业务应用软件框架的一种分析方法*

何克清[1]，应 时[1]，田中茂[2]，冈本泰次[2]

[1](武汉大学 软件工程国家重点实验室，湖北 武汉 430072)；

[2](富士通株式会社 软件事业本部 Middleware Software 事业部，日本)

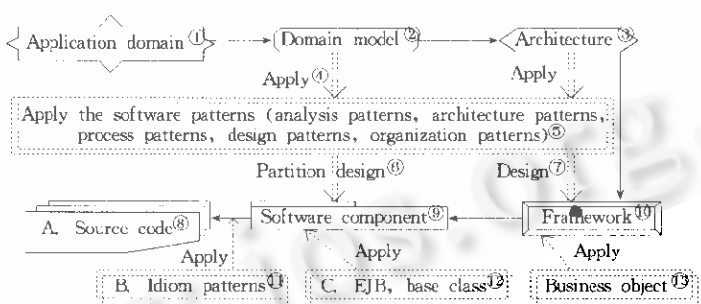
E-mail: hekeqing@public.wh.hb.cn

摘要：为了开发成熟的、可重用的软件框架和组件，提倡尽量地抽出和组入软件模式．讨论了基于软件模式的面向对象软件开发方法．在分析业务应用领域需求规格的基础上，给出了软件框架的分析方法和基本角色模型，抽出了框架的体系结构分析模式、基本角色类及其结构并设计了数据存取的"抽象工厂"模式．该方法适用于应用框架和软组件的开发．

关键词：面向对象；软件框架；组件；体系结构分析模式；设计模式；角色模型

中图法分类号：TP311 **文献标识码**：A

随着软件模式(patterns)[1,2]、面向对象的 UML 技术[3]、框架 FW(framework)和组件(componentware)[4]的部件化及其重用技术的进步，面向对象软件开发方法产生了很大的变革．图 1给出了基于软件模式的面向对象软件开发的新方法和过程．目前，该开发方法已经在世界上的大型软件企业中得到了实际运用，一部分业务应用软组件已经进入软件市场流通．可以预见，在计算机网络技术发展的今天，一个以开放型软组件及软件框架为单位的电子商务软件将成为今后软件市场的主流．



①应用领域，②领域模型，③体系结构，④应用，⑤应用软件模式(分析模式，软件体系结构模式，开发过程模式，设计模式，组织模式)，⑥分割设计，⑦设计，⑧源代码，⑨软组件，⑩软件框架，⑪经验模式，⑫EJB，基本类，⑬业务对象．

Fig. 1 The method based on software patterns for object-oriented development

图1 基于软件模式的面向对象软件开发方法

目前，软组件的开发、重用、销售和流通的单位还主要是小粒度的软件代码，如用 JBK(Java business kit，Fujitsu，Japan)，EJB(enterprise Javabeans)开发的 Beans 等．为了提高软件的重用效

* 收稿日期：2000-07-12；修改日期：2000-12-04

基金项目：国家教育部科学技术重点资助项目；武汉市科技攻关项目(20011001001)

作者简介：何克清(1947－)，男，湖北荆门人，博士，教授，主要研究领域为面向对象的软件工程方法论，软件开发环境，软件模式，软件框架；应时(1965－)，男，湖北武汉人，博士，副教授，主要研究领域为软件开发方法，组件重用；田中茂(1948－)，男，日本东京人，参事，主要研究领域为计算机软件；冈本泰次(1949－)，男，日本神奈川人，副参事，主要研究领域为计算机软件．

率、降低开发成本,在不同的应用领域,分别开发大粒度可重用的软件框架和组件是十分必要的.但是,如何开发成熟且便于重用的软件框架和组件,而不仅仅是简单地提供源代码,是大家所关心的技术问题,其开发方法和案例的研究已为软件界所关注.IBM 公司的"旧金山研究开发课题"以及 Taligent 等公司所提供的应用软组件,其公共软件框架给予了我们很大的启示,但是它们并没有给出系统地分析与设计软件框架的方法.Relph Johson 等人提出了使用软件模式开发软件框架的思想[4],但是他们同样也没有给出基于软件模式的一个应用领域软件框架的分析与设计方法及其相应案例.

我们认为,为了开发成熟的、可方便重用的软件框架和组件,必须组入软件模式.本文给出了业务应用(business application)领域的软件系统基本框架的分析和设计过程;使用角色模型化(role modeling)方法,抽出了框架的体系结构分析模式;分析和设计了业务应用软件系统框架的角色模型,设计了角色类(role class)的基本责任和操作;此外,还设计了框架的数据存取"抽象工厂"(abstract factory)模式.

1 基本概念

(1) 软件框架:为了解决某类问题(系统或子系统领域)的一组抽象类的集合及其实例(instance)对象间交互协调行为的可重用的设计与实现方案[4,5].即软件框架由抽象类的集合、交互的实例对象群以及相关的实现代码组成.软件框架是设计一个问题领域的软件系统的骨架,其对象模型和代码是可重用的.软件框架分为以下 3 种:① 应用(application)框架:面向应用领域中应用系统的骨架,但它并不提供完整的应用软件系统的全部;② 支撑(support)框架:不是应用系统的骨架,而是软件框架的支撑部分;③ 基础(foundational)框架:不是应用系统的骨架,而是应用系统共同使用的公共框架,如 CORBA 等.

(2) 白盒框架:其抽象的类集向用户公开,用户生成抽象类的子类,经修改扩充后使用.本文主要讨论白盒框架的分析设计方法、角色模型、软件模式及其事例.

(3) 黑盒框架:提供基本完成的软组件.用户组合这些组件,修改其界面,构筑自己的应用系统.

(4) 软件模式:表示软件开发中可重用的解决方案(solution)所包含的经验和知识,本质上是一种抽象的概念性软件对象模型.与框架不同的是,它本身一般不提供具体的实现.

为了开发成熟的软件框架和组件,在每一个开发步骤中组入软件模式是很必要的.我们提倡的面向对象软件框架和组件的开发过程如图 1 所示.根据应用领域模型,使用软件分析模式来分析和设计多个软组件;根据领域的体系结构(architecture),使用体系结构模式和设计模式来设计软件框架;一个软件框架可以含有多个组件,使用程序设计经验(idiom)模式来实现这些组件.因此,成熟的软件框架的一个技术标志为是否在其中组入了多个软件模式.此外,在软件框架和组件的实现中,我们还应该尽量使用可重用的成熟的业务对象(business objects)、EJB、基本类等.

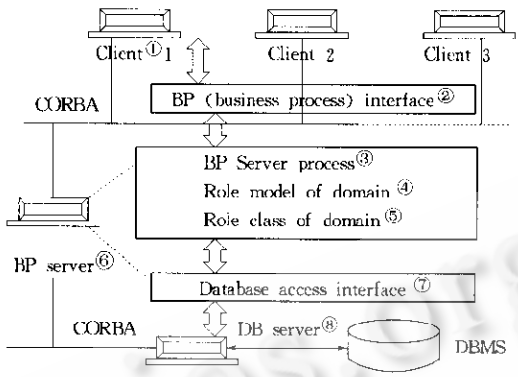
2 业务应用软件系统框架的基本角色模型和体系结构分析模式

我们根据分布式业务应用软件的多客户/服务器系统结构,确定软件框架的设计范围,分析其角色模型,定义角色类,抽出模式,利用已有的模式来分析设计软件框架和组件.

2.1 分布式业务应用软件系统框架的范围

图 2 给出了一般的业务应用软件 C/S 系统的基本构成,作为该领域的软件基本框架,至少包

含BP界面(business process interface)、BP服务处理(BP server process)和数据库存取界面(DB access interface)这3个组件,我们将它们确定为该软件框架的分析设计对象和范围.



①客户,②业务过程界面,③BP(业务过程)服务处理,④领域的角色模型,⑤领域的角色类,⑥BP服务器,⑦数据库存取界面,⑧数据库服务器.

Fig. 2 Base construction of business application framework
图2 业务应用软件框架的基本构成

2.2 业务应用软件框架的基本角色模型和体系结构分析模式

在本文中,框架的分析和定义使用角色模型化方法.在分析和定义框架过程中,我们给出以下原则:(1) 领域中的诸多业务应用系统能够共用的业务用例(use-case)作为分析和设计软件框架的功能对象;(2) 角色类的定义应该方便界面的定义;(3) 尽可能地抽出软件模式,或尽可能地适用于已有的软件模式.

2.2.1 业务应用软件框架的基本角色模型

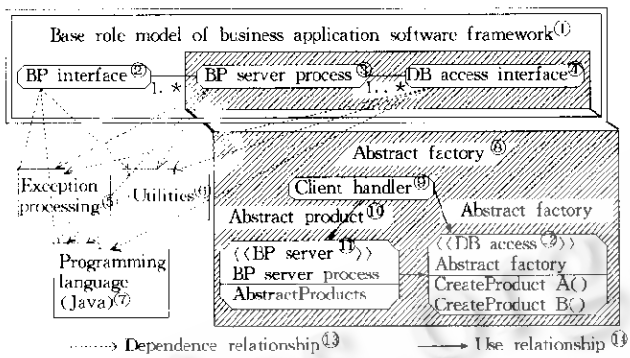
如图3所示,业务应用软件系统中可以共用的体系结构分析模式由两部分组成,即框架的基本角色模型及其所依赖的3个软件包.我们针对诸多业务应用系统能够共用的用例,诸如查询(BP select)、更新(BP update)等的要求,分析它们的处理过程,抽出处理过程中共用的角色对象;使用角色模型化方法,构造软件框架的基本角色模型.角色模型是框架的核心,它由3个角色组件构成:(1) BP界面(BP interface)角色组件,为了描述和提供由多个画面或状态组成的业务用例界面的角色组件;(2) BP服务处理(BP server process)角色组件,描述在一个画面或状态中可执行业务用例的服务处理角色组件;(3) 数据库存取界面(data access interface)角色组件,描述数据库存取的角色组件.

由于BP界面角色组件是描述由多个画面或状态组成的业务用例,而BP服务处理角色组件是描述在一个画面或状态中可执行的业务用例单位,所以它们之间存在着1对多的协调关系.同样,BP服务处理角色组件和数据库存取界面角色组件之间也存在着1对多的协调关系.BP服务处理角色组件控制系统的数据操作、BP界面及其相互之间的行为协调.它为系统的业务用例提供用户输入事件处理的Handle.数据库存取界面角色组件担负着业务应用软件系统中数据的存取、生成、更新等主体角色.

2.2.2 业务应用软件框架的体系结构的分析模式

我们抽出软件框架的基本角色模型和3个软件包及其它们之间的依赖关系,定义为如图3所示框架的体系结构分析模式.基本角色模型为软件框架的分析和设计提供了核心的角色组件和角色类以及它们之间的协调关系.但是,角色类实例对象的运行和处理必须依赖于软件包"程序设计

语言(如 Java)"、"例外处理"和"实用程序",其中"程序设计语言"是最基本的软件包.这些软件包提供了公共的软组件,有利于业务应用软件框架的设计与实现.



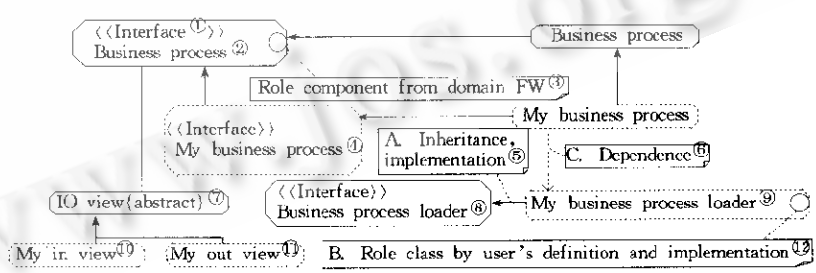
①业务应用软件框架的基本角色模型,②业务过程界面,③业务过程服务处理,④数据库存取界面,⑤例外处理,⑥实用程序,⑦程序设计语言(如Java),⑧抽象工厂设计模式,⑨客户处理器,⑩抽象产品,⑪BP服务器,⑫DB存取,⑬依赖关系,⑭使用关系.

Fig. 3 Base role model and analysis pattern of architecture for business application framework
图3 业务应用软件系统框架的基本角色模型和体系结构分析模式

为了提高软件框架的可扩充性和可重用性,我们在分析和设计过程中使用了"抽象工厂"模式[1].当分析框架中的 BP 服务处理的角色组件、数据存取的角色组件以及它们之间的协调行为时,我们设定 BP 服务处理的角色类为一个抽象的数据服务产品,而实际的数据操作由数据存取的"抽象工厂"角色类来实现.

3 BP 界面的角色模型

BP 界面与显示画面及其状态转换的设计密切相关.例如,订货(order)登录处理的业务过程包含一连串的画面及其状态转换:顾客的查询→订货输入/检查→订货的登录/确认.如图 4 所示,框架的 BP 界面角色模型主要包括 4 个角色类,即⟨⟨Interface⟩⟩类型的 Business Process,IO View{abstract},Business Process Loader 和 Business Process.



①界面,②业务过程,③领域FW提供的角色模型,④我的业务过程,⑤继承、实现,⑥依存关系,⑦输入输出视点,⑧业务过程装载,⑨我的业务过程装载,⑩我的输入视点,⑪我的输出视点,⑫用户定义及实现的角色类.

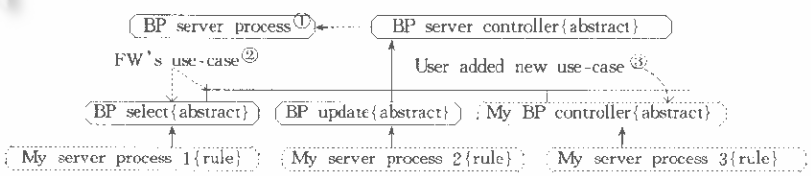
Fig. 4 Role model of BP interface
图4 BP界面的角色模型

图 4 给出了业务过程抽象的角色类 Business Process 的公共可重用操作:startSession(),commitSession(),abortSession()和 isOnSession(),其中 Session()是业务处理的执行单位.用户的 My Business Process 角色类在继承抽象角色类 Business Process 的操作和属性的前提下,定义自己的业务用例,如商品查询、商品更新、商品登录、商品删除等.框架应该预先定义和设计一些基本的

My Business Process 角色类提供给用户,即为框架对外公开的业务处理的可重用单位,当然也允许完全由用户自行定义.我们在分析业务用例的 BP 界面时,使用 IO View 角色类来抽象地描述 BP 的界面.因此,业务过程角色类 Business Process 的公共可重用的操作还应包括调用 BP 服务处理角色组件的 getOutView(IO View)操作等.

4 BP 服务处理的角色模型

BP 服务处理角色模型描述一个画面或状态中执行的业务服务处理单位.例如,商品查询、商品更新、商品登录、商品删除的服务处理等,它们分别用来定义业务服务处理的内容.在如图 5 所示的角色模型中,我们设计了 BP Server Process 角色类中可重用的操作为 getInView()和 getOutView(IO View).BP Server Controller 是 BP 服务处理角色类的子类,也是框架的核心控制角色类.业务应用软件框架提供的每个业务用例都可以定义一个 BP Server Controller 类.在应用软件框架时,用户也可以定义自己的 My BP Server Controller 类.作为 BP Server Controller 子类的设计,我们在框架中提供了两个角色类:BP Select,与查询和检索有关的业务处理;BP Update,与更新有关的业务处理.对于这两个业务处理的角色类,我们设计了可重用的操作:check(),输入检查;Furiwake(),根据输入值、切换 DB 检索条件;CheckForData()(在 BP Update 的情况下),DB 一致性检查、DB 更新、其他 DB 检查;Edit(),编辑;getInView(),getOutView(IO View).



①业务过程服务处理,②FW提供的用例,③用户追加定义的新用例.

Fig. 5 Role model of BP server process
图5 BP服务处理的角色模型

5 数据存取界面的角色模型

我们设计了如图 6 所示的数据存取界面的角色模型.模型主要由 DB View Factory 和 SQL 语句角色类构成.

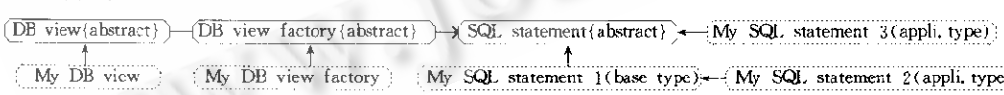


Fig. 6 Role model of data base access interface
图6 数据存取界面的角色模型

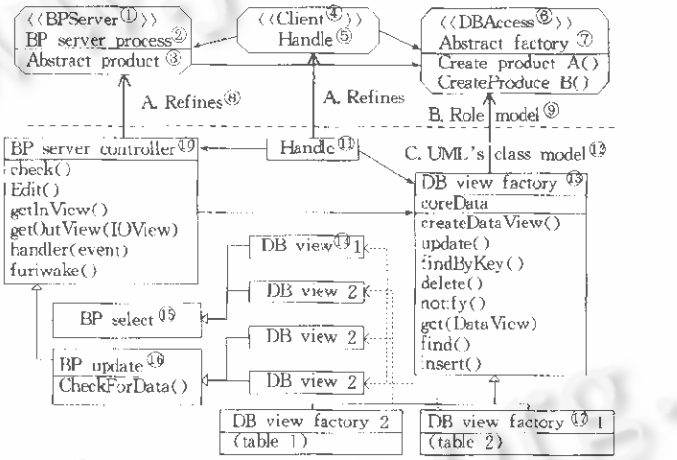
DB View Factory.这个角色类使 DB View 的数据实例对象可独立于 DB 中的数据,使得 BP 服务处理不必关心数据的区别.它控制数据的存取、生成和组合.对应于 DB 中的每种表,我们设计 DB View Factory 的一个子类.数据表的存取通过 DB View Factory 及其子类控制执行.根据 DB 检索要求,以 DB View 的实例对象或向量对象(结果为 n 件时)的形式返回检索结果.DB View Factory 管理所有的检索结果 DB View 对象.我们设计 DB View Factory 的操作包括:createDB-View,建立 DB View,在新登录记录时生成空的 DB View 实例对象;findByKey,根据关键字进行检索;findByKeyWithLock,根据关键字进行检索(带锁);find,根据关键字进行 n 件检索;insert,在 DB 中登录新的 DB View 实例对象;update,在 DB 中更新已有的 DB View 实例对象;delete,在 DB

中删除与已有的 DB View 实例对象相对应的记录.

SQL 语句角色类.它是含有 SQL 语句的角色类,分为基本 SQL 和应用 SQL 两种.我们设计基本 SQL 语句角色类的操作为 findByKey,findByKeyWithLock,find,insert,update,delete.我们还规定,只有 DB View Factory 才能调用这些操作.应用 SQL 语句类有基本 SQL 语句类所不能包括的复杂检索,如含有不等号的检索条件或 Join 检索行为等.我们设计的应用 SQL 语句角色类的操作为 find.

6 数据存取的"抽象工厂"模式

为了分析设计框架的 BP 服务处理和数据存取之间的协调结构,如图 7 所示,本节讨论数据存取的"抽象工厂"模式设计[11].其目的有以下几个:(1) 使框架的设计从数据存取、生成、组合等功能设计中分离出来;(2) 提供可供选择的多个数据存取的功能单位;(3) 提供使用一连串的关联数据存取的功能单位,如一连串的表存取等;(4) 在提供数据存取、生成、组合的功能时,只想公开界面,不想公开实现;(5) 方便框架的更新与扩充.



①业务过程,②业务过程服务处理,③抽象产品,④客户,⑤处理器,⑥数据库存取,⑦抽象工厂,⑧求精,⑨角色模型,⑩业务过程服务控制器,⑪处理器,⑫UML的类模型,⑬数据库视点工厂,⑭数据库视点,⑮BP查询,⑯BP更新,⑫数据库视点工厂.

Fig. 7 UML's class model of abstract factory pattern for DB access
图7 数据存取"抽象工厂"模式的UUML类模型

在图 7 上部所示的"抽象工厂"角色模型中,定义 BP 服务处理角色类为抽象的数据产品,数据库存取角色类为数据的抽象工厂.Client Handle 使抽象工厂生产所需要的抽象数据产品.图 7 的下部给出了求精"抽象工厂"角色模型的 UML 类结构设计.

(1) 模式的结构元素.Abstract Factory.是一个角色类,定义了 DB View Factory 的数据生成的作用.DB View Factory 为 Abstract Factory 的子类,定义了生成 BP Server Controller 的子类 BP Select,BP Update 等的操作界面.DB View Factory 1 和 DB View Factory 2:在框架中,数据和数据表的存取通过 DB View Factory 进行.我们对应于 DB 中的每个表,如表 1 和表 2 分别设定一个 DB View Factory 1 和 DB View Factory 2 子类.BP Server Process 为一个抽象的角色类,定义了 BP Server Controller 的抽象数据产品的作用.BP Server Controller 为 BP Server Process 的子类,定义了检索业务用例"BP Select"、更新业务用例"BP Update"等的抽象界面.定义用户输入事件处理 Handle,以调用 DB Access Service.DB View:根据用户对 DB 存取的要求,以 DB View 的

实例对象或向量对象(结果为 n 件时)返回存取的结果.DB View Factory $i(i=1,2,\ldots)$管理所有的存取结果 DB View $i(i=1,2,\ldots)$的实例对象.Handle 是一个业务应用软件系统框架中 Use-Case 的利用者,BP Sclect 和 BP Update 分别有不同的 My Handle.用户输入事件启动 Handle.

(2) 模式的协调关系.在本模式中,作为 Concrete Factory 的 DB View Factory 1 和 DB View Factory 2 实例对象在执行时生成,而且该实例对象可以生成所要求的 Concrete Product 1 和 Concrete Product 2,即 DB View $i(i=1,2)$数据对象.如果要求生成其他 Concrete Product j,用户必须生成并定义其他 Concrete Factory j 对象.

(3) 模式的特点.使用框架的数据存取、生成和组合的"抽象工厂"模式,能够给该软件框架的设计和重用带来以下优点:

① 实例对象生成的局部化

利用者能够简单地使用框架中的 DB View Factory 类生成对象.这是由于在本模式中隐蔽了数据对象(DB View)的生成与组合过程,能够使 Handle 与实现数据对象分离开来,而且数据对象(DB View 1,2,...)的名字被局部化在 Concrete Factory 类的实现中,所以 Handle 只需通过 DB View Factory 的抽象界面操作各自的例示对象即可.

② 能够容易地变更数据对象(DB View)的集合

在应用软件框架时才进行 Concrete Factory 类的实例化,所以变更 Concrete Factory 对象是容易的.我们只要变更 Concrete Factory 对象,就可以构成不同的数据对象(DB View)集合.也就是说,Concrete Factory 对象可以定义和生成我们所需要的数据对象.

③ 增强数据对象间的完整性

我们应用"抽象工厂"模式的约束能力,即 Concrete Factory 与数据对象的对应生成能力,能够有计划地设计系统中的数据对象集合,增强了数据对象之间的完整性.

(4) 问题和对策.因为 Abstract Factory 类的界面在某种程度上决定了所生成的数据对象的集合.为了增加新种类的数据对象,必须修改 Abstract Factory 类的界面.而修改 Abstract Factory 类的界面,还必须修改 Abstract Factory 的所有子类的界面.因此,扩充新的数据类是一个麻烦的问题.作为一种改进的方法,我们在生成对象的操作中附加参数(如类的识别 ID、整数、字符串等),即在 Abstract Factory 类中增加带有所生成数据对象种类参数的 Make 操作.

7 总结以及今后的课题

本文讨论了基于软件模式的面向对象软件开发方法与过程.为了开发成熟的软件框架和组合件,我们提倡尽量将软件模式抽出和组入,以保证业务应用系统的软件框架的成熟度以及大粒度的可重用性.本文给出了业务应用软件系统框架的分析方法和过程,抽出了框架的体系结构分析模式,设计了软件框架基本角色模型的 3 个组件模型及其类结构,给出了组件之间的协调关联,同时还设计了框架中数据存取的"抽象工厂"模式.

作为今后的课题,业务应用软件系统中 BP Server Controller 的子类(即业务用例的抽出和标准化)、软件框架中组入业务对象的方法、软件框架的组合和分解、支撑框架等问题还有待进一步研究.

References:

[1] Gamma, E., Helm, R., Johson, R., *et al*. Design Patterns, Elements of Reusable Object-Oriented Software. Tokyo:

Soft-Bank Publishers, 2000. 95~104.

[2] Buschmann, F., Meunier, R., Rohnert, H., *et al*. Pattern-Oriented Software Architecture: a System of Patterns. Tokyo, Toppan Publishers, 1999. 118~140.

[3] OMG. OMG Unified Modeling Language Specification (draft), Version 1.3 alpha R5. 1999. 22~99. http://www.omg.org/library/issuerpy.htm.

[4] Roberts, D., Johnson, R. Evolving frameworks: a pattern language for developing object-oriented frameworks. Technical Report, University of Illinois, 1997. 3~19.

[5] Woolf, B. Framework development using patterns: developing the file reading. In: Fayad, M., Schmidt, D., Johnson, R., eds. Object-Oriented Application Framework: Object-Oriented Frameworks at Work. New York: John Wiley and Sons, 1999. 9~43.

An Analysis Method of Business Application Software Framework*

HE Ke-qing[1], YING Shi[1], TANAKA Shigale[2], OKAMOTO Taiji[2]

[1](*State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China*);

[2](*Middleware Software Division, Software Group, Fujitsu Limited, Japan*)

E-mail: hekeqing@public.wh.hb.cn

Abstract: For developing mature software frameworks and components, it is necessary to elicit and incorporate software patterns in the process of software development. Based on software patterns, the object-oriented software development method is discussed in this paper. Based on the analysis of requirement specifications for business application domains, an analysis approach and a basic role model of software framework are presented in this paper. An analysis pattern of framework architecture, basic role classes and their structure are elicited, an abstract factory pattern for data access is designed. This method can apply to developments of application software frameworks and components.

Key words: object oriented; software framework; component; analysis pattern of architecture; design pattern; role model

* Received July 12, 2000; accepted December 4, 2000

Supported by the Key Sci-Tech Project of the Ministry of Education of China; the Key Sci-Tech Project of Wuhan Province of China under Grant No. 20011001001