

基于OM T 的快速原型法及其在决策支持系统开发中的应用

孙龙清

(中国农业大学计算机网络中心)

摘要 结合决策支持系统——最低限制价格计算系统 PRCSYS01 的研究和开发过程,探讨了以OM T (Object Modeling Technique)方法作指导,将系统界面的开发与系统内部功能的开发分离,快速开发用户驱动需求分析原型系统的方法。

关键词 决策支持系统; 面向对象; OM T; 原型系统; 快速原型法

中图分类号 TP 311.52

Rapid Prototype Method Based on OM T and the Application of Developing the Decision Support System (DSS)

Sun Longqing

(Computer Network Center, CAU)

Abstract With the guidance of OM T method, separating the developing method of system interfaces and the system internal function, the rapid prototype method was discussed in detail through a real example of development of Decision Support System (DSS)——the lowest limited price calculation system PRCSYS01.

Key words Decision Support System (DSS); object-oriented; OM T (Object Modeling Technique); prototype system; rapid prototype method

“决策支持系统(Decision Support System, 简称DSS)”是以支持非结构化与半结构化的决策问题为本质特征的。DSS 这种新技术一出现,便引起了人们的高度重视,特别是在美国和日本对于DSS 的应用研究比较广泛和深入。与其他软件开发所面临的问题一样,在系统的开发和维护方面DSS 也存在着许多的问题,主要有: 1) 系统分析人员与决策者(特别是高层决策者)之间缺乏必要的沟通,由于传统的系统分析方法对系统中人的因素和作用考虑不够或缺乏有效的手段去考虑^[1],因此,在DSS 的开发过程中,决策者不能很好地发挥其作用,使得开发的软件不能满足用户的需求; 2) 缺乏足够的灵活性,导致影响其实用性^[2]。主要表现在面对不断变化的复杂的决策问题时,难于重组模型、重构解决逻辑,与不断变化的需求不相适应; 3) 软件开发周期过长,而需求已发生了变化。究其原因,除了与DSS 本身的特点有关外,更重要的是传统的软件开发和维护方法存在着不可克服的缺点。

为了克服传统软件开发方法用于开发DSS 时存在的缺点,笔者根据DSS 本身的特点,结合用于招标时的决策支持系统——最低限制价格计算系统 PRCSYS01 的研究和开发过程,探

收稿日期: 2001-08-03

中日合作项目

孙龙清,北京清华东路17号 中国农业大学(东校区)53 信箱,100083

讨以OM T (Object Modeling Technique) 方法作指导, 将系统界面开发与系统内部功能开发分离, 快速开发用户驱动需求分析的原型系统。

1 基于OM T 的快速原型系统的建立

1.1 OMT 的建模特点

面向对象的程序设计(Object Oriented Programming, 简称OOP) 是一种新的软件设计方法, 应用该方法进行系统分析和设计, 可达到使人们的认识系统与计算系统一致, 问题空间与解空间在结构上一致的目的, 从而提高了软件的可重用性、可扩展性、易移植性和易维护性。

OM T 是美国通用电气研究发展中心概括多年应用OO (Object Oriented) 的实践经验而总结出的一种自底向上和自顶向下相结合的软件设计方法, 它以对象建模为基础, 不仅考虑了输入、输出数据结构, 而且也包含了所有对象的数据结构^[3]。OM T 全面支持OO 概念, 具有很强的应用性。在软件开发过程中, OM T 给出了最为详细的步骤, 尤其是详细规定了动态模型的建模过程。例如, 对于状态迁移图的制作, OM T 有“指示方案(脚本) 事件轨迹图 事件流程图 状态迁移图”, 这一连串的启发式说明和步骤, 从而增强了分析时的可操作性^[4]。在建模方面, OM T 将对象模型、动态模型和功能模型平等对待。OM T 支持软件开发中的分析、设计和实现的全过程, 并且对开发过程的描述比其他方法要详细清楚, 工作步骤具体, 建模能力强。

1.2 原型系统模型的确立

根据DSS 的特点选择高度迭代的动态方法——B ilb 递增模型^[5]。基本作法是: 经过初步调研和分析获知用户的基本需求, 利用适当的软件工具快速实现一个原型系统作为沟通各方的基础和用户实践的场所; 开发人员根据用户的意见, 对原型系统进行修改和扩充; 交给用户试用, 并根据试用后的意见, 再次对原型系统进行修改和扩充。这样多次迭代直到用户满意为止。实际上这是一种用户驱动需求分析的递增模型(图1)。

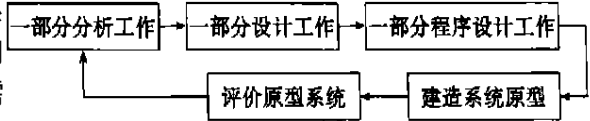


图1 Gilb的递增模型

1.3 基于OM T 的快速原型法模型

为了快速建立系统原型, 需要适当的软件工具。目前可视化的第4代语言比较流行, 且支持原型法。用面向对象方法所设计的软件具有良好的稳定性、可重用性、封装性、继承性、多态性和可维护性, 这些优点对于用快速原型法开发的软件来说, 是至关重要的。该方法避免了DSS 具有的运行不稳定、难于维护、移植困难和扩展升级麻烦等问题, 同时也保证了在面对不断变化的复杂的决策问题, 需要重组模型、重构解决逻辑时, DSS 有足够的灵活性。

一般来说, 在采用面向对象的快速原型法开发软件时, 面向对象方法的工作重点在分析阶段, 即在软件开发初期定义一系列面向问题的对象(而在由原型系统不断向目标系统演进的整个过程中, 不断充实和扩充这些对象)。虽然这些对象模型是在分析阶段得到的, 但也适用于设计阶段和实现阶段, 这样由于整个系统开发过程都使用统一的概念和符号, 整个开发过程的对象模型是吻合一致和无缝连接的, 因此, 容易实现各个开发步骤多次反复迭代。每次迭代都会增加或明确一些目标系统的性质, 但不对先前工作结果做本质性改动, 从而减少了不一致性, 降低了出错的可能性。综上, 面向对象的软件设计方法不仅保证了决策者与系统分析人员之间的沟通和DSS 能最大限度地满足决策者的需求, 而且也缩短了软件开发周期, 提高了软件开

发效率。

将OMT与原型系统模型结合起来,就得到了基于OMT快速原型法的模型(图2)。

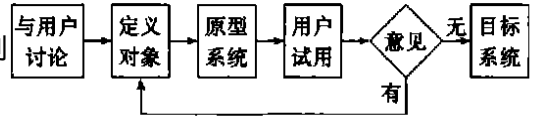


图2 基于OMT快速原型法的模型

1.4 基于OMT建立系统模型的步骤

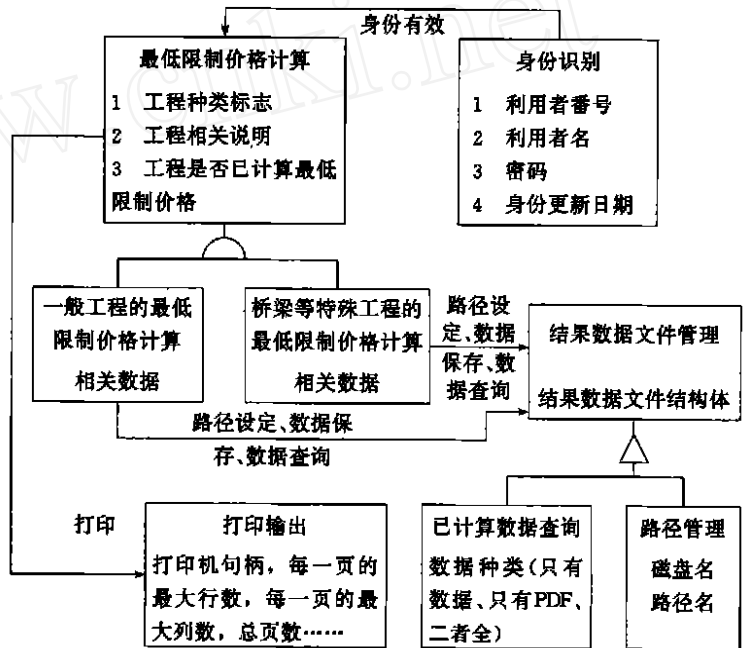
下面以笔者开发的决策支持系统——最低限制价格计算系统PRCSYS01为例来详细讨论基于OMT建立对象模型、动态模型的步骤。

1) 对象模型的建立。

类——对象层。标识对象的主要目的是使系统的技术表示更接近现实世界的概念视图,标识对象就是对问题域中某个实体的抽象,设立了某个对象就表示软件系统保存了该类的有关信息以及其他对象与该对象进行交互的能力。由于采用了用户驱动需求分析的递增模型方法,所以对象的定义方式也应在用户驱动需求分析时起到提示、启发和引导作用,以便让用户更深入、更细致地考虑他们的需求是什么;同时对象是属性和方法的封装体,具有数据和施于其上的操作以及与其他对象通讯的接口。因此对象定义为:对象= D、MS、DS、MI。其中: D 为对象标识;MS 为对象操作集合;DS 为对象数据结构;MI 为对象受理的消息名集合(对外接口)。

在对类——对象进行抽象时,首先采用基于语言的信息分析法(Linguistic-based on Information Analysis 简称LIA),从用户对需求的描述中,抽象出类的初步集合,然后根据分析,对初步确定的类进行审查和删除(图3)。

类——对象的泛化和特化。泛化是一个自底向上的过程,即由特殊到一般的过程。泛化是指对几个类的共同的操作和相同的数据结构进行提取,形成父类,继而将这几个类通过继承,作为该父类的子类,这样可以减少重复劳动,提高软件的可重用性,且



便于统一维护。为了充分体现软件的可重用性和可维护性,父类的抽象是关键,在这里考虑了充分利用类的多态性。特化是自顶向下的过程,即由一般到特殊的过程。特化是指利用类的继承性,在继承父类的基础上,添加子类特有的属性、操作等,从而实现软件开发由粗到细,由上到下的分层。这里考虑了利用类的数据隐蔽性,将局部的操作和数据私有化。经过泛化和特化后,建立的类的关联关系见图3。

主题层。为了降低软件设计的复杂程度,按问题领域将系统再划分为几个不同的主题,即将系统包含的内容分解为若干个范畴。在这里只考虑问题域,不考虑其实现技术。在PRC-

SYS01 中, 划分为 5 个主题(图 4)。

属性定义及对象模型的建立。属性是对象的性质。属性的确定与问题域有关, 也与目标系统任务有关。将分析结果综合在一起得到了DSS 的对象模型(图 3)。

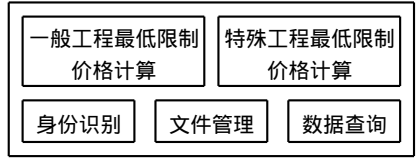


图4 主题层

2) 动态模型的建立。

动态模型表示瞬时、行为化的系统“控制”性质, 它规定了对象模型中对象的合法变化序列。建立动态模型的步骤: 系统运行脚本 事件轨迹图 事件流图 状态迁移图 动态模型图。

编写系统运行脚本。系统运行脚本分为系统正常和非正常运行脚本。首先, 编写正常运行脚本, 除了考虑系统正常运行时的一般情况以外, 还应考虑特殊情况, 如输入、输出数据的临界值(最大、最小); 其次, 编写非正常运行脚本, 主要考虑操作错误导致系统非正常运行时的系统运行状况。

事件轨迹图。从脚本出发, 确定每个事件的发送和接受对象后, 就可以用事件轨迹图的形式把事件序列以及事件与对象的关系, 形象、清晰地表达出来。事件轨迹图实质上是脚本的图解, 它克服了用语言描述的脚本在阅读和理解时可能产生的二义性。

基于消息的事件流图。对象之间通过消息建立了联系。对象根据消息的不同, 选择不同方法对消息进行处理和响应, 从而实现了操作过程的间接调用。每一个对象相当于一个信息处理中心, 接受并处理各种消息。事件驱动构成了面向对象程序设计方法的核心, 使得应用程序的通信标准化。

状态迁移图及动态模型。状态迁移图描绘事件与对象状态的关系。当对象接受一个事件后, 它的下一个状态取决于当前状态和所接受的事件。将各个类的状态图通过事件合并起来, 就构成了系统的动态模型。图 5 示出 PRCSYS01 系统状态迁移图 and 动态模型图的一部分。

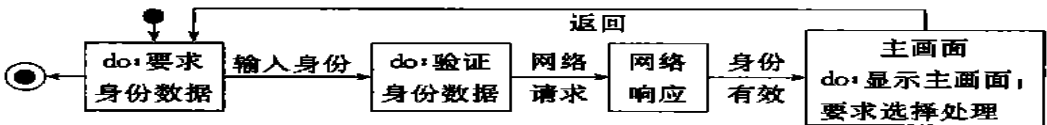


图 5 状态迁移图及部分动态模型

1.5 原型系统的设计及快速实现

为了快速实现原型系统, 采用将软件界面设计与实现系统内部功能分离的方式, 也就是说最初的原型系统只有界面切换, 而无用户需要的实际功能, 在 PRCSYS10 中, 利用 Borland C + + 5.01J 集成开发环境的资源编辑器, 根据事件轨迹图和状态图, 采用用户自定义消息驱动各个界面, 得到了用户能自由操作的原型系统。

1) 人机交互界面的详细设计。在设计 PRCSYS01 的人机界面时, 首先对用户所用的现有的其他系统作了认真考察, 为系统界面的设计提供依据; 其次, 对有些主要界面, 用户根据经验提出界面蓝图, 再根据软件开发及操作上的要求来设计系统界面。

2) 任务管理设计。虽然在 PRCSYS01 系统中存在多个任务, 但他们都是串行执行的, 因此从本质上讲, 该系统是一个单任务系统, 没有必要像多任务系统那样建立任务(Task)或任务协调类(Task Coordinator); 但为了灵活、快速地实现原型系统, 在系统的框架窗口类里, 利用

了自定义消息和对自定义消息的响应来协调各任务,从而实现界面的打开和关闭.主要界面间自定义消息的发送和接受见图6.

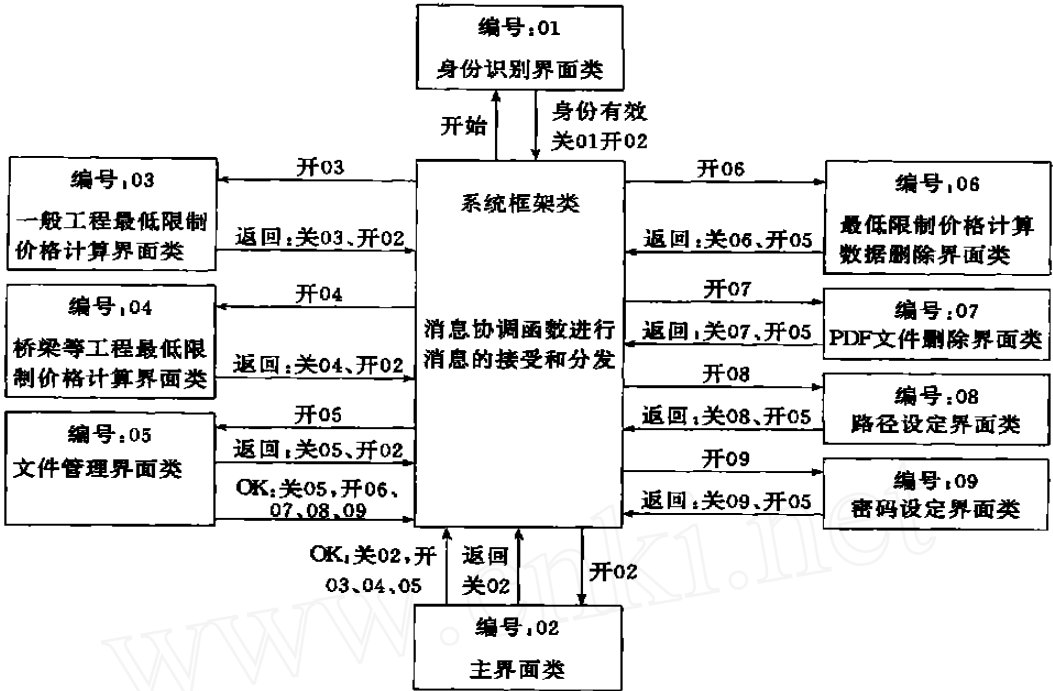


图6 主要界面间自定义消息的发送和接受

2 结束语

所开发的软件系统在1999年初完成,经过用户的严格测试后,已成功地用于实际工程的招标过程中.实际运用结果表明,系统运行稳定,界面友好,操作简便,具有良好的容错能力,在线帮助功能完善;由于充分利用了OMT的特性,将原型系统与目标系统统一,由原型系统直接渐增为目标系统,因此系统具有良好的开放性、可扩充性和可维护性.

参 考 文 献

- 1 高洪深 决策支持系统(DSS)理论·方法·案例 北京:清华大学出版社,南宁:广西科学技术出版社,1996 1
- 2 单承戈 决策支持系统问题模型的可视化构造方法 计算机应用:2000(9):25~27
- 3 陈增荣 软件开发方法述评 计算机世界—技术专题,http: www.computerworld.com.cn/week/tech/012794.html
- 4 雷战波,许倩钰 面向对象方法OOSA、OOA、OMT的比较研究 电脑技术信息,1999,8(8):32~34
- 5 张海藩 软件工程导论 第3版 北京:清华大学出版社,1998 182