文章编号:1005－0523(2005)01－0099－04

从静态库到通用组件接口——复用与设计模式

雷 强[1],陈红丽[2],曾润忠[2]

(华东交通大学 1.信息工程学院;2.高职学院,江西 南昌,330013)

摘要:阐述了软件开发中复用的各类方法,比较各自的优劣.配合一个软件开发中的实例,提出设计模式实现了设计的复用,提高了软件的灵活性和健壮性.

关 键 词:复用;模式;库;组件

中图分类号:TP393 **文献标识码**:A

软件开发面临着新的变革.生命周期的缩短和客户需求的迅速变化要求软件的开发速度更快,普遍开发一个软件的计量时间从年缩短到了月.应用环境从以前的单机演进到网络继而发展到跨平台的环境,环境中变动的因素越来越多,风险加大.这些都使得传统的软件开发模式需要改变.

1 复用是提高开发效率和质量的最直接的方法

软件开发的效率始终是个难以评估、难以提高的因素.因为软件开发主要靠人的智力进行,自动化程度很低,国际上软件开发人员的月平均代码行为 300 行.为提高软件的生产效率,单纯提高人均代码产出量是不现实的.

从软件工程的角度出发,复用是最直接提高开发效率和质量的方式.学术界和产业界都对复用进行了广泛研究与实践,其中 Ivar Jacobson 提出了在软件过程中贯彻复用的思想[1],Carma McClure 从工程角度总结了多种复用技术和具体实施方法[2].伴随着软件技术的发展,复用在软件开发中的应用也越来越深入,已成为国际上软件过程标准的重要部分.比如 IEEE 1517(信息技术－软件生命周期过程－复用过程标准)[3]和 CMM(软件成熟度模型).

2 从低级到高级——代码复用到对象复用

从认识到复用的重要性起,人们就开始摸索如何在软件开发中实现最大程度的复用.早期开发语言比如 C 是面向过程的语言,为了实现软件的复用,研究者提出模块化的设计思想.自顶向下软件分为许多模块,模块可以被重复调用.对 C 而言,模块就是一个函数.

这样的设计思路要把系统的所有功能彻底划分成一个个模块,以后的开发可以把模块复制进来.如此,设计的难度和开发的强度仍旧很大.人们进一步把公用的函数都抽取出来,编译成库文件,供不同的软件在链接中使用.由于这是静态的代码编译,仍属于简单的代码复用.

面向对象的思路是把代码和相关数据封装到对象中,由一个个自治的对象构成软件.这种思路符合人认识世界的观点,也革新了软件开发模式.面向对象技术支持继承、封装、多态等特性,复用的范围更广、方式更灵活了.

比如继承,子类自动拥有父类的方法和属性.由于继承可以有许多层,写代码时就可以把最基本的方法和属性放在顶层,留下功能扩展的空间给下

收稿日期:2004－09－08

作者简介:雷 强(1975－),男,江西南昌人,华东交通大学硕士研究生.

层的子类.这样的处置使得要复用代码只需简单地声明继承某层的类即可.

又如多态,在运行时根据环境不同可以加载不同的对象.子类都可以看成是父类型的,那么加载任何子类代替父类都不需要编码逻辑上改变.这样复用的灵活性大大增强了.

3 模式——面向对象到标准化组件的桥梁

对象是一个个自治的单元,通常完成的功能简单,但是类别繁多.面向对象模式设计的软件特别适合变化多或快的环境,这时,软件中的对象体系趋向细化、分散化和分布化.

软件工程中有个概念"粒度",对象的粒度很小,使用不易.如何把小粒度的对象组合成大粒度的单元? 组件技术就应运而生了.

组件通常是由对象打包编译成的独立软件单元,它比对象的独立性强,内部通常封装了配置单元、接口单元,有标准化的编码规格,常与框架一同构成设计的宏观层面.

组件技术强调的是组件的易于使用、便于开发等特性,因此所需的设计技术比如模式也得到了很大发展.现在的组件技术主要有 EJB 和 COM 等等,都不同程度地使用了模式,特别是 EJB 在制定标准时就已经贯彻了模式思想,在 SUN 官方网站上有专门介绍 EJB 模式的白皮书.

4 开发实践中的模式应用

模式是人们在开发过程中遇到类似问题而归纳出的一套通用设计方案.模式已经总结出了几十种,并还在不断推出.常用的模式有约七八种,其中工厂模式是创建模式的基础.工厂模式又有三种:简单工厂、工厂方法、抽象工厂[4].简单工厂最为简单.下面我们看一个实际开发中简单工厂的例子.

4.1 简单工厂方式的框架类设计

在深圳市地方税务局登记分局网上登记系统[5]的项目中,前台客户端取得数据有很多种方式,有可能是从 Ms SqlServer 数据库取得用户管理数据或者是从 IBM AS/400 DB2 数据库中取得业务数据,还可能从内存缓存中取得代码表.如果都由客户端负责建立连接,显然客户端负担太大,客户端应关注商业逻辑而不是它的实现.

毅兴公司(深圳)使用了自主开发的系统框架,连接数据库部分由框架封装在一个 jar 包 utils.jar 中.其相关的类图如下:

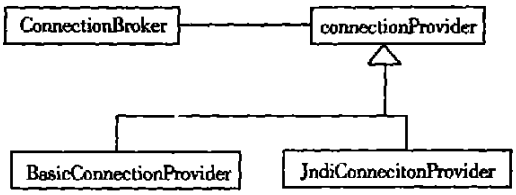

图 1 数据库连接类关系图

图中 ConnectionProvider 是定义了 getConneciton()方法的接口,BasicConnectionProvider 类和 JndiConnecitonProvider 类实现了这个接口.客户端创建数据库连接的责任被转移到工厂类 ConnectionBroker,而这个工厂类通过生成产品类 XXConnectionProvider 实现真正的连接.

4.2 异质数据库连接接口方式的代码实现

在客户端代码中连接数据库部分都是用以下一条语句:

```
con = ConnectionBroker.getConnection();
```

在 getConnection()方法中,工厂类 ConnectionBroker 生成产品类(Product)[4],如上图中的 BasicConnectionProvider 类和 JndiConnecitonProvider 类.由生成的具体产品类返回数据库连接(Connection).

类装载机制(Classloader)[6]被工厂类用来在运行时(Runtime)生成产品类.代码如下:

```
provider = (ConnectionProvider) Class.forName
(providerClass).newInstance();
```

而在 ConnectionBroker 类初始化过程中读取了配置文件,以确定将要生成的产品类[4]

```
ConnectionBroker.initialize(myFile.getProperties
(ConstantName.DBPROPERTYNAME));
```

4.3 数据库连接的配置方法

ConnectionBroker 类初始化中用到常数类 ConstantName,其属性 DBPROPERTYNAME 是指向文件 dbtjf.properties 的.

```
public static final String DBPROPERTYNAME =
"/dbtjf.properties";
```

dbtjf.properties 是按 JAVA 数据结构"属性"(Properties)的规定书写的文件[6],其内容中包含了数据库驱动程序名称、用户名及密码等全部数据库连接的信息.典型的如下

```
jdbc.driver = com.microsoft.jdbc.sqlserver.
SQLServerDriver
jdbc.url = jdbc:microsoft:sqlserver://localhost:
```

```
1433; SelectMethod = cursor; SendStringParametersAsUnicode=true;DatabaseName=szdsdjfj
jdbc.user = sa
jdbc.password =1111
dblib=
#如果在生产环境下,要将 weblogic 下的此项设置 taxdblib=TAXDTA.
taxdblib= TAXDTA.
# jdbc.driver = com.ibm.as400.access.AS400JDBCDriver
#jdbc.url =jdbc:as400://192.168.3.2
#jdbc.user =ggll001
#jdbc.password =ggll001
#dblib=WSDJ.
#taxdblib=TAXDTA.
provider.class =com.iaworld.util.jdbc.providers.basic.BasicConnectionProvider
pagesize=10
```

表 1　数据库连接配置文件 dbtjf.properties

表 1 中#开头的部分将被程序忽略.

这样,数据库连接部分即作为一个通用的组件接口存在,系统可以引用任何外部的驱动程序.开发人员可以通过书写 dbtjf.properties 数据文件,系统将完成具体驱动程序的加载和数据库的连接设置.无论连接是通过 JDBC 驱动程序还是连接池、是 MS SQL SERVER 还是 IBM DB2,也无论请求来自前台的 WEB 服务器还是后台的应用服务器,都可以通过不同的配置文件[7]正确地连接到数据源.改变数据库连接方式和类型只须改变配置文件,不需要重新编译和发布.由此,系统获得了极大的效率和灵活性.

5　结　论

模式是复用的高级阶段,是设计上的复用,它是最新的组件开发模式中不可或缺的一个组成部分.研究和应用模式对提高开发人员的软件复用水平及层次有重大的意义.

参考文献:

[1] Ivar Jacobson,等.软件复用:结构、过程和组织[M],北京:机械工业出版社,2003.

[2] Carma McClure.软件复用技术:在系统开发过程中考虑复用[M],北京:机械工业出版社,2003.

[3] Carma McClure.软件复用标准指南[M],北京:电子工业出版社,2004.

[4] 阎宏.JAVA 与模式[M],北京:电子工业出版社,2002

[5] 吴　斌,王德春.深圳市地方税务局网上税务登记系统需求说明书(第三版)[Z],深圳毅兴科技有限公司,2004

[6] Steven L. Halter,等.Java 技术精髓[M],北京:机械工业出版社,2002.

[7] Cbuck Cavaness.Jakarta Struts 编程[M],北京:清华大学出版社,2004.

From Static Library to Common Components' Interface ——Reuse and Design Patterns

LEI Qiang[1], CHENG Hong-li[1], ZENG Run-zhong[2]

(1.School of Information Engineering; 2.High Occupation Technology College, East China Jiaotong Univ., Nanchang 330013, China)

Abstract: The article describes different methods of reuse in software development, and makes a comparison on these methods. With an instance of a project practice, it concluded that design pattern realizes reuses design and, improves software's flexibility and robustness.

Key words: reuse; patterns; library; components