

文章编号:1671-6833(2003)03-0024-05

# 软件复用与基于面向对象框架的软件开发方法

王丽娟<sup>1</sup>, 孙西超<sup>2</sup>, 底松茂<sup>3</sup>, 王哲光<sup>1</sup>, 冯志慧<sup>1</sup>, 刘倩<sup>1</sup>

(1. 河南农业大学现代教育部, 河南 郑州 450002; 2. 河南农业大学财务处, 河南 郑州 450002; 3. 郑州大学财务处, 河南 郑州 450002)

**摘要:** 介绍了软件复用的基本概念和意义, 分析了框架的特征以及它与类库、组件、设计模式等常用的面向对象复用技术(BFD) 的联系, 给出了基于框架的软件开发方法的开发过程. 首先进行领域分析即在给定的领域中, 通过分析找出对所有应用程序来说明共同的类和对象, 建立领域模型; 其次, 结合领域分析, 给出系统需求说明书; 第三步是框架设计, 通过对应用程序进行分析和设计, 得到应用程序在该领域的特殊要求, 从而扩展框架来实现该应用程序; 最后是软件的测试. 目前BFD 的研究更多的集中于特定的领域如 制造业、银行业、卫生系统等. 在开发方法上, 将较多的采用黑盒框架, 在框架集成方面, 更注重将框架与类库、组件和设计模式等联合在一起使用, 更注重对框架继承性的研究.

**关键词:** 软件复用; 面向对象; 框架; 组件; 设计模式

**中图分类号:** TP 301 **文献标识码:** A

## 0 引言

目前计算机技术的飞速发展使得软件开发的环境日益复杂, 应用软件需要具有开放的系统结构和跨越多个异构平台的能力, 这样就导致在应用系统中的软件越来越庞大, 传统的开发技术已远远不能满足需要, 在这种情况下, 有关软件复用的研究也就应运而生了.

软件复用是软件工程中具有很高前瞻性的研究领域, 面向对象的软件复用技术是软件复用的一种重要方式, 基于面向对象框架的软件开发技术BFD(Based -framework Development) 是一种面向特定领域的复用技术, 随着面向对象技术的成熟, 面向对象框架的软件开发被认为是一种最有前途的提高软件重用度和软件生网络产力的面向对象技术. 该方法对特定领域复杂系统的高可靠专用软件具有很高的实用价值, 为特定领域的软件开发提供了一种强有力的技术支持<sup>[1]</sup>.

## 1 软件复用的概念和意义

软件复用是指重复使用“为了复用目的而设计的软件”的过程, 而可复用软件则是指为了复用目的而设计的软件, 是指在开发新的应用系统时重复使用以前的资源, 如设计、代码、文档等. 软件

复用更强调的是复用的目的, 它是一种系统化的方法, 为了复用而进行设计, 为了复用而开发, 并且要有效地组织和管理这些复用产品, 方便人们查找和使用, 从而提高系统开发效率及软件质量. 软件复用一定要有软件开发的积累, 有了一定的积累后才能进行软件复用. 总的来说, 软件复用有三个基本问题: 一是必须有可复用的对象, 二是可复用的对象必须是有用的, 三是使用者知道如何去使用被复用的对象. 软件复用还包括两个相关的过程, 即可复用软件的开发和可复用软件的应用系统的构造. 解决好这几方面的问题, 才能实现真正意义上的软件复用<sup>[2]</sup>.

软件复用通常可分为两类: 产品复用和过程复用. 产品复用指对软件开发过程中生成的各种产品(需求规约、设计、程序、测试计划 and 数据等) 构件的复用, 涉及可复用构件的建造及可复用构件的使用两个方面, 通过专业性的构件开发和基于构件复用的系统集成实现软件的工业化生产. 过程复用指通过采用自动化技术, 通过生成或变换, 自动生成最终所需的系统, 完全通用的过程复用意味着软件生产的自动化, 当前过程复用的实践大多和领域相关, 如特定领域的应用生成器等. 过程复用是非常理想的软件复用方式, 成为主要的研究课题<sup>[3]</sup>.

收稿日期: 2003-04-02; 修订日期: 2003-06-25

作者简介: 王丽娟(1966-), 女, 河南省周口市人, 河南农业大学副教授, 主要从事计算机理论教学及应用研究.

软件复用的意义,具体说主要体现在五个方面:①软件复用可以提高软件生产率,从而减少开发代价.采用可复用构件构建系统还可以提高系统的性能和可靠性,因为可复用构件经过了高度的优化,并且在实践中是经受过考验的.②软件复用能够减少系统的维护代价.由于使用检验的构件,减少了可能的错误,同时软件中需要维护的部分也减少了.③软件复用能够提高系统间的互操作性.④软件复用可以支持快速原型设计.利用可复用构件和构架可以快速有效地构造出应用程序的原型,以获得用户对系统功能的反馈.⑤软件复用还能够减少培训开销,如同硬件工程师使用相同的集成电路块设计不同类型的系统一样,软件工程师也将使用一个可复用构件库来构造系统,而其中的构件都是他们熟悉和精通的<sup>[1~4]</sup>.

## 2 基于面向对象框架的软件开发方法和开发过程

基于面向对象的开发方法是一种强有力的开发方法,它在模块化、可复用性、可扩展性、反向控制等方面有其特有的优越性.采用基于框架的方法开发软件,可以以框架为复用部件的基本单元来实现软件工业化生产,有效地降低软件开发的成本,提高生产效率和软件可靠性,尤其对特定领域复杂系统的高可靠专用软件更具有实用价值,为这些特定领域的软件开发提供了一种强有力的技术.基于框架的软件开发过程如图1所示,下面详细介绍它的开发过程<sup>[1~3]</sup>.

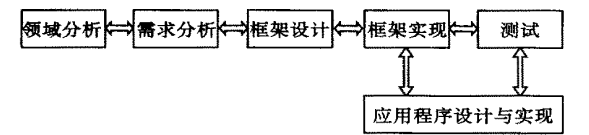


图1 基于框架的软件开发过程

Fig. 1 Software development process based on framework

### 2.1 领域分析

领域分析是在给定的领域中,通过分析找出对所有应用程序来说明共同的类和对象,建立领域模型,领域模型中包含了该领域中的需求关系、领域中的相关概念及它们之间的关系.该模型应该集中于领域实体而不是实现细节<sup>[3]</sup>.

### 2.2 需求分析

需求分析的目的是为了捕获所有有效的需求,给出一个能满足这些需求的理想系统,并结合领域分析,找出系统的共同点,给出系统需求说明书.

### 2.3 框架设计

框架设计包括框架体系结构设计和详细设计.在体系结构设计阶段,结合设计模型,定义许多抽象类,使得整个系统结构的设计支持更灵活的变化.在详细设计阶段,将描述在体系结构设计中抽象出来的类,并精确地定义它.①框架实现:框架的实现即是对框架的抽象类和具体类进行编码,一般采用自顶至下的开发方法.在整个实现过程中,还必须定义一个完整的实现标准,如编码的统一性.这样做可以时的开发出来的框架更容易被应用程序开发人员所理解和应用.②应用程序设计和实现.通过对应用程序进行分析和设计,得到应用程序在该领域的特殊要求,从而扩展框架来实现该应用程序.这一阶段,需要应用程序开发人员对框架进行系统的学习,从而能够有效地使用框架来进行开发.

### 2.4 测试

该过程一般分为单元测试和集成测试.一个单元是指某个操作、类或包含多个类的模块,甚至整个框架.单元测试又分为结构测试和功能测试.结构测试需要熟悉该单元的内部结构,对单元的代码和分支结构进行测试,主要是验证单元的可靠性;而功能测试则不关心该单元的内部结构,主要测试单元对特定输入的响应,检查是否符合该单元的功能需求.集成测试则测试这些软件单元集成在一起时,是否能满足整个程序的功能要求.同其它的软件开发过程一样,以上所述的阶段都是相互重叠,不断反复的过程.

## 3 框架和其它面向对象复用技术及其联系

在基于框架的软件开发中,有一些其它的面面向对象复用技术和框架密切相关,它们是类库、组件和设计模式<sup>[4~9]</sup>.

### 3.1 类库

在基于框架的开发过程中,类库扮演着重要的角色.类库是一组相关的、可复用的类的集合,这些类提供了通用的功能.类库强调的是代码重用,他们是面向对象环境下的“子程序库”.框架将类库集成到其内部来简化开发工作,并通过类库来完成一些基本的任务如字符串处理、文件管理等.所有的框架都是类库,但并不是所有的类库都是框架.框架通过下列方式来扩展类库的功能.框架是包含特定领域对象体系结构和功能的“半成品”.框架中的组件一起合作,为一类相关的应用程序提供一个标准的体系结构骨架,通过框架组

件集成和组装可以得到完整的应用程序. 与之相比, 类库不是针对特定领域, 而只是提供较小范围内的复用.

与框架运行时的“反向控制”相比, 类库则是被动的. 当使用类库时, 需要写应用程序的主体并强调你想要复用的代码; 而当使用框架时, 应该复用应用程序的主体, 写主体调用的代码, 如图 2 所示.

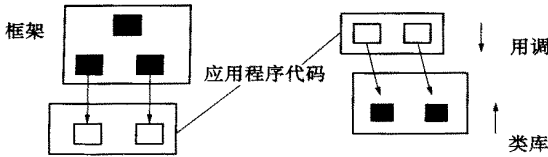


图 2 框架的反向控制  
Fig. 2 Opposite control of framework

3.2 组件

组件是自包含的抽象数据类型 (ADTS), 并能够被组装在一起形成完整的程序. 组件典型的例子如 Microsoft 公司的 ActiveX 和 Borland 公司的 VCL 组件通过黑箱技术, 被设计成为定义了一个相互协调的操作的集合, 该操作集可以通过在语法上一致的接口来访问.

组件与框架是不同的, 但是它们可以相互协作. 首先, 组件为框架提供了一个可复用的环境. 每一个组件在开发时都假定了它的存在环境, 如果组件假定的环境不同, 则它们就不可能在一起很好地工作; 而框架则为组件提供了一个标准的环境来处理错误, 交换数据, 激活彼此间的互操作. 通常所谓的“组件系统”如 OLE, OpenDoc 和 Beans 实际上是框架, 框架与组件在一起协同工作的另一种方式就是框架使得开发新的组件变得更为容易. 实际上, 无论所设计的组件库是如何优秀, 应用程序的功能看起来还是无法完全满足, 从而必须开发新的组件, 框架则提供了设计新组建的规范和实现它们的模版.

与框架相比, 组件是松耦合的, 并支持二进制级别的复用. 在基于框架的开发过程中, 组件可以被黑盒框架用作为可嵌入的策略模式. 通常, 框架被用来简化体系结构和中间件的开发, 而组件则被用来简化最终用户程序的开发.

3.3 设计模式

设计模式代表了在软件开发过程中特定场景下解决重复发生的问题的方案. 每一个设计模式都集中于一个特定的面向对象设计问题或设计要点, 描述了什么时候使用它, 以及使用的效果和如何取舍等. 一个使用设计模式的框架比不用设计

模型的框架更可能获得高层次的设计复用和代码复用. 它们的主要区别为: 框架是一个可执行的软件, 而设计模型代表了软件设计过程中的设计经验. 从这个意义来说, 框架是个物理实体, 而设计模型是个逻辑实体. 框架可以看成是一个或多个设计模式解决方案的物理实现, 而模式则指导如何实现这些方案. 设计模型是比框架更小的体系结构元素, 典型的框架都包含多个设计模式, 反之则不成立. 框架总是针对一个特定的应用领域, 而设计模式可以被用于任意应用. 框架的威力在于它们能够用程序设计语言编写, 不仅可以被学习, 还能被直接执行和复用.

在实际的开发过程中, 以上所讨论的复用技术经常是联系在一起来完成应用程序的开发工作的<sup>[1~9]</sup>, 如图 3 所示.

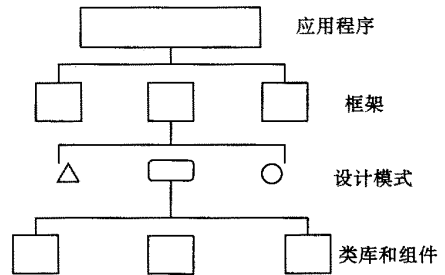


图 3 基于框架的应用程序分解  
Fig. 3 Disassembly of application program based on framework

4 基于面向对象框架的软件开发应用示例

防空 CCI 通信网仿真系统就是基于面向对象框架的软件开发方法的应用实例, CCI 由指挥、控制、通信和情报四个方面组成, 是一个庞大的系统, 防空 CI 通信网包括军指挥中心 (见图 4) 和若干 (如 3 个) 师属防空指挥系统 (见图 5)<sup>[1]</sup>. 我们仅以单机环境下的军师级防空作战指挥系统的面向对象仿真框架为例. 由于师级防空作战指挥系统 (图 4) 和军级防空中心 (图 5) 在结构上有一定的相似, 它们有相同类型的方舱, 有相似的内部结构. 当军级防空作战指挥系统完成后, 对应用领域内的软件系统作深入了解分析后得到基本类, 在软件框架的指导下寻找可复用构件同时开发一些新构件并进行系统组装. 这样, 我们以军级防空作战指挥系统为基础, 抽象出其框架, 复用到师级防空作战指挥系统, 如图 6 所示. 类似地, 再向下拓展至团、连、营等, 这样就减少了开发整个 CCI 通信网的费用, 也为系统的扩展提供了条件<sup>[1]</sup>.

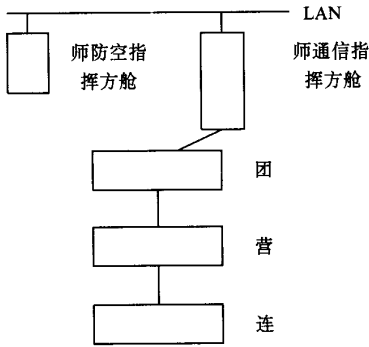


图4 军防空指挥中心示意图

Fig. 4 Aerial defence commanding center of the corps

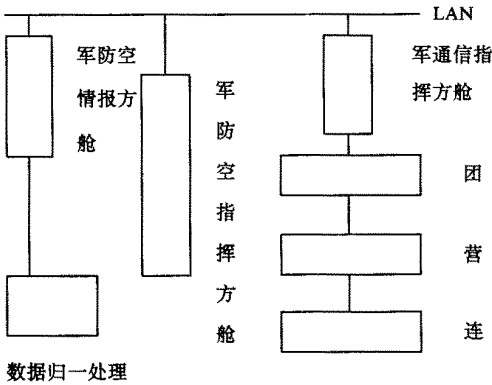


图5 师防空指挥中心示意图

Fig. 5 Aerial defence commanding center of the division

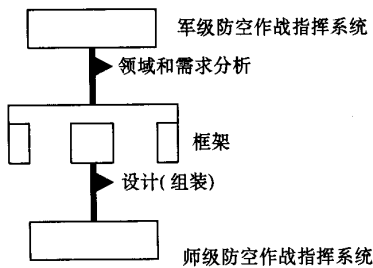


图6 师级防空作战指挥系统开发过程示意图

Fig. 6 Exploitation process of aerial defence combat commanding system

## 5 BFD 的研究重点和最新发展趋势

BFD 的研究更多的集中于特定的企业领域如制造业、银行业、卫生系统等. 在开发方法上, 与白盒框架相比, 许多框架方面的专家更赞成黑盒框架, 白盒框架在很大程度上依靠面向对象语言的特征如继承和动态绑定( Dynamic Binding) 来获得可扩展性. 在这种框架中, 已有的功能被复用和扩展通过: ① 从框架基类继承得到新的子类; ② 重载预定义的钩子方法. 白盒框架需要应用程序开发人员对框架的内部结构非常熟悉. 黑盒框架通过给组件定义接口, 使其能通过对象组合技术嵌

入到框架中来获得扩展性. 在这种框架中, 已有的功能被复用和扩展通过 ① 一致的接口规范来定义组件; ② 使用设计模式如策略(Strategy) 将组件集成到框架中去. 与白盒框架不同, 黑盒框架使用组合代理而不是继承, 所以, 黑盒框架通常比白盒框架更容易使用和扩展. 但是, 黑盒框架开发起来更困难, 需要框架开发者定义良好的接口和钩子方法. 因为黑盒框架强调对象的动态联系而不是静态的类与类之间的联系, 所以更容易扩展和动态配置. 随着开发人员对框架认识上的深入, 越来越多黑盒框架被开发出来. 在框架记录方面, 采用引入设计模式和元模式来更进一步描绘框架, 捕获其设计策略和其中组件的合作关系. 在框架集成方面, 更注重将框架与类库, 组件和设计模式等联合在一起使用, 更注重对框架继承性的研究.

软件复用预示着更好、更快、更方便的软件开发过程, 有利于提高软件生产力, 可以为软件生产产业提供有效的支持. 面向对象是一种强有力的软件开发方法, 可为软件复用提供基本的技术保证. 只要我们在软件开发的过程中, 重视并利用它, 就一定会给软件开发带来勃勃生机. 而且随着软件系统的日益复杂, 面向对象应用程序框架变得愈来愈普遍和重要, 它们是面向对象系统获得最大复用的方式. 较大的面向对象系统将会有多层彼此合作的框架组成, 对框架的深入研究给软件开发人员提供了一个重要的工具, 以更好地实现复用, 更好地抓住那些成功的设计模式、体系结构和编程机制<sup>[1~9]</sup>.

## 参考文献:

- [1] 韩秀蓉, 李立峰. 面向对象框架技术及应用[EB/OL]. <http://game.ccidnet.com/tech/paper/>, 2000-12-12.
- [2] 杨芙清. 软件复用及相关技术[J]. 计算机科学, 1999, (5): 1~4.
- [3] 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.
- [4] 邵维忠, 杨芙清. 面向对象的系统分析[M]. 北京: 清华大学出版社, 1998.
- [5] 李克勤. 面向对象的领域分析[M]. 北京: 北京大学出版社, 2000.
- [6] 杨芙清, 王千祥, 梅宏, 等. 基于复用的软件生产技术[J]. 中国科学(E辑), 2001, 31(4): 363~371.
- [7] 张海藩. 软件工程化导论[M]. 北京: 清华大学出版社, 1992.
- [8] 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.

国防科技大学出版社,1992.

[ 9 ] 朱海滨.面向对象技术——原理与设计[ M] .长沙:

Soft ware reuse and Soft ware Development Technology Based on the  
Object oriented Framework

WANG Li -juan<sup>1</sup>, SUN Xi -chao<sup>2</sup>, DI Song -mao<sup>3</sup>, WANG Zhe -guang<sup>1</sup>, FENG Zhi -hui<sup>1</sup>, LIU Qian<sup>1</sup>

( 1. Department of Modern Education , Henan Agricultural University , Zhengzhou 450002, China ; 2. Department of Financial Affairs , Henan Agricultural University , Zhengzhou 450002, China ; 3. Department of Financial Affairs , Zhengzhou University , Zhengzhou 450002,China)

**Abstract** : In this article , the author introduces the basic concepts and the significance of the soft ware reuse , analy -  
ses the characteristics of the framework and its connection with the frequently used object -oriented reuse technolo -  
gy , such as , the class libraries , components and the designing patterns etc . , points out the development process of  
the soft ware exploitation based on the framework . Firstly , the author carries out the field analysis in a given area  
and finds out all the application program to illustrate the common class and object and to establish the field model .  
Secondly , the author presents the system requirement explanation . Then comes the design of the framework .  
Through the analysis and the design of the application program , we can obtain the special requirement of the appli -  
cation program in the field so as to expand the framework to realize this application program . The last step is the  
testing of the soft ware . The above phases are the processes of gemination and reiteration . Nowadays , the study on  
BFD mainly focuses on the specified fields , such as , industry , banking and sanitary network . As to the exploitation  
method , we mainly adopt the black -box framework , pay attention to the combinative use of the framework with the  
class libraries , groupware and the designing mode and think much of the inheritance research on the framework .

**Key words** : soft ware reuse ; object oriented ; framework ; components ; designing pattern