

面向对象的管理信息系统开发研究

曾慧城 (重庆钢铁高等专科学校 400053)

周庆忠 (重庆后勤工程学院 400016)

摘要:本文通过面向对象的管理信息系统开发研究,对OOSE在软件开发过程中的应用进行分析,论叙OOA、OOD和OOP方法与技巧,阐明OOSE对缩短软件开发周期,提高系统性能具有重要作用。

关键词:对象 类 OOSE OOA OOD OOP

引言

大型油料管理信息系统开发的复杂性及对质量和可靠性的严格要求,需要行之有效的技术措施来保证。传统的结构化软件开发方法面向过程,其Structure Analysis、Structure Design和Structure Programming以不同的模型作为开发基础,按照数据变换的过程寻找问题的结点,易造成功能模块的细分因对过程的理解差异而不同,各模块与数据库逻辑模式间无映射关系,程序与数据结构难于封装在一起。因而结构化软件开发方法日渐萎缩,面向对象软件工程方法(OOSE)则脱颖而出。

OOSE的基本单元是对象,它封装了与其有关的数据库和相应处理方法,从问题空间到解空间同态映射。其任务即是要找出应用领域中与用户需求相关的对象,识别出对象在系统中的属性、所具有的行为及各对象间的联系,将它们用语言机制表示出来,从而开发出与用户需求一致的、完整有效的软件系统。OOSE模拟人认识客观世界的过程,分为面向对象分析(OOA)、面向对象设计(OOD)、面向对象编程(OOP)三个阶段。OOA和OOD是整个过程的重心,其重要性随系统的复杂性而递增,OOP则借助相应语言机制,实现各对象预定任务。三个阶段联系自然,体现了抽象、信息隐蔽、模块化开发原则,减少了用户与开发者间的通信障碍,缩短了开发周期,提高了软件质量。

在工作实践中,我们应用OOSE完成了多项大型软件系统的开发任务,受益匪浅。现以油料保障决策系统MEOSDS为例讨论OOSE工作的全过程。

一、面向对象分析 OOA

需求分析是软件开发生命周期的关键一步。OOA即是在系统调查资料基础上,对面向对象方法所需

的素材进行归类分析和整理。它建立在对象及其属性、类及其成员、整体及其部分等概念之上,以对象及其交互关系为手段,将非形式化的需求说明表述为明确的软件系统需求。面向对象分析模型从对象模型、动态模型和功能模型三个侧面进行描述,主要肩负三大任务,其一:通过对问题空间的分析,识别出问题所涉及的对象、对象间的关系和服务,建立对象模型;其二:以对象模型为基础,完成相应需求描述;其三:对需求描述进一步作需求评审。OOA步骤为标识对象、标识结构、定义属性和定义服务。

1. 标识对象

对象是实际问题域中有意义的概念实体,具有目标软件系统所关心的属性,并与系统中其他有意义的对象进行消息传递和提供外部服务。标识对象即为找出应用中相对独立的实体,也即是把相关数据及操作组织在一个对象内。

在油料保障决策系统中,有众多与应用问题直接对应的对象(如领油单位、发油单位、油品等),还存在某种由处理方式或约束条件间接给出的对象(如收支类别、指标类型等)。如油料收发证件处理子系统涉及以下对象:

收发单位:名称、隶属单位、时间、收发依据;

凭证:名称、凭证字号、日期、数据;

油品:名称、编码、密度、价格、体积、数量、种类、类别(实物/价拨);

业务部门:名称、级别、隶属单位、负责人、经手人、承办人。

其处理模型由一个外部系统(凭证)、一个事件(收发单位)、一个扮演的角色(油品)和一个组织单元(业务部门)构成。

2. 标识结构

结构在问题空间里表示复杂性。在分类结构中,对

象共享在它之上定义的服务,也可增加它所继承的属性,从分类结构可获得类—对象、父类—子类的组织形式。而组装结构则刻划整体一部分的关系。标识结构即考虑对象间的关系,以简化问题的分析和描述。例:图1(a)所示业务人员与负责人/经手人/承办人/填表人等为分类结构;(b)所示凭证与凭证字号/油品数据/收发单位等是组装结构。

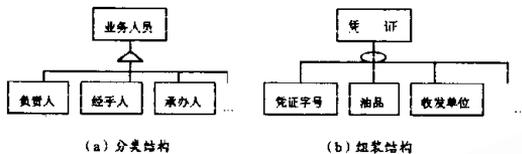


图1 对象结构图

3. 定义属性

属性是描述对象实例的数据单元。定义属性即是考虑对象所涉及的数据,检查问题空间,将现实中的对象与实际描述该对象的属性联系起来。属性的取值决定对象所有可能的状态。定义属性分识别属性、定位属性和指定属性三步进行。

识别属性 属性通常对应于用户需求描述中的名词或形容词,它在问题陈述中并非均有完整的显式描述,潜在属性的发现有赖于开发人员对领域知识的深刻理解。如显式属性有业务部门、收发油单位、凭证、收发油数等,潜在属性有用户姓名、凭证唯一标识号等。

定位属性 直接将各属性放到客观存在最合适的对象中,在分类结构中公共属性放在结构的高层,特殊属性放在底层。适用于大多数特殊情况的属性将其放在通用位置。例如,油料收发作业的属性有日期、负责人、收支类型等,这些属性可被收发单位和油品所继承,同时收发单位又有自己的属性:名称、编号、隶属单位,油品也有自己的属性:名称、编码、密度、价格和种类等。

指定属性 用标准词汇规范属性的命名,力求含义准确、易读。

4. 定义服务

服务即为收到消息后所执行的处理。定义服务即是考虑对数据所需的操作,其中心问题是为每个对象定义所需要的行为。标识服务的基本策略有增加、修改、删除、选择、计算和统计等服务,辅助策略有生命历程和状态—事件—相应。例如,凭证类的服务是产生库存总帐、

收支帐、收支明细表和决算表等;油料档案类的服务是产生油品名称、编码、密度和价格等,且凭证类与油料档案类之间有消息传递,凭证类调用油料档案类中的油品名称、编码和价格等服务。

定义服务的过程是:从用户需求描述入手,找出其中的动词。判断是否构成某个对象的合适操作服务,如“增加/删除一种油品”可视为“油品”对象的操作服务。还可通过对对象生命周期、对象应展现的计算性行为的分析,寻找潜在操作服务。对象的行为由消息传递引发,消息是对象间相互作用和语言的唯一途径。消息由发送对象传向接收对象,接收对象收到消息后,为发送对象提供相应服务。例如,在支拨单修改模块中,“编辑”按钮对象为消息发送者,与支拨单数据相关的库存总帐、收支帐、收支明细表和决算表等对象为消息接收者,“编辑某一油品数据”的消息由“编辑”按钮对象发出,传向各接收对象,各接收对象收到编辑消息后,对各自数据作相应处理,提供相应的查找、定位、更改、删除、添加、重新排序、汇总等操作服务。在系统需求文档中,对象及其服务等描述应简洁无歧义,用词准确,格式统一。例:“油品”对象的需求文档为:

对象:油品;

属性:名称;

编码;

种类(汽油/煤油/柴油/...);

密度;

价格;

油料类别(主油/附油)

指标类型(实物油/正常价拨油/专项价拨油);

油品的状态(多出/短少/...);

油品的唯一标识号;

.....

服务:收油/发油/结转/代储/代供/...;

添加/查找/更改/删除;

排序/汇总/统计/...;

找出最近一次收(发)油单位的信息(如名称、隶属单位、负责人);

.....

二、面向对象设计 OOD

从 OOA 到 OOD 是一个逐渐扩充模型的过程,OOA 模型反映问题域和系统任务,OOD 模型则进一步反映需

求的一种实现,即在 OOA 模型中,根据所应用的开发环境功能的强弱程度,填入和扩展有关实现方面的软件设计信息。OOD 工作内容主要有:主体部件 PDC 设计和数据管理部件 DMC 设计。

1. 主体部件设计

PDC 任务是结合使用的开发语言、工具对 OOA 模型作适当改造与调整,重用软部件,引入新类或对象,拆分原有类和对象,转换继承结构以适应程序语言对实现的限制,其中重用软部件是 PDC 的主要内容。

对象是天然的可重用单元,对共性对象只需开发一次,它便可在系统内部及外部重用。重用依存于超类设计,超类设计效果好坏主要取决于其内部重用率的高低。在超类设计时,应力求发现实例对象的共同行为,如“打印报表”、“组合模糊查询”和“操作权限验证”等。超类的多态性设计应使用超类继承关系,以满足各子类的操作要求,如在支拨单、收发证件、指标转供单等凭证数据录入模块中,继承同一个“数据录入”祖先窗口来实现不同结构的库表数据录入。

超类设计应在 OOD 阶段完成,以便在后继的 OOP 阶段有共同继承的祖先对象供开发者使用。在后期开发过程中,仍需根据特化过程的反馈信息,对超类对象作相应调整和修改。这要求超类设计员应精通设计与编程,在 OOD 乃至 OOP 阶段均担负设计、实现和修改的责任。

2. 数据管理部件设计

数据管理部件定义专用对象—数据库,将目标系统依赖于开发平台的数据存取操作与相关的存取机制分离,以提高对象独立性。显然,数据库设计是 OOD 最重要的设计环节,将系统对象模型向数据库模式映射则是面向对象数据库设计的关键。

因应用系统的各实体(如油品、发油单位、收油单位、业务部门、油桶、油罐、指标、库存状况及收支情况等)之间客观存在着复杂的多重关系,用面向对象方法进行数据库设计,使系统结构与数据库结构相互映射,形成多重继承结构,避免构建出系统结构与数据库结构脱离的网状结构,减轻数据库前端应用编程和数据库维护的负担,使数据库结构清晰简单,数据库对象独立性、自治性强,重用率高,便于实现 OOP。

换言之,对象模型向数据库模式的映射即是向数据库表的变换过程。步骤为:①将各种对象类中定义的所有属性表示成库表的形式,一个对象类可映射一个或多个库表,当类间有一对多关系时,一个表也可对应多个类;②对映射后的库表进行冗余控制调整,使其达到合理

的关系范式;③将每一范式定义成一个关系,关系的映射有一对一、一对多、多对多以及三项关系等多种情况;④对存储限制和其他性能作出评估,必要时修改范式等级。⑤进行数据操作定义。

例:在油料业务管理子系统中需处理大量的凭证、帐目和报表等资料,分析油料库存月报表、油料收支情况月报表和油料指标使用情况月报表等众多报表,可知每月结算上报的月报表是同构的,可创建一张库表存储同一种报表。各下属单位的上报表中均对应各自的上报日期、单位名称、负责人、填表人。库表标识主键为上报日期和上报单位。为避免在库表的每条记录都加上这两个字段,而造成库表冗余度加大,操作效率降低。可设计超类对象基码表和作业码表。基码表和作业码表的表结构如图 2 所示:



(a) 基码表表结构

(b) 作业码表表结构

图 2 基码表和作业码表的表结构图

将它们加入由系统对象模型映射出的数据库模式后,得到图 3 所示的结构。对应各应用模块对象建立一张作业码表,同一类报表属于同一作业码表,由作业码表统一管理。作业码表给各下属单位上报的每一张报表分配一个作业码。作业码的数据类型为 Char(10),前 4 位为表号,后 6 位为序列号,取自基码表中最大序列号。作业码是报表对象的唯一标识符。它在这张报表的生存期

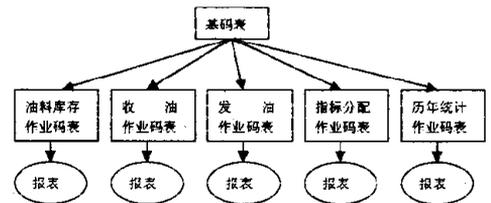


图 3 超类对象间关系示意图

内具有永久不变性,不因报表内容和结构的更改而变化,

只随报表的删除而作废消失。此外,该作业码在整个数据库中的唯一性也便于在库中对任何一张报表进行查询、编辑等操作。作业码表是父表,报表是子表,通过基码表建立父子表之间的关联。这样设计使得各应用模块对象完全映射到数据库模式中,形成应用对象与数据库对象的一一对应,应用对象独立性好,具有较好的封装性和信息隐蔽性。

安全性设计是数据库设计的重要部分。可用失效模式及效应分析方法(FMEA)、潜在危险与可操作性分析(HAZOP)等技术作潜在危险识别,进行因果关系分析,构造故障分析树,针对所发现的原始故障采取相应的防范机制。如对各子系统设计一个局部数据库,用于存放在一定时间范围内相对独立的数据,既可实现空间上的分布性,又可减少各部分间的耦合度,降低故障对全局的影响力。再提供一个中心数据库作为各子系统局部数据库的备份,定期保存系统中的各种数据。当某子系统的局部数据库被破坏时,可用中心数据库的备份进行数据恢复,达到减灾目的。

面向角色的动态功能组合也是有效的安全性措施之一。建立用户信息表、角色表和功能表,通过用户-角色-功能的逻辑控制数据,对不同角色身份的用户按其拥有权限生成界面表示级的功能集成,阻止对数据库的非法访问。

三、面向对象编程 OOP

OOP任务是实现OOD预定各对象应完成的功能,分为可视化设计和代码设计两阶段。可视化设计阶段主要是进行用户界面设计,将系统所有功能与界面中的控制或菜单命令联系起来,即在某一界面对象(如表单)上集合功能所需的控件对象(如按钮、编辑框、标签、组合框、库表等),设置各对象属性,布置窗口。OOP为创建应用程序展示了一个新境界:一旦某一对象放入窗体后,其特征便自动转换为程序代码。在程序执行前,对象已呈现在屏幕上,犹如运行时一样可视地工作,直接观察到效果,这对程序设计员极富吸引力。应注意使用屏幕界面模版超类的继承关系,结合特化处理,统一界面风格,如报表、按钮和窗口大小一样,色调一致。

代码设计阶段的主要任务是为对象编写所需要响应的事件代码,为对象发挥必要的功能,建立不同对象间的

正确连接关系。常用事件驱动方法来管理程序和用户间的交互行为,程序对不同事件所作出的不同反应需要开发者为之编写代码。事件与代码之间通过事件句柄相联系。每类对象都有响应特定事件的能力。可见,欲对OOP游刃有余,必须熟知对象的属性、方法,深刻理解事件驱动的内涵,这正是OOP的关键所在。

OOP支持多态性,允许子类利用内部定义功能代替父类中同名的子程序,这便于处理不同媒介对同一操作做出不同反应的问题。例如同样是一个Open操作,文本、图像、声频/视频信息对其反应则不同。将父类中与此事件对应的函数体以虚函数定义,在不同子类中根据所代表不同信息种类,利用同名函数对其进行覆盖,即可实现多媒体处理中同一事件不同反应的模式,明显提高了代码生成效率。

提高程序的可重用率是OOP的重要内容之一。分离静态数据,设计面向对象数据库的静态表系列,是保持程序和数据库表结构与内容的稳定性,提高程序的可重用率的有效措施之一。程序是静态数据的最大来源,应将静态与动态数据间的控制关系抽象为静态表与动态表间关系,用数据库存放静态数据,使程序脱离数据的存储方式。静态表应由系统管理员统一维护,尽量少用外键,对访问频繁的静态表,设计通用查询函数做为程序接口。例:油料部的各下属单位每年年底上报一次数据,每类数据理由数据录入、查询和打印和修改模块完成,“单位名称”是四个模块的共同选择控制数据,将它做成一张静态表存入数据库,由各模块向系统发出操作命令,再反馈操作结果给模块,因数据库提供了很好的共享和并发处理机制,便于实现数据共享。当更换、删除、添加单位时,只需修改“单位名称”静态表的数据,不必修改程序。

四、结语

OOSE以拟人思维方式去理解现实世界对象的实质,其OOA、OOD和OOP三个阶段均以对象为研究出发点,从问题空间到解空间同态映射,自始至终贯彻面向对象思想,克服了传统结构化思想的弊端,给软件开发工作带来了勃勃生机,显示出强大生命力和天然合理性。

(来稿时间:1998年9月)