

软件工程新途径——面向对象方法学

徐瑞鑫, 朱成禹

(中国科学院长春光学精密机械与物理研究所, 吉林 长春 130021)

摘要: 基于面向对象方法的软件工程是一个解决软件危机、提高软件生产率和质量的有效途径。本文简要分析了传统软件工程实现方法的缺陷, 阐述了面向对象方法学的基本概念和优点。在详细介绍了面向对象系统的三个基本要素的基础上, 对面向对象方法在软件工程中的实现方案做了论述。

关键词: 软件工程; 对象; 面向对象方法

中图分类号: TP311.5 文献标识码: A

1 引言

软件工程是指导计算机软件开发和维护的工程学科。它传统的实现途径——生命周期方法学, 在软件开发成功率提高和软件危机的缓解等方面都发挥过重要的作用。但随着计算机软件的规模和数量以惊人的速度急剧膨胀, 这种传统的方法逐渐显得陈旧和落后。于是, 一种崭新的软件工程实现途径, 面向对象方法学应运而生。

2 传统的生命周期方法学的缺点

生命周期方法学使用的基本技术是结构分析和结构设计技术。相对于早期的那种只重视编程、轻视对用户需求的了解和分析、最终产品只有程序代码, 没有相应的文档资料的个体化“软件”开发方法来说, 生命周期方法学是一个巨大的进步。但其本身也具有着明显的缺点:

a. 稳定性差

结构分析、设计技术是围绕实现操作的过程来构造系统的, 然而用户对系统的各种需求变化大部分是针对功能的。因此, 这种变化对基于过程的设计来说是灾难性的, 也就是说, 用户需求的变化往往造成系统结构的较大变化, 从而要花费很大的代价才能实现这种变化。所以说, 由这种技术设计出的系统常常是不稳定的。

b. 可扩充性差

结构分析和结构设计技术对其目标系统的边

界有着清楚的定义。由于用结构分析、设计技术开发出的系统结构依赖于对系统边界的定义, 因此很难把这样的系统扩展到新的边界。从而也就造成了系统难以修改和扩充的缺点。

c. 重用性差

将数据和操作作为分离的实体来看待和处理, 这也是结构分析和设计技术的一个主要特征。这使得在软件的实现阶段, 一些本来具有潜在可重用价值的软件部分已与具体应用环境密不可分, 从而使得用该方法开发出的软件的可重用性较差。

d. 开发效率低下

生命周期方法学强调需求分析的重要性, 强调在每个开发阶段结束之前必须进行严格的评审, 通过严格的质量管理, 提高了软件开发的成功率。但是, 随着计算机应用的不断普及, 人们对计算机软件的需求量急剧增长。这就使得这套“有条不紊”、“步步为营”的开发方法的生产效率越来越难以满足用户的需求。

3 面向对象方法学

近年来, 面向对象方法学日益受到人们的普遍重视。面向对象方法就是尽可能模拟人类习惯的思维方式, 使开发软件的方法和过程尽可能接近人类认识世界解决问题的方法和过程。它的出现, 克服了传统设计方法的缺点, 提高了软件系统的稳定性、可修改性和重用性。

面向对象的系统包含了三个要素:对象、类和封装,这三个要素反映了面向对象的基本观念。

对象是一个状态和操作(或方法)的封装体。状态是由对象的数据结构的内容和值定义的,方法是一系列的实现步骤,它是由若干操作构成的。面向对象的方法认为现实世界是由各种对象组成的,任何事物都是对象,复杂的对象可以由比较简单的对象以某种方式组合而成。对象是事件的主体,它们之间只能通过传递消息来相互联系。每当一个操作被调用,就有一条消息被发送到这个对象上,消息带来了将要被执行的操作的详细内容。

类是创建对象的样板,它包含着所创建对象的状态描述和方法的定义。类的完整描述包含了外部接口和内部算法以及数据结构的形式。由一个特定类所创建的对象叫做这个类的实例,因此类是对象的抽象及描述,它是具有共同行为的若干对象的统一描述体。抽象是一种从一般观点看待事物的方法,它要求程序员集中于事物的本质特征,而不是具体的细节或具体的实现。面向对象方法把所有对象划分成各种对象类,每个对象类都定义了一组数据和一组方法,数据是方法的属性,用来表示对象的状态。当建立一个对象类的新实例时,可以根据要求为新对象定义其专用的数据,用来描述该对象独特的属性值。例如,在显示器上画不同风格的直线,需要赋给画笔对象不同的属性,以改变直线的样式和颜色等。面向对象系统提供了创建新类的一种方法——继承,继承的本质特征就是行为共享。新类共享已有类的行为,而自己还具有修改和添加的行为。从一个类中继承下来的新类,将继承已有类的所有方法和属性,还可以添加属于自己的方法和属性。新类被称为已有类的子类(派生类),已有类被称为父类(基类)。例如,对于 CControlBar 类,其下面派生出几个新类 CDialogBar、COleResizeBar、CStatusBar、CToolsBar,这些子类均具有父类 CControlBar 的某些特征。它们都是控件栏,但它们都扩展了自己的新特征使彼此相互区分。它们分别用来模拟对话框、分隔窗、状态栏、工具栏。

封装是指将对象的方法程序和属性代码包装在一起。封装要求对象通过操作接口与外部相互联系,外界不能直接访问对象的私有数据,只有局部于对象的代码才可以访问该对象的私有数据,这种对象内部私有数据结构的不可访问性叫做数据隐藏。封装使得一个对象可以像部件一样用在各种程序中,而不用担心对象的功能受到影响。可

以说,封装就像是在对象的数据结构周围砌上一道“围墙”,该对象的方法是“围墙”上的门,只有通过这道门才可以进入“围墙”内部。

只使用对象和消息的方法被认为是基于对象的方法,如果进一步把所有对象都划分为类,则这种方法可称为是基于类的方法,但是它们都不是面向对象的方法。只有同时使用对象、类、继承和消息的方法才是真正的面向对象的方法。面向对象方法具有以下几个优点:

a. 稳定性好

面向对象方法基于构造问题领域的对象模型,是以对象为中心来构造软件系统。对象是一个独立的模块,是类运行时的实例。而软件系统中的类是软件开发者经过多次严格测试后的产品,具有很高的稳定性,而对象作为这些类运行的实例亦具有很好的稳定性。

b. 可重用性好

面向对象方法把数据和处理数据的方法封装在对象里,对象的封装性和数据隐藏性,使得它具有较强的独立性。如果把软件设计比作产品设计,则对象就相当于该产品的零件,这些零件独立于产品,但人们可以用它们生产出不同的产品。同样,程序员也可以将多个独立的对象组合在一起设计出多种不同的软件产品。由于程序设计中使用了以前开发过的具有一定独立功能的模块对象,使得软件开发的工作量大大降低,提高了软件的可靠性。

c. 可维护性好

面向对象方法设计的软件系统主要由对象组成,因此,对面向对象的软件进行维护也主要通过修改对象类来实现。例如:把一个汽车看作一个软件系统,那么这个软件系统将由发动机、轮胎、车体等对象构成,若要对汽车进行修理,只需修理其出故障的对象即可,由于这些对象独立于其它对象,因此在维护上容易实现。

4 面向对象方法的分析、设计与实现

将面向对象方法运用到软件工程中,可以将整个软件系统的开发过程划分为以下几个主要阶段:

4.1 分析阶段

分析过程就是提取软件系统的需求状况并以总结归纳的过程。在此过程中,分析人员必须同用户一道工作,从用户所提供的对问题的陈述(尽

可能详细周到) 中充分了解现实世界的基本特征, 然后根据这些特征把实际问题加以抽象, 逐步建

立起从三个互不相同而又密切相关的角度来描述现实世界的模型。下图表示了这一过程:

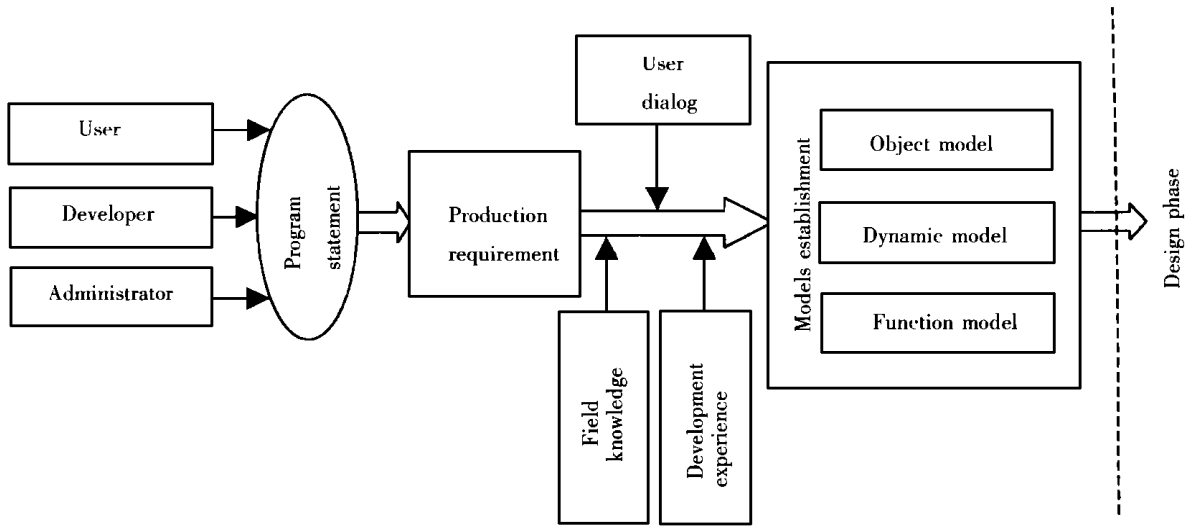


Fig. 1 Analysis process

分析模型中所构造出的“对象”是指应用领域的事物对象, 并非以后计算机程序中的对象。

作为用户和软件开发人员的“接口”, 分析阶段必须满足两点要求: a. 应该能被用户方面领域内的专家(而不只是程序员)所充分了解或能指出不对之处。分析模型决不可以包含任何实现细节或任何程序代码。b. 分析模型作为后续设计和实现阶段的工作框架, 必须准确明了地对对象、动态和功能模型进行描述和构造, 要为设计人员所充分理解。

4.2 系统设计阶段

分析阶段结束之后, 就要对整个软件系统得以实现的方法进行研究和设计。在这里主要完成两个较大的任务: a. 把整个系统划分为若干子系统, 设计出合理的系统结构, 使得后续的工作可以由几组开发人员分别在各自的子系统中独立工作。b. 作出一些明确的决策用以解决各自系统的软硬件资源分配的问题, 从而为下一步的设计作出必要的准备工作。

在这里每个子系统应拥有某些公共特征, 它定义了寻找问题的一个方面解的本质方式。一个子系统不是一个对象, 也不是一种功能, 而是类、关联、操作、事件和约束所组成的一个包。一个子系统通常由提供的服务所确定, 如 I/O 处理、文件系统的组织、界面规划等。

系统设计人员还必须对各子系统的软硬件资源要求进行分配实验并做进一步优化。例如, 系统

设计者决定: 在工作站上, 当窗口移动或擦除时, 屏幕上的变化应该是快速而且平滑的。则根据系统设计者的决定, 就要选择合适的通信协议和内存缓冲区的构造策略。

4.3 对象设计阶段

对象设计阶段是面向对象方法的软件工程中最为特色且又最为繁重的设计阶段。它是在前述的分析和系统设计基础上的进一步加工, 加进了具体实现方法的细节。

在此阶段中, 程序员需要利用已有类和定义自己的新类, 来进行对象的封装, 并通过消息的传递将这些对象组合在一起, 实现软件系统所提出的要求。在这一阶段, 在类之间创建一致接口的能力, 即如何实现对象之间消息的传递, 是程序设计的关键。

现在许多面向对象的语言, 为解决这一问题, 提供了很出色的技术。以 Visual C++ 为例: VC++ 提供了一个基础类库 MFC, 在 MFC 库中所有类形成了建立应用程序的框架, 它的内部封装了 Windows 应用程序的接口(API) 函数、数据结构和宏, 并以对象形式提供给程序员, 它充分利用了面向对象技术的数据抽象特性, 使 C 程序员从繁重而枯燥的 SDK 编程方法中解脱出来, 大大提高了编程效率。

4.4 实现阶段

在对象设计期间开发的对象类和它们之间的联系, 最后要进入到具体程序设计语言、数据库和

硬件的实现阶段。所有关键性决策已在设计阶段作出,这时程序设计只是开发周期的一部分。语言对设计决策有一定程度的影响,但设计的最后决策并不依赖于程序设计语言。

一个好的软件工程的实施,设计阶段是非常重要的,它不仅能够非常简洁地实现系统设计时所确定的目标,而且能够保留系统的灵活性和可扩充性。

现在比较流行的 C++、Java、Delphi、JavaScript 等程序设计语言都是面向对象或基于

对象的,并以它们强大的功能得到了广泛的应用,使得面向对象方法得以最终完美实现。

5 结 束 语

正如由上向下和模块化程序设计方法给软件工程带来巨大的发展一样,面向对象方法作为软件工程的一种新途径,同样使软件工程产生了深刻的变革。现在,以面向对象为基础的可视化编程软件已成为使用最广泛的编程工具。

参考文献:

- [1] Shooman Martin L. Software Engineering: Design, Reliability, and Management [M]. McGraw-Hill Book Company, 1983.
- [2] Code Peter, Yourdon Edward. Object-Oriented Analysis (Second Edition) [M]. Yourdon Press, 1991.
- [3] 汪成为,等 面向对象分析、设计及应用[M]. 北京:国防工业出版社, 1992.
- [4] 张海藩. 软件工程导论(第三版)[M]. 北京:清华大学出版社, 1998.
- [5] 宛延恺. C++ 语言和面向对象程序设计[M]. 北京:清华大学出版社, 1998.

New way of software engineering—object-oriented method

XU Rui-xin, ZHU Cheng-yu

(Changchun Institute of Optics, Fine Mechanics and Physics,
Chinese Academy of Sciences, Changchun 130021, China)

Abstract: Software Engineering based on object-oriented method is a new way to solve software crisis and to improve software productivity and quality. In this paper, the defects of the traditional Software Engineering realizing method is simply analyzed, and the fundamental concepts and the advantages of object-oriented are discussed. Based on three factors of object-oriented system, the application of object-oriented method in the Software Engineering is illustrated.

Key words: software engineering; object; object-oriented method

作者简介:徐瑞鑫(1972-),男,吉林省长春市人,现为中国科学院长春光学精密机械与物理研究所在读硕士研究生;朱成禹(1975-),男,黑龙江省铁力人,现为中国科学院长春光学精密机械与物理研究所在读硕士研究生。