Computer Engineering

December 1999

开发应用与经验交流。

文章编号: 1000-3428(1999)12-0094-02

中图分类号: TP315

基于UML的管理信息系统开发

张卫山, 巫家敏, 严新民

(西北工业大学CIMS研究所, 西安 710072)

摘 要:以一个管理信息系统为背景,探讨了利用统一建模语言对系统进行面向对象分析、设计以及实现问题,顺利实现了对象模型向关系 数据库模型的转换。

关键词:统一建模语言;面向对象分析;面向对象设计;面向对象实现

Development of MIS Based on UML

Zhang Weishan, Wu Jiamin, Yan Xinmin

(CIMS Research Institute of Northwest Polytechnic University, Xian 710072)

[Abstract] In this paper, we base our research on a Management Information System (MIS), and discuss OOA(Object-Oriented Analysis), OOD(Object-Oriented Design) and implementation of the OO(Object-Oriented) models by utilizing UML. The mapping from the object model to the relational database model is carried out smoothly. The practice manifests that UML can elegantly express the whole developing process with flexibility and consistency.

[Key words] UML; OOA; OOD; Implementation of OO models

1 UML基础及基于UML的系统开发过程

统一建模语言UML(Unified Modeling Language)^{[1]-[3]} 代表面向对象技术的最新发展,它统一了一些主要的面向对象方法,如Booch,OMT和OOSE等,融合了当今技术发展的成果以及未来的发展趋势,例如组件技术、分布式计算等。UML为软件系统的可视化分析与构建以及文档的产生,提供了一致的语言。UML能够满足如下的需求:

- 1) UML通过提供公共对象分析和对象设计元模型(OA &D)的正式定义,来表达OA&D模型的语义,其中包括静态模型、动态模型和体系结构模型。
- 2) 对象分析和对象设计(OA&D)工具所产生的模型之间 的交互机制提供接口定义语言(IDL)规范,支持用户模型的 动态构建以及遍历。
- 3) 提供通俗易懂的标记来表达OA&D模型,UML标记 法能够对UML语义提供一致的完美图形表达。

UML中通过提供不同形式的图形来表达从软件分析开始的软件开发全过程的描述。主要的图形有以下几种:

- 1) 静态结构图 用于描述模型的静态结构,不表示时间上的信息,着重描述系统中内部结构以及系统中存在的事物和事物之间的相互联系。其中主要是类图(class diagram),对象图(object diagram)是类图的实例。对象具有生命周期,因此,对象图只能在系统的某段时间内存在。
- 2) 包(package) 包由一组模型元素组成,可包含从属 包或普通的模型元素。所有的UML模型元素和图形可以组 织成为包。
- 3) 用例图(use case diagram) 从用户的角度对系统的 功能需求进行描述,表述系统中执行者与用例之间的关系。 用例图驱动需求分析之后各阶段的开发工作。
- 4) 顺序图(sequence diagram) 顺序图显示对象之间在时间和顺序上的动态合作关系,它强调对象之间消息发送顺序,同时显示对象之间的交互。顺序图不表达对象之间的关联关系。

- 5) 合作图(collaboration diagram) 合作图描述对象之间的交互关系以及对象之间的联系。合作图并没有将实践作为单独的一维进行显示,所以必须用序列号来确定消息发送和并发线程。
- 6) 状态图(statechart diagram) 状态图描述对象或交互 在生命周期中对激励所做出的反映以及动作,事件发生时状 态转移的条件。
- 7) 活动图(activity diagram) 活动图是状态机的一种变形,其中的状态是活动,表示操作的激发以及完成是由于操作的完成而触发的。
- 8) 实现图(implementation diagram) 实现图显示实现时的状态,包括系统源代码的结构和运行时刻的实现结构。实现图包括组件图 (component diagram)和配置图(deployment diagram)。

• 组件图

组件图描述软件组件之间的代码组件的依赖关系,包括源代码组件、二进制代码组件、执行组件。组件可存在于编译、连接、运行时间,可以在不同时间多次出现。组件包括接口,因此可以用来描述组件之间的接口关系和调用关系。

・ 配置图

配置图显示运行时刻处理元素的配置,以及依赖于它们的软件组件、过程、对象。配置图定义系统中软硬件的物理体系结构,它可以显示实际的计算机和设备以及它们之间的连接关系(用节点表示),也可以显示连接的类型及组件之间的依赖性。

在UML中没有提供过程的概念,用户可以根据自己的 需要定制适合自己需求的过程,但是一般要遵循面向对象开 发的一般过程,即迭代式开发、分阶段提交,以用例驱动整

作者简介: 张卫山(1970~), 男, 博士生, 主研方向: FMS的建模与仿真, 管理信息系统的开发与研究, 软件工程

收稿日期: 1999-04-12

-- 94 --

个开发过程。以UML为基础进行整个开发过程与一般的面向对象方法并没有本质的区别,因为UML本身就来源于一些比较先进的面向对象的建模方法,只不过UML本身更能够满足用户的表达需要,能对组件技术,分布式计算等进行精确描述,适应未来软件工程发展的要求。基于UML的开发在每一次迭代过程中沿着面向对象的分析、面向对象的设计、面向对象的实现,软件测试与交付进行。

2 基于UML的管理信息系统开发过程

利用UML所提供的各种图形表达,可以对从需求分析 开始的整个开发过程进行描述。结合某劳资处管理信息系统 (以下称LZC-MIS)的开发实例进行叙述。

2.1 面向对象分析

首先是利用用例图捕捉劳资处对本系统的用户需求,对于重要的以及复杂的用例,为了表达清楚,可以使用活动图等进行辅助描述,对于一些限制条件的书写,应遵循UML的约定[4]。

然后,分析LZC-MIS所涉及的对象,找出系统中的对象类,以及各对象类之间的关系。确定对象类的方法可以参考文献[5],这个过程主要是通过类图来完成的。当然,这一步所确定的类基本上还只是一个框架,其操作还有待进一步细化。UML提供了顺序图、状态图、活动图、合作图等来描述对象类的动态行为。

2.2 面向对象的设计

在面向对象分析的基础上,综合利用UML提供的各种 图形,对LZC-MIS进行设计。

・系统结构设计

利用UML的包进行LZC-MIS的高层设计,这个过程就是定义包与包之间的关系的过程。软件系统要符合高内聚低耦合的原则,所以各个包之间的关系应该简单明了,减少包间的依赖关系,尤其是双向依赖关系。考虑到所选用的数据库服务器Oracle及前端开发工具Powerbuilder,对本系统定义如下几个包:

- 1) 用户接口包: 用户直接同用户接口包打交道, 通过该包访问系统中的数据。包括窗口类、用户对象类、数据窗口类等。
- 2) 数据接口包:定义系统数据与用户之间的接口,包括各种视图类。
- 3) 劳资处对象包:面向对象分析中所产生的对象类, LZC-MIS中主要是职工类、离退职工类、在职职工类、临 时工类、工伤保险类、养老保险类、养老金类、调整文号 类、工资类、单项奖类、综合奖类等等。有关系统权限定义 而产生的对象类,也包含在该包中。

・详细设计

从实现的角度对系统进行更详细的描述,可以利用 UML提供的各种图形,如类图、状态图、合作图、活动 图、顺序图等,对分析阶段产生的劳资处对象类进行扩展和 细化。在确定和细化对象类的操作时,可以按照客户机/服 务器方式进行操作的认定,分为以下3类:

1) 基础操作 维护对象自身的存在所必须具有的操作, 例如, 最常见的增加、删除、修改等。

- 2) 客户操作 一个对象为获取其它对象的状态信息而发出的请求操作。
- 3) 服务操作 一个对象为应答客户请求而做出的操作。 2.3 面向对象实现

选择的数据库服务器是Oracle7,它是一种关系型数据库系统(RDBMS)。关系模型中,数据在用户观点下的逻辑结构是二维表,其中的每一个分量是不可再分的原子数据。而对象由属性和操作组成,所以关系型数据库系统不直接支持对象的存取。而对象模型最终由二维表及表间的关系来描述,所以对象模型与关系模型的转换成为面向对象实现的必不可少的一步。为此,提出了如下的转换模型,见图2。

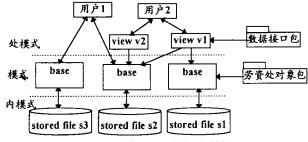


图2 对象模型到关系模型的转换

数据接口包中的各种视图类,映射成为关系数据库系统的外模式,这相对简单一些。不过有些视图的定义可能现在还不好确定,要在具体开发过程中才能最后确定。关键是劳资处对象包到模式的映射,实际上是类图中的每个类到基表的转换过程。在转换的过程中,需要遵循以下原则:

- 1) 对于类图中的每个类, 视不同的具体情况, 可以映 射成一个或多个表, 类中的属性与表中的属性相同。例如, 对于工伤保险、养老保险, 都各自映射成两个表, 包括减少 表、增加表。
 - 2) 要分清类之间的关系, 然后进行具体映射。

当类之间存在泛化关系(generalization)时,父类和子类可各自都映射成基表,也可以将子类表的属性添加到父类表中,而不建立子类表,也可以不建立父类表,而只有子类表。以上几种情况可以灵活处理,结合起来使用。例如,对于职工类、离退职工类、在职职工类、临时工类之间的泛化关系,将离退职工、在职职工的固定不变的那部分信息建立在一张表emp中,将经常变动的信息另外建立一个表emp_chan,退休职工的特殊信息建立一个表retire保存,临时工的信息建立临时工信息表temp。

- · 类之间存在聚集关系时,可以按与继承关系同样的 方法进行处理。
- · 类之间存在关联关系,应视具体情况,进行相应的转换。可以将关系转换成相应的外键,例如,对于职工类单位类映射到关系表时,只需在职工变动信息表中增加外键dwdm,表示此时职工所在单位的单位代码。如果类之间的关联关系涉及到别的对象,例如,职工与单位之间有工作合同,则应将工作合同这一关联映射成一个关联类工作合同类,将此类映射成工作合同表。
- 3) 将得到的表进行冗余控制调整,使其达到3NF范式的要求。

按照以上几条规则,将劳资处对象包中的对象类,映射成数据库中的基表。 (下转第107页)

string v_copies i_dwname=message.powerobjectparm st_1.text='打印机: '+string(i_dwname.object.Datawindow. printer) v_copies=string(i_dwname.object.Datawindow.print.copies) cbx_2.checked=(upper(string(i_dwname.object.Datawindow. print.collate))='YES') i_savename=trim(string(i_dwname.object.Datawindow.print. filename)) cbx_1.checked=(i_savename<>") (5) Cb_3 Clicked事件的Script printsetup() st_l='打印机:'+string(i_dwname.object.Datawindow.printer) (6) Cb_1 Clicked事件的Script integer nindex string v_modify,v_page,v_return v_modify=' Datawindow.Print.Copies='+em_1.text

> v_modify=v_modify+' Datawindow.Print.Collate=No ' end if

v_modify=v_modify+' Datawindow.Print.Collate=Yes '

If cbx_1.checked then

If cbx_2.checked then

v_modify=v_modify+' Datawindow.Print.FileName='+i_ savename

else

else

v_modify=v_modify+" Datawindow.Print.FileName=" " end if

If rb_1.checked then

v_modify=v_modify+" Datawindow.Print.Page.Range=" " elseif rb_2.checked then

v_page=i_dwname.describe("Evaluate('Page()',"+string(i_ dwname.getrow())+')')

v_modify=v_modify+" Datawindow.Print.Page.Range=""+ v_page+" "

(上接第95页)

接下来的工作,就是充分利用前端开发工具的面向对象 编程特性,例如,继承、封装等,进行面向对象编程工作。 Powerbuilder主要是通过事件来封装对象功能的。具体的编 程在此不加以叙述,可以参考文献[6]。

2.4 软件测试与交付

完成系统编程以后,需要对系统进行测试,通常包括单 元测试、集成测试、系统测试和验收测试。在单元测试中使 用类图和类的规格说明,测试一个类或一组类。集成测试和 系统测试中使用组件图、合作图、用例图,进行测试是否能 满足用例的要求。系统的交付,要求提交完善的文档,并要 对客户提供相应的培训工作。

3 结论

以上利用UML进行了一个实际管理信息系统的开发工 作,UML能够对整个开发过程提供灵活、一致、易读的表 达,显示出UML的魅力。UML融合了各种建模语言的优良 特性,为面向对象系统的开发,软件生产自动化工具与支持

else

v_modify=v_modify+" Datawindow Print Page Range=""+ sle_1.text+"' "

end if

v_ddlbnum=ddlb_1.finditem(ddlb_1.text,0)

v_modify=v_modify+"Datawindow.Print.Page.RangeInclude ="+string(v_ddlbnum -1)

i_dwname.modify(v_modify)

parent.visible=false

i_dwname.print(true)

this.setfocus()

close(parent)

(7) 调用W_1的Script

integer v_count,v_tcount

Datawindow dwc

v_tcount=upperbound(parent.control[])

for v_count=1 to v_tcount

choose case parent.control[v_count].typeof()

case Datawindow!

dwc=parent.control[v_count]

end choose

next

openwithparm(w_1,dwc)

3 结束语

PowerBuilder环境中的Datawindow具有很强的灵活 性, 能较快地生成不同风格的报表, 如单纯用打印函数打 印, 实际应用效果不好, 如果采用上面介绍的方法可以使打 印更具灵活性,当然还可以利用其它Datawindow的Print特 性、让用户具有更大的自由度。

- 1 PowerBuilder Documentation(随机资料)
- 2 何军, 刘红岩. PowerBuilder5.0原理与应用开发指南.北京: 电子 机械工业出版社,1997

环境的建造提供了一个丰富、严谨、扩充性强的表达方式。 将UML与软件重用的研究以及软件开发过程的自动化工作 相结合,具有非常广阔的应用前景,这方面的工作在我国刚 刚开始。

参考文献

- 1 Rational Software Corp. UML Summary. URL: http://www. rational.com/uml
- 2 OMG Unified Modeling Language Specification(draft). URL: http://www.omg.org/uml, 1999.1
- 3 Rational Software Corp. UML Notation Guide, URL: http:// www.rational.com/uml
- 4 Rational Software Corp. Object Constraint Language Specification. URL: http://www.rational.com/uml
- 5 方文,何建敏.消防MIS的面向对象分析.东南大学学报, 1998, 28(4): 54-58
- 6 [美]Gallaghar S, Herbert S. Powerbuilder 6.0程序设计大全. 北 京: 机械工业出版社, 1998