Modeling Technology, Relationship Database Mapping

1 引言

面向对象的技术已经成为构建主流商业系统的方法, 广泛应用于诸多领域。然而把对象建模技术用于数据库的数据模型分析, 却是一个非常值得探讨的新领域, 理由有以下两点:

其一, 关系数据库的数据模型变得越来越复杂, 用传统方法分析和描述变得越来越困难。开发一个复杂的数据库应用系统, 数据模型的分析是十分关键的。如果只涉及一张表, 其模型是很直观很容易理解。然而现在的数据库应用涉及的领域越来越广泛, 问题越来越复杂, 常常包含越来越多的表格, 在这种情况下, 传统方法就显得力不从心^[1]。

采用面向对象的方法进行分析, 可以控制问题的复杂性^[2], 从而使数据库的数据模型变得清晰、易懂和易于开发。因此早在 1996 年, 人们就开始使用面向对象的方法进行数据库的逻辑设计。面向对象方法的引入, 使模型中的实体和最终的表的数量大大减少, 这正是利用面向对象技术进行数据模型设计的最大优点。此外, 还增强了结构的健壮性和灵活性, 当用户提出新的无法回避的需求时, 大大降低了在数据模型上做较大修改的可能性^[3]。

其二, 在数据密集型的应用系统中, 应用逻辑与数据存储存在差异, 不利于开发人员的交流和开发过程的管理。现在的开发环境越来越多的是面向对象的, 而存储机制却是不同于此的关系型数据库, 这两者之间存在着很大差异。这种差异使系统的开发活动不能统一。典型的情况是, 越来越多的应用系统是三层甚至多层体系结构^[3, 4], 在此情况下, 用户接口层和应用逻辑层是用面向对象技术开发的, 而数据库多数仍然是关系型的。

因此采用对象技术对数据库建模是一个必然选择。然而问题是, 对象模型毕竟与 E—R 模型是有差别的, 在采用对象建模技术分析获得面向对象的数据模型后, 怎样得到关系型的数据库呢? 这正是该文要讨论的问题。

2 UML 的模型概念

面向对象方法的种类非常多, 它们的理论体系虽然是一致的, 但工程体系却常常有较大差别^[2], 这给人们的使用带来了很大困难。另一方面, 随着系统日益变得复杂, 用来描述它们的清晰、简洁的方法正变得越来越重要。因此由 Grady Booch、Jim

Rumbaugh 和 Ivar Jacobson 开发出统一建模语言 (UML)^[5]。UML 是一种定义良好、易于表达、功能强大且普遍适用的建模语言。它溶入了软件工程领域的新思想、新方法和新技术。它的作用域不仅限于支持面向对象的分析与设计,还支持从需求分析开始的软件开发的全过程。

UML 比 ER 具有更新、更丰富的模型语言。使用 UML 可以捕获 ER 方法能够捕获的一切信息。而且,由于 UML 可以描述方法或操作,因此可以通过这一点捕获像触发器、索引和各种各样的约束信息。UML 比传统的 ER 技术更富有语义上的表达,例如可以表达继承关系、聚合和行为特征等。通过可视化的建模,UML 可以使应用开发与数据库开发共享可视的约束信息,从而使开发人员交流顺畅,更有效地一起工作。通过 UML 连接对象和数据模型,用户的需求可以直接跟踪到数据库中去,从而方便了建立、维护和更新。UML 的面向对象的概念与其他面向对象著作所提及的概念有不同之处。为了很好地应用 UML,有必要将这些概念明确一下。

UML 的概念和模型可以划分为以下范围^[6]:静态结构、动态行为、实现构造、模型组织和扩展机制。表达 UML 的静态结构的概念包括类 (class)、属性 (attribute)、关联 (association)、依赖 (dependency)、泛化 (generalization)、聚集 (aggregation) 等。表达 UML 的动态结构的基本概念有事件 (event)、操作 (operation)、动作 (action)、链 (link) 和消息 (message) 等。

在一个典型的数据密集型应用中,并非所有的概念所表达的信息都需要考虑永久存储。因此,该文只关注与数据存储有关的概念。在用对象建模技术分析获得 UML 表达的数据模型后,通过映射可以得到关系型的数据库。

3 将 UML 的静态模型映射到关系数据库

3.1 基本元素的映射

这里的基本元素指的是类、对象、对象标识和属性类型等有关类的基本方面。

(1) 类的映射

类可以映射为一个表,同时将属性映射为表的各列。有时需要创建附加的列,用于对象标识,或者关联及泛化关系的识别。类对象的实例就映射为表的各行。见图 1。

(2) 对象标识 OID 的映射

在系统中唯一地标识一个对象非常重要。可以将对象标识映射为表的主键^[4]。见图 1。

(3) 属性类型的映射

UML 中的属性类型这一概念可以映射为数据库中的“域”。域的使用提高了设计的一致性,且优化了应用的移植性。

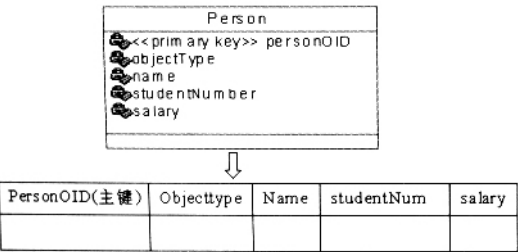


图 1 把类映射至单个表

3.2 将关系映射到关系数据库

UML 模型的对象之间的关系即关联、泛化、聚集等。本节

考虑如何把这些关系映射到关系数据库中。

3.2.1 关联的映射

关联表达的是对象之间的静态联系,即可用对象属性表达的一个对象与另一个对象之间的依赖关系。在 UML 模型中,关联有一对一、一对多和多对多三种类型。在关系数据库中,表与表之间也存在同样的关系,这种相似性是将关联映射到关系数据库的基础。下面把关联分为两类进行讨论。

(1) 简单关联:一对一、一对多的关系的映射

在面向对象方法中,简单关联可用对象指针(或用对象标识)实现。在一对一的连接中,在被连接的两个类中选择其中一个,在其对象中建立一个指针类型的属性,用于指向另一个类中与它有连接关系的对象实例。在一对多的连接中,选择有多个实例的一方,在其对象中建立一个指针类型的属性,用于指向另一个类中与它有连接关系的对象实例^[2]。

在关系数据库中,这种关系是用外键来实现的。一对一的关系比较少见。在一对多的关系中,选择有多行(多个对象实例)与另一方相对应的表,在其中建立一个外键指向另一方的主键。因此,把关联映射到数据库时,表的外键就相当于指向另一个对象指针。如图 2。

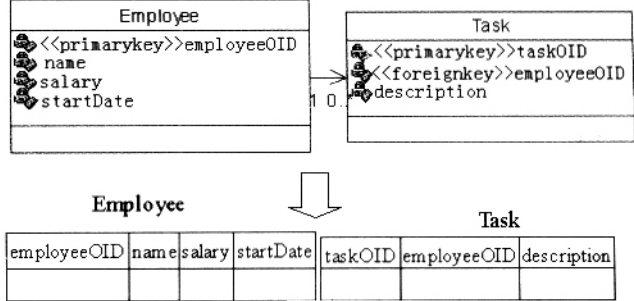


图 2 一对多关系的映射

(2) 复杂关联:多对多的关系的映射

在面向对象方法中,在两个对象间增加一个新的对象(称之为关联对象)来表示关联,使原来多对多的连接变成两个一对多的关系,以简化这种复杂连接。这种连接的实现方法是在表示关联的新对象中建立两个指针类型的属性,以分别指向原来的两个对象。

在关系数据库中,在两个表之间增加一个表,用于映射关联对象,从而也把原来多对多的关系简化为两个一对多的关系。这样映射方法就与简单关联一样了。实现时,在增加的一个表中,建立两个外键,用于分别指向原来的两个表的主键。

3.2.2 泛化关系的映射

泛化关系即反映了继承关系。继承是为了获得一般类的全部属性和服务,从而达到一种复用,简化人们对事物的认识和描述。

在面向对象的高级语言中,继承一般是通过“继承机制”来实现的。这种机制通过“自动地拥有”或“隐含复制”来达到复用的目的。

在关系模型中并没有明确的方法实施继承,因此在映射时,要么把所有的属性放到一个表中,要么将它们分成两个或多个表。

(1) 一个表的方法

就是把一般类和特殊类的所有属性都放到一个表中。这种方法表达的概念不清晰,也不能体现泛化所带来的好处,即泛

化的实体属性可以被更多的实体所共享。

Q)多个表的方法

这种方法是把一般类和特殊类分别映射到对应的 (不同的)表。在代表一般类的表中建立一个候选键 ,以标明该对象实例的特殊类别 ,例如是一般雇员还是经理。在代表特殊类的表中建立外键 ,这些外键都指向一般类的表的主键。通过这种方式 ,特殊类可以“继承”一般类的属性。

这种关系的映射方法与一对多的关系的映射方法相同。但它们表达的语义是不同的。

应该注意的是 ,在关系数据库处理泛化关系的能力是有限的 ,并没有明确的方法表达“泛化”的意思。每个表事实上是一个独立的结构 ,表之间无显式连接 ,需要被逻辑指明。从这个意义上说,“继承”不是自动的 ,即对使用者不是透明的 ,在访问时需要注意表间的连接。

3.2.3 聚集关系的映射

在面向对象技术中 ,这种关系的实现有两种方法。一种是嵌套的方法 ,一个对象嵌套在另一个对象内 ,用于实现紧密的聚集关系。另一种是指针方法 ,在一个对象中建立一个指向另一个对象的指针 ,用于实现松散的聚集关系。指针方法看起来与关联中一对多的关系相同 ,但在逻辑上它们所表达的语义是不同的。

把这种关系映射到关系数据库中 ,也有两种对应的方法 :对于紧密的聚集关系 ,可以在一张表中实现。对于松散的聚集关系 ,与一对多的关联映射方法一样。

4 把操作映射到关系数据库

关系和对象思想之间在理论上一个主要的不同是 :在关系数据库领域 ,数据和应用是分开考虑的 ,而在对象思想中 ,合起来考虑数据和与它关联的操作。数据对象和它们各自的操作与方法结合起来考虑的面向对象的方法 ,是一种更自然、更符合逻辑的方法。利用这种建模方法 ,开发与维护速度可以期望得到改进。对象有操作 ,这是强调了模型的计算能力。

UML 用动态模型表达模型的计算能力。要把动态模型的所有概念都考虑进来非常复杂而且也没必要。这里主要考虑在数据库应用中最关心的对象或对象属性的遍历问题。

遍历就是用穷尽枚举的方式在对象内部或在对象与对象之间查找想要得到的信息。对数据库而言 ,遍历表达式可以直接映射为 SQL 代码^[1]。

UML 的对象约束语言 (Object Constraint Language)OCL 可以用于描述这种遍历表达式^[5]。其中的点符号用于描述从对象到对象 ,或从对象到属性的导航。方括号用于描述施加于一个对象集的过滤。表 1 是一个例子 ,在一个有关航线、航班和飞

行员的数据库中遍历有关内容。在 SQL 语句中 ,变量的前面附加了一个冒号。

表 1 遍历表达式向 SQL 语句的映射

遍历表达式	含义	SQL 代码
anAirline.Employee	给定一架航班 ,查找相应的机组人员	SELECT employeeID FROM Employee WHERE airlineID= :anAirline ;
anAirline.Employee.name	给定一架航班 ,查找相应的机组人员姓名	SELECT name FROM Employee WHERE airlineID= :anAirline ;
anAirline.Flight [getMonth (late)=aMonth]].pilot	查找在给定时间内飞行一架航班的飞行员	SELECT pilotID FROM Flight WHERE airlineID= :anAirline AND getMonth (late)= :aMonth ;

5 结束语

通过上述分析可以看到 ,为了在数据库领域利用面向对象技术带来的益处 ,可以采用 UML 得到面向对象的数据模型。UML 模型与关系数据库之间是存在差异的 ,通过该文探讨的一些技巧性的处理 ,UML 模型可以映射到关系数据库。可以把这种方法称为对象-关系方法。

采用对象-关系方法解决数据库应用问题 ,可以得到以下好处 :能够以更快的速度和更少的花费创建数据库 ;能够更容易地、以更少的花费维护数据库 ;系统将会更加灵活和健壮 ,因而事务中的变化或新发现的需求能够更容易地、以更少的花费被修改。这正是对象-关系方法的前景所在。

在 UML 模型向关系数据库映射的问题上 ,还有许多课题可以讨论 ,例如参考一致性的问题、触发器问题、存储过程问题和数据库访问接口问题等。这些都是有待于进一步研究的。

(收稿日期 :2001 年 9 月)

参考文献

1.Michael Blaha ,William Premerlani.Using UML to Design Database Applications[M].Rational Software Corporation ,Monterey California ,1999
2.邵维忠 ,杨美清.面向对象方法[M].广西科学技术出版社 ,清华大学出版社 ,1998
3.Paul Dorsey ,Joseph Hudicka.Oracle 8 Design Using UML Object Modeling[M].McGraw Hill ,1999
4.Ben Ezzell.MCSD :Analyzing Requirements and Defining Solution Architectures 学习指南[M].北京 :电子工业出版社 ,1999
5.The Unified Modeling Language for Object-Oriented Development Documentation v 1.1[M].Rational Software Corporation ,Monterey California ,1997
6.OMG Unified Modeling Language Specification[M].Version 1.3 ,Object Management Group 2000

(收稿日期 :2001 年 9 月)

参考文献

1.[美]Matt Pietrek 著,侯俊杰译.Windows 95 系统编程奥秘[M].松岗出版社 ,1997
2.[美]Jeffrey Richter 著,王建华等译.Windows 核心编程[M].机械工业出版社 ,2000
3.[美]Peter Norton 著,孙凤英等译.MFC 开发 Windows 95/NT4 应用程序[M].清华大学出版社

(上接 146 页)
个简单的用户界面显示结果。此程序的代码比较长 ,在此不做详尽的分析 ,有兴趣的读者可以向笔者发 EMAIL 获取源代码。

6 结束语

总之 ,Windows API 拦截是一项涉及到 Windows 操作系统内核的技术 ,它就好比是一扇通往 Windows 神秘内部世界的大门。笔者通过对 API 拦截技术长期研究和多次试验得出了总结性综述 ,希望能对广大科研人员有一些启发。