

软件设计模式及其应用研究

计春雷

(上海电机学院 电子信息学院, 上海, 200240)

摘要 介绍了以面向对象技术为基础的软件设计模式的基本概念及分类, 描述了各种设计模式之间的关系, 讨论了使用设计模式的正确方法, 并结合实例介绍了多种设计模式在教学信息平台开发中的具体应用。

关键词 设计模式; 面向对象; 模式使用; 教学信息平台

中图分类号 TP 311 **文献标识码** B

Study of Design Pattern of Software and its Application

Ji Chunlei

(School of Electronic Information, Shanghai Dianji University, Shanghai, 200240)

Abstract The paper introduces the basic conception and classification of software design pattern on the basis of object-oriented technology, systematically describes the relation among design patterns, discusses the correct method of using design pattern, and gives an introduction of many kinds of design patterns in the specific application of developing teaching information platform by using examples.

Key words design pattern; object-oriented; pattern using; teaching information platform

设计面向对象的软件比较困难, 而设计可重用的面向对象的软件则更加困难。必须找到相关的对象, 以一个合适的粒度作为依托, 加上通用的接口设计以及合理的继承关系, 建立起对象之间的基本关系, 为以后的编码以及软件的可重用性提供无尽的便捷。而设计模式就可以使人们更加简便地复用成功的设计和体系结构, 它将面向对象软件的设计经验作为模式记录下来, 每一个设计模式系统地命名、解释和评价了面向对象系统中一个重要的和重复出现的设计。因此, 系统分析和设计模式分析在整个

软件的开发过程中就显得尤为重要。设计既要对手头的问题有针对性, 又要对将来的问题和需求有预见性和通用性。

1 设计模式概述

设计模式是面向对象技术的“刀刃”部分。设计模式的概念最早来源于建筑学, 由建筑师 Alexander 提出来的, 它是记录、提炼存在于软件开发人员头脑中或文档中的一些反复出现的共性问题及其经过多次验证的成功解, 它表达了在特定上下文情形

收稿日期: 2006-09-03

作者简介: 计春雷(1964), 男, 副教授, 专业方向: 软件工程。

下产生的、反复出现的典型性问题以及相应的一整套解决方案之间的关系。设计模式关注的是特定设计问题及其解决方案,它描述了如何利用面向对象的基本概念和机制来解决可扩展的软件设计中经常出现的问题,并针对设计问题给出可复用的解决方案,而这个解决方案是对反复出现的设计结构进行识别和抽象得到的。同时,每个模式都伴有定义的语境和强度,语境解释了模式的适用情况,强度是语境中的元素。如果问题的环境与模式的语境和强度相匹配,该模式便可应用。

每一个设计模式都系统地命名、解释和评价了面向对象系统中的一个重要的和重复出现的设计。这样,只要搞清楚这些设计模式,就可以完全或者很大程度上吸收了那些蕴含在模式中的宝贵经验,对面向对象的系统能够有更完善的了解。更为重要的是,这些模式可直接用来指导面向对象系统中至关重要的对象建模问题。只要问题有相同的背景,就可以直接套用这些设计模式,从而大大减轻软件开发人员的工作量。

关于软件设计模式,目前的研究方向主要有^[1]设计模式与其他面向对象设计方法(如特定领域的框架)的关系,它们各自的优劣和适应范围。除此以外,人们还在各个方面总结设计模式(如通信领域等),以及研究如何让设计模式的使用更加自动化。

2 设计模式的组成

2.1 设计模式的基本成分

- 一个设计模式被划分为4个基本组成部分^{[2]2}。
- (1) 模式名称。一个助记名,它用一两个词来描述模式的问题、解决方案和效果。
- (2) 问题。描述了应该在何时使用模式,它解释了设计问题和存在的前因后果,描述了特定的设计问题,如怎样用对象表示算法等;也可能描述了导致不灵活设计的类或对象结构。
- (3) 解决方案。描述了设计的基本要素,它们之间的相互关系及各自的职责和协作方式。
- (4) 效果。描述了模式应用的效果及使用模式应权衡的问题。模式效果对于评价设计选择和理解使用模式的代价及好处具有重要意义。软件效果大多关注对时间和空间的衡量,它们也表述了语言 and 实现问题。模式效果包括它对系统的灵活性、扩充

性或可移植性的影响,这些效果对理解和评价这些模式很有帮助。

2.2 设计模式的描述

为了达到设计复用,必须同时记录设计产生的决定过程、选择过程和权衡过程。可用统一的格式描述设计模式,目前最常用的格式是 Erich Gamma 等^[2]采用的下面固定模式描述。

模式名和分类、意图、别名、动机、适用性、结构、参与者、协作、效果、实现、代码示例、已知应用、相关模式等13个要素。在特定的软件开发领域中,可以忽略其中某些要素,也可合并几个要素,甚至可以使用不同的描述方法来描述所采用的设计模式。

设计模式的描述方法^[3]包括自然语言描述法、统一标记语言(UML)描述法、形式化语言描述法。自然语言描述法比较简单、方便,但对现实与设计之间的过渡描述不够流畅。对象建模技术(OMT)描述法是利用类图和对象图对设计模式中的类、实例以及整体模式结构进行图形描述的方法,而UML是在OMT基础上进一步发展起来的,其描述更加清晰和统一,符合大部分软件设计人员的习惯,也便于设计人员的理解 and 应用。形式化语言主要包括 DisCo, LePUS, LayOM, ADV/ADO, CDL, PDL, PDSP 等,其中 DisCo 侧重于描述设计模式中参与者的交互行为。

3 设计模式的分类

软件设计模式的分类有不同的方法。本文主要参考文献[4]10-12来进行分类,如表1所示。

表 1 软件设计模式分类表
Tab.1 Classification chart of software design patterns

创建型		结构型	行为型
类	Factory Method	Adapter	Interpreter Template Method
	Abstract Factory	Adapter	Chain of Responsibility
对象	Builder	Bridge	Command
	Prototype	Composite	Iterator
	Singleton	Decorator	Mediator
		Facade	Memento
		Flyweight	Observer
		Proxy	State
			Strategy Visitor

设计模式依据其目的可分为创建型、结构型、行为型3种。创建型模式与创建的对象有关;结构型模式处理类或对象的组合;行为型模式对类或对象怎样交互和怎样分配职责进行描述。根据模式是用于类还是用于对象,可将其分为类模式和对象模式。类模式处理类和子类之间的关系,这些关系通过继承建立,是静态的,在编译时便确定下来了;对象模

式处理对象间的关系,这些关系在运行时是可以变化的,更具动态性。类模式只指那些集中于处理类间关系的模式,而大部分模式都属于对象模式的范畴。

还有一种方式是根据模式的“相关模式”部分所描述的它们怎样互相引用来组织设计模式。图1给出了模式关系的图形说明。

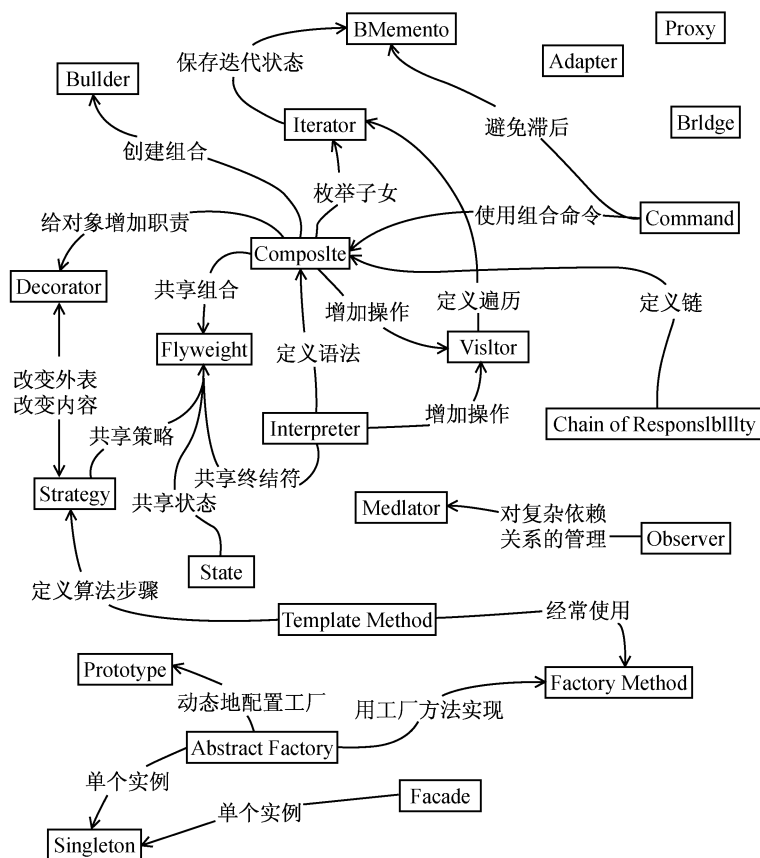


图1 23种设计模式间关系示意图

Fig. 1 Diagram of the relation among 23 kinds of design patterns

3.1 创建型设计模式

创建型模式与对象的创建有关,即描述怎样创建一个对象,它隐藏对象创建的具体细节,使程序代码不依赖具体的对象。因此,当增加一个新对象时几乎不需要修改代码即可。创建型类模式将对象的部分创建工作延迟到子类,而创建型对象模式则将它延迟到另一个对象中。创建型类模式有 Factory Method(工厂方法)模式;创建型对象模式包括 Abstract Factory(抽象工厂)、Builder(生成器)、Prototype(原型)、Singleton(单件)4种模式。

3.2 结构型设计模式

结构型模式处理类或对象的组合,即描述类和对象之间怎样组织起来形成大的结构,从而实现新的功能。结构型类模式采用继承机制来组合类,如 Adapter(适配器类)模式;结构型对象模式则描述了对对象的组装方式,如 Adapter(适配器对象)模式、Bridge(桥接)模式、Composite(组合)模式、Decorator(装饰)模式、Facade(外观)模式、Flyweight(享元)模式、Proxy(代理)模式。

3.3 行为型设计模式

行为型设计模式描述算法以及对象之间的任务

(职责)分配,它所描述的不仅仅是类或对象的设计模式,还有它们之间的通讯模式。这些模式刻画了在运行时难以跟踪的复杂的控制流。行为型类模式使用继承机制在类间分派行为,如 Template Method(模板方法)模式和 Interpreter(解释器)模式;行为型对象模式使用对象复合而不是继承,它描述一组对象怎样协作完成单个对象所无法完成的任务,如 Chain of Responsibility(职责链)模式、Command(命令)模式、Iterator(迭代器)模式、Mediator(中介者)模式、Memento(备忘录)模式、Observer(观察者)模式、State(状态)模式、Strategy(策略)模式、Visitor(访问者)模式。

4 设计模式的选择与使用

现有软件设计模式一般应用:①在软件系统设计的开始阶段就应用设计模式对软件体系结构进行设计;②在系统的体系结构设计初步完成后,对系统另有要求的组件或模块通过加入设计模式使其更加优化、灵活。由于设计模式具有一定复杂性,所以很难将其应用到具体的软件设计中,主要原因有:①软件设计人员没有正确把握和理解软件设计模式;②没有一种有效的方法来指导使用这些设计模式。

为此,在总结了一些软件设计模式的应用经验的基础上,借鉴了对面向对象软件工程过程的研究^{[4]25-28},提供一些如何选择和使用设计模式的参考。

4.1 设计模式的选择

选择合适的软件设计模式通常需要考虑以下几个方面^[5-6]:①考虑设计模式是怎样解决设计问题的;②浏览模式的意图部分;③研究模式怎样互相关联;④研究目的相似的模式;⑤检查重新设计的原因;⑥考虑设计中哪些是可变的。

4.2 设计模式的使用

一旦选择了一种软件设计模式,怎么使用它呢?这里给出一个有效应用设计模式的循序渐进的方法^{[2]20-21}。①大致浏览一遍设计模式;②回头研究结构部分、参与者部分和协作部分;③看代码示例部分,看看这个模式代码形式的具体例子;④选择模式参与者的名字,使它们在应用上下文中有意义;⑤定义类;⑥定义模式中专用于应用的操作名称;⑦实现执行模式中责任和协作的操作。

5 设计模式应用举例

结合上海电机学院教学信息平台的开发,通过对面向对象软件的各种设计模式的研究与分析,在项目开发过程中应用了多种设计模式,实现了软件设计开发的灵活性和可复用性。

(1)外观设计。在页面设计上使用了 Decorator 的设计模式,Decorator 模式描述了以透明围栏来支持修饰的类和对象的关系。浏览器的选择使用了 Abstract Factory 的设计模式,Factory(浏览器)和 Product(界面)是 Abstract Factory 模式的主要参与者,该模式描述了怎样在不直接实例化类的情况下创建一系列相关的产品对象,它最适用于产品对象的数目和种类不变,而具体产品系列之间存在不同的情况。

(2)接口设计。对于这个系统,采用的是 Strategy 的设计模式。在对象中封装算法是 Strategy 模式的目的。Strategy 模式应用的关键点在于为 Strategy 和它的环境设计足够通用的接口,以支持一系列的算法。

(3)登录模块。用户登录功能是常见的模块,在这里使用了 Facade 的设计模式,同时使系统的设计模式建立在 MVC 框架下。

(4)文件档案模块。在文件模块的设计过程中,主要使用了 Adapter, Command, Singleton, Abstract Factory 这 4 种设计模式。在本模块调用文件功能的地方,在接口上文件功能类和所需要的接口并不完全一致,需要进行转换,因此,在这个地方使用了 Adapter 的设计模式;文件的功能多种多样的,设计的时候需要考虑可扩展性,使用 Command 设计模式,使得内部功能透明;考虑到文件操作存在资源的主占和共享的问题,文件在打开时可以多人同时访问,然而在删除、上传、修改时需要保证文件当时只有一个用户使用,因此,使用 Singleton 模式保证了这一点;同样考虑到文件的访问需要读取文件的属性和他们的内容,对于不同格式的文件,文件的访问的方法各不相同,使用 Abstract Factory 模式保证了接口的统一性和可扩展性。

(5)短信模块。在短信模块的设计上,使用的是 Mediator 的设计模式。整个系统的通讯模块有

(下转第 70 页)

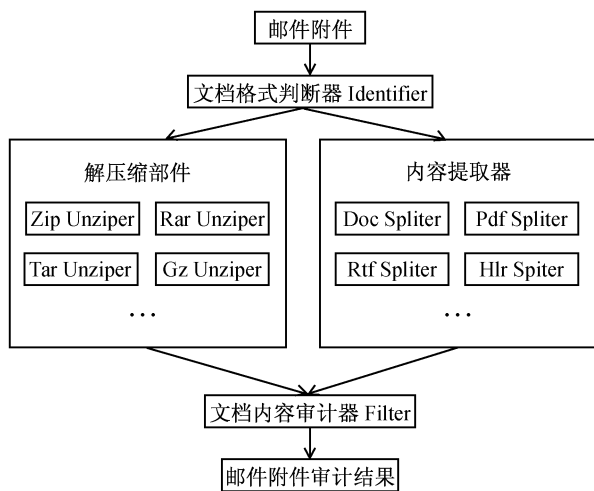


图 2 邮件附件的深层次过滤示意图

Fig. 2 Diagram of deep filtering of mail annex

3 结 语

在基于 Linux 内核的防火墙上,利用 NetFilter/Iptables 的协议重定向功能,以透明代理的方式实现了邮件过滤功能,并与认证模块相结合实现了

基于用户的规则的细粒度的、灵活的邮件过滤。同时在邮件代理内部,通过文档格式判断器(Identifier)、文档内容提取器(Splitter)、文档内容审计器(Filter)、解压缩部件(Unziper)等组件的引入,实现了对于邮件附件的深层次的审计。

参考文献

- [1] 袁津生,吴砚农. 计算机网络安全基础[M]. 北京: 人民邮电出版社, 2002.
- [2] Postel Jonahan B. Simple Mail Transfer Protocol[EB/OL]. (1982-08-10)[2005-03-22]. <http://www.ietf.org/rfc/rfc0821.txt>.
- [3] Myers J, Rose M. Post Office Protocol Version 3[EB/OL]. (1996-05-10)[2005-03-10]. <http://www.ietf.org/rfc/rfc1939.txt>.
- [4] Borenstein N, Freed N. Mechanisms for Specifying and Describing the Format of Internet Message Boded[EB/OL]. (1993-09-08)[2005-03-20]. <http://www.ietf.org/rfc/rfc1521.txt>.

(上接第 49 页)

一个 Daemon 在一直运行,相当于一个 Mediator。用户是否在线,以及新短信存在与否,这些 Colleagues 都是通过这个 Mediator 与 Client 发生交互的。

(6) 留言板模块。在留言板的模块设计上,使用的是 Composite 的设计模式。将对象组合成树形结构以表示“部分-整体”的层次结构,Composite 使得用户对单个对象和组合对象的使用具有一致性。

6 结 语

过去几年中,面向对象领域取得的最大进展就是设计模式概念的提出和发展,设计模式由于其实用性而受到欢迎,在表达上既简单又清楚而受到越来越多的重视。本文通过对面向对象软件工程的研究,总结了软件设计模式应用的一般规律,描述了设计模式在设计可重用性面向对象软件中的重要性、各种软件设计模式的概念和它们之间的关系,以及如何选择和使用这些模式。有效地应用软件设计模式开发面向对象软件系统,既可以提高软件设计

效率及完善设计,也可以优化系统的性能,提高软件设计的灵活性。随着面向对象理论的发展,软件设计模式会得到进一步的充实和发展。

参考文献

- [1] 楚荣珍,刘建国. 软件体系结构设计模式的分析研究[J]. 计算机系统应用, 2005(9): 48-51.
- [2] Gamma Erich. 设计模式:可复用面向对象软件的基础[M]. 李英军, 马晓春, 蔡敏, 等, 译. 北京: 机械工业出版社, 2000.
- [3] 刘海岩,锁志海. 设计模式及其在软件设计中的应用研究[J]. 西安交通大学学报. 2005, 39(10): 1043-1045.
- [4] Gamma, E, Helm R, Johnson R, et al. Design Patterns: Elements of Reusable Object-oriented Software[M]. Boston: Addison-Wesley, 1995.
- [5] Jean-Marc Jezequel, Michel Train, Christine Mingins. Design Patterns and Contracts[M]. Boston: Addison-Wesley, 2000.
- [6] 邹娟,田玉敏. 软件设计模式的选择与实现[J]. 计算机工程, 2004, 30(10): 79-81.