

软件复用技术初探

王波波, 崔荣亮

(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:面向对象方法是软件复用技术的基本思想。文章在软件复用概念的基础上探讨了面向对象方法, 然后介绍软构件的概念并分析了它的一些特征: 它具有封装性和继承性的优点; 它由构件与技术说明书组成, 这个说明书用来详细描述构件的功能; 它比类具有更广阔的应用前景。最后给出了基于复用技术的软件开发方法和开发原则。

关键词:软件复用; 面向对象方法; 软构件

中图分类号:TP311.5 **文献标识码:**A

1 引言

软件复用是解决“软件危机”的重要手段之一, 也是软件技术发展的目的之一。无论是早期的结构化软件设计方法还是后来的面向对象的软件设计方法都朝着这个方向努力。自1968年D. McIlroy提出软件复用的思想以来, 软件复用技术已取得了一些突破。例如: 面向对象设计思想的引入, 基于体系结构的设计方法, 特定域的复用, 这都显示了软件复用的优越性。本文从软件复用的概念入手, 结合面向对象方法的思想介绍了软构件, 然后介绍了基于复用技术的软件开发方法。

2 软件复用及其技术

2.1 软件复用的概念

软件复用是从已定义的软件构件和为复用而设计的部件中构造新系统的过程。在软件开发过程中, 复用是指能充分利用已有的软件开发技术和软件开发周期中各阶段的产品^[3]。软件复用分三个层次:

- 领域知识的复用;
- 方法和标准的复用: 如领域模型, 软件需求说明书, 测试方法等;
- 软件成分的复用: 如功能模块, 测试用例及面向对象程序设计中的类等;

另外, 就可复用的软件产品而言, 应具有以下三个特征:

- 模块化的结构: 模块化的结构能保证内部的修改不影响系统总体。
- 软件的运行不依赖于具体的环境, 即把依赖于具体环境的部分模块化, 如果环境变化时则用相应的模块进行替换。
- 要用标准的统一的数据模式: 这样的数据模式能减少模块之间的数据交换和模块之间的依赖性。

2.2 面向对象的设计思想

传统的面向数据/过程的软件设计方法把数据和过程看作相互独立的实体。数据用来表达问题中的信息, 而程序则用于对数据的处理。这样的设计方法可复用的成分很少。实际上, 计算机所要解决的问题都是相互联系的事物, 这种事物我们称之为对象, 它可用两个特征来描述: 数据结构和对数据的操作。也就是说, 数据结构和对数据的操作结合在一起才反映了我们要解决的问题。

面向对象的设计方法与传统的面向数据/过程的方法不同, 它的基本思想是: 对问题进行领域分析, 按照人们认识世界的方式建立问题领域的模型; 进而设计软件。这样的软件由完整反映问题本质的对象组成。在这里引入几个重要的术语:

- 对象, 是数据结构和对数据的操作的总称。
- 类, 是对一组具有相同数据结构和相同操作的对象描述。

- 消息, 它要求某个对象执行类中定义的某个操作的规格说明。它由三部分组成: 接受消息, 消息选择子, 零个或多个元。

- 方法, 即对象所能执行的操作。

另外, 面向对象的程序设计方法有以下优点:

- 模块化, 信息隐蔽和数据抽象; 对象具有封装性, 它是数据和过程的结合, 是模块化的理想机制。就对象而言, 从外部只能看到对象的外特性, 即它对外是不可见的, 这称为信息隐蔽。这样, 依据类的外部活动就可以使用它, 从而减少了程序之间的相互依赖。数据抽象是信息隐蔽的一种表现形式, 一个类就是一种抽象的数据类型。

- 继承性, 一个类的上层可以有父类(超类), 下层可以有子类。一个类可以直接继承其父类的全部描述。继承性使相似的对象可以共享程序代码和数据。可见, 继承性使我们把已有的一般性解具体化, 以达到复用的目的。

从以上分析中我们可以看出, 面向对象方法给软件复用提供了一个良好的思想方法。通过类的继承和封装机制来实现代码的复用, 通过定义类把数据和函数集成在一起, 外部程序只能通过类规定的接口和类进行通讯, 这如同硬件电路中的集成电路芯片一样, 可见封装增强了类之间的独立性, 这是软件复用的基础。继承是用来表示类之间的相似性机制。在以往的情况中, 即使是可复用的部件也很少能确切地满足新需求, 但是, 有了继承机制, 已有的部件可以通过扩展和具体化来适应新的要求, 从而方便了软件的复用。

2.3 软构件

在软件复用技术的 30 年发展历程中, 并没有出现实质性的突破, 这是因为可复用的成分没有足够的抽象性, 表现在其规模, 适用范围和易用性等方面都有很强的约束要求, 仅在某些专用领域(如数学函数库)应用很成功, 但在通用领域都无甚作为; 同时软件应用的范围日益广阔也造成了软件形式的多样性。这样, 很难设计出适合于通用领域的构件。并且, 随着软件复用研究的深入, 复用成分已从早期的代码扩展到一切可用来重构软件的成分, 包括软件开发方法, 需求规格说明书, 源代码, 开发工具与支撑环境, 测试与维护信息等。面对如此复杂的复用, 人们提出了软构件的概念。

软构件指软件开发中的可重用部件。它由构件及其详细的说明书构成, 这个说明书用来阐明构件的功能。它可以是面向对象方法中的对象

(类), 应用框架, 也可以是非面向对象方法提供的复用部件: 如数据类型, 操作, 函数, 过程, 甚至文档和规格说明。因此, 构件被看作软件复用的基本单位^[4], 并且我们可明显看出软构件是类的泛化, 它拥有比类更加广阔的应用背景。图 1 示出了软构件和类的区别。

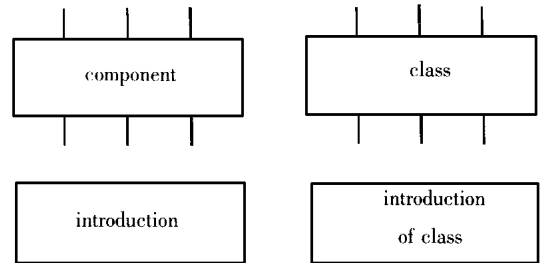


Fig. 1 Difference between class and component

3 基于软件复用技术的软件开发方法

现代大型软件一般要综合多个学科, 多个空间的信息。所以要首先进行领域分析, 建立领域模型, 然后在软构件的基础上进行软件设计。

3.1 领域分析

领域是指一个问题或任务的范围。在其中要开发多个高度相似的应用系统来满足客户的需求。简言之, 一个领域是一组相关的系统, 相关的方向是公共设计, 公共服务, 公共技术和公共信息体系结构^[3]。领域分析是标识和捕捉特定域中相似系统的有关信息, 尤其是现在和将来需要使用的系统信息。通过分析挖掘出其内在的规律及特征, 并对信息整理与有效的组织形成正确的模型的活动。也就是说领域分析用来识别应用软件系统所属领域的共同特征, 建立领域模型。有报道^[4]GIS 系统中将应用模型划分为空间模型和非空间模型两大类, 然后再对二者进行细分。在领域知识, 领域模型, 领域框架, 领域分类和领域语言的基础上生成可复用的部件和着眼于构建领域内系统的领域标准。

3.2 面向对象的软构件及其实现方法

在复用软件设计中, 如何设计和构造可复用的软构件是软件设计者首先要面对的问题。一般地, 在运用面向对象方法进行软件开发时应着眼于以下几个方面^[2]:

1) 封装性

在软构件的定义中, 我们强调封装性(信息的

隐蔽性), 意味着用户只关心事件的输入或输出, 而对事件内部不必关心。也就是说, 方法和事件的实现是独立于应用的。用户可以在软构件中定义自己的事件, 至于内部的复杂性调用者并不知晓, 从而提高了隐蔽性。由此可见封装是抽象的一种具体应用, 它同时体现了过程抽象和数据抽象。

2) 重载

重载就是在同一软构件中用同一名字来表示不同的方法名。一般有两种实现方法: 一种是方法参数的个数重载; 另一种是方法参数的类型重载。这些过程类似于 C++ 中的重载过程。

3) 继承

继承就是高层的类(父类)在不同范围的复用。低层的类(子类)是高层的具体化, 它包含有更多的信息。因此, 在对一个具体应用系统进行面向对象分析时, 可以考虑某些类在本领域的多个应用系统中的共性, 从而抽象出对整个领域可复用的一般类即领域构件。例如, 在已知的软构件中寻找构件之间的属性、结构及其它特征之间的共性, 并将它们之间的共性抽取出来形成父辈构件, 如此循环往复得到一个基本的构件。图 2 给出了用继承性来实现复用的一个例子。

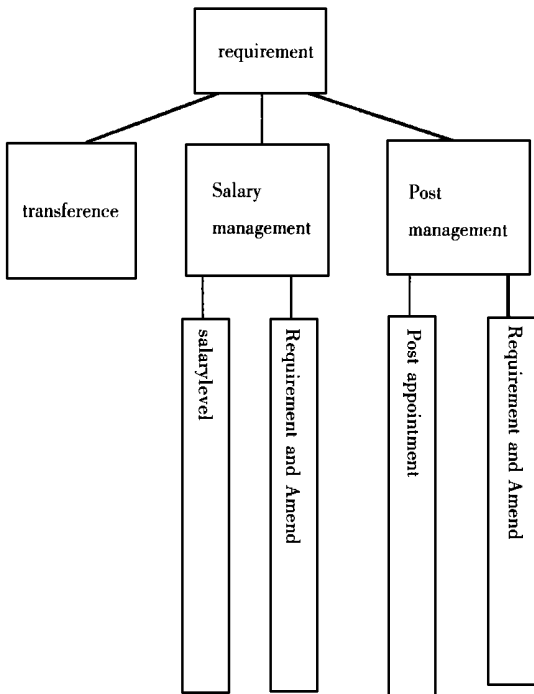


Fig. 2 Realization of reuse through inheritance

4) 聚合

聚合就是把一个难以复用的大软件分成若干部分构件, 使其中的某些构件成为可复用的构件。可以这样进行: 对一些不需要动态变化的属性与

服务要稳定地保持, 不必受部分特征的改变而改变; 而把需要进行动态变化的属性与服务进行独立地处理并且实现与整体的灵活对接, 同时, 在其实现上不应有特殊的机制。例如图 3 中学生的家庭住址是不变的属性, 其成绩是变化的, 而其身体状况是相对稳定的。

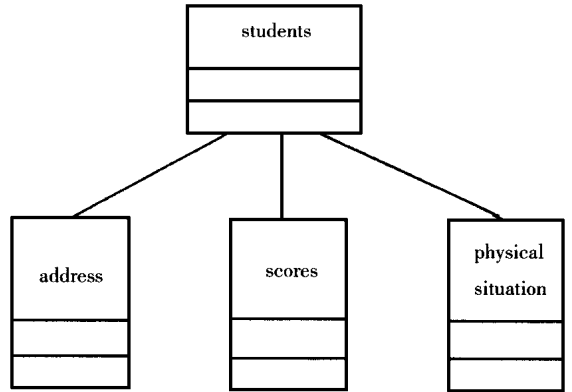


Fig. 3 Realization of reuse through aggregation

5) 多态性

多态性是针对方法而言的, 指某方法不仅可以处理有相同结构的对象也可处理具有不同结构的对象。主要包括通过方法的重载而体现出来的重载多态性和通过继承体现出来的继承多态性。

总的说来, 面向对象方法可以促进可复用性, 易于维护, 易于修改, 可合理利用共同性, 减少了复杂性。

3.3 基于软件复用的开发原则

在软件工程中, 无论是软件开发还是软件维护均应将软件复用思想作为重要的指导原则: 从领域分析, 需求分析, 概要设计, 详细设计, 到编码和单元测试, 集成测试和维护阶段, 注意各说明书的结构化, 强调可复用文档的准备, 为软件体系结构的复用准备相应的模型系统, 最后在维护阶段注意复用工程的问题。

4 结 束 语

本文主要讨论了软件复用的概念及相关技术, 并介绍了面向对象方法与软构件之间的关系, 最后阐述了基于复用思想的软件开发原则。尽管在面向对象方法基础上实现软件复用还有许多理论和实践的问题有待研究, 但是面向对象技术集数据抽象, 抽象数据类型及继承于一体, 使软件工程中所要求的模块化, 信息隐蔽, 抽象性及可复用

得以体现,是软件复用的基本思想方法。它的应用 会越来越广泛。

参考文献:

- [1] 张海藩. 软件工程化导论[M]. 北京:清华大学出版社,1992.
- [2] 游晓明,刘升. 软件复用技术实现策略研究[J]. 武汉大学学报(自然科学版),1999,45(5):706-710.
- [3] 李庆如,麦中凡. 域分析:为软件复用产生有用的模型[J]. 计算机研究与发展,1999,36(10):1188-1196.
- [4] 关信红,等. 软件复用技术与GIS软件开发研究[J]. 武汉大学学报(自然科学版),1999,45(5)719-722.
- [5] 贺岚,狄玉来. 基于构件的软件设计模型[J]. 计算机研究与发展,1998,35(5):452-454.
- [6] 朱海滨. 面向对象技术——原理与设计[M]. 长沙:国防科技大学出版社,1992.

Moderate research on software reuse technology

WANG Bo-bo, CUI Rong-liang

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

Abstract: Object-oriented Paradigm is the basis of software reuse. The article probed into the OOP (object-oriented paradigm) on the basis of definition of software reuse. Then, the definition of software component is derived and analyzed: It has many special advantages such as encapsulation and inheritance; It is composed of component and technology introduction which described its function. It has broader applications than class in future. At last, the software development method and principle are derived on the basis of software reuse technology.

Key words: object-oriented paradigm; software reuse; soft component

作者简介:王波波(1974-)男,河南灵宝人,1997年毕业于长春光学精密机械学院电子工程系,1999年进入长春光学精密机械与物理研究所攻读机械电子工程专业,研究方向为计算机图像处理。

崔荣亮(1976-)男,山东莒县人,1999年毕业于长春光学精密机械学院光学物理系,同年进入长春光学精密机械与物理研究所攻读机械电子工程专业,研究方向为电子及计算机应用。