

一种支持变化性的构件模型 JBCOM/E

张文娟, 赵俊峰, 谢冰, 杨芙清

(北京大学计算机科学技术系, 北京 100871)

摘要: 在基于构件的软件开发中, 构件模型是基础, 领域变化性是关键因素之一. 为此, 本文在原有构件模型的基础上, 提出了抽象构件的概念, 使之能对一族具有领域共性和变化性的构件进行定义与描述. 文中就抽象构件的定义、规约语言描述以及相应的构件组装和构件管理等问题进行了讨论, 定义了一种支持变化性的构件模型 JBCOM/E, 并给出了一个应用实例 青鸟 POS 领域平台.

关键词: 软件复用; 软件构件; 领域工程; 构件模型

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2003) 06-0899-04

A Component Model to Support Variability

ZHANG Wenjuan, ZHAO Junfeng, XIE Bing, YANG Fuying

(Dept. of Computer Science & Technology, Peking University, Beijing 100871, China)

Abstract: In the component-based software development method, component model is the basis and domain variability feature is the critical factor. From this, on the basis of the existing component model, this paper proposed a new concept, abstract component concept, which can be used to express a set of components that have domain commonness and domain variability. The corresponding techniques such as component description language, component composition and component management was also discussed. This variability supported component model was called JBCOM/E, which had the ability to describe domain variability. At the end, a case study example, POS Domain Software Development Platform, was demonstrated. It can be used to make new POS systems quickly through the above techniques. Sum up the above, aimed at domain variability, this paper did research on variability supported component model, component composition and component management. It provides a more consistent and integrated technical basis for component based software development method.

Key words: software reuse; software component; domain engineering; component model

1 引言

软件复用是解决软件危机、提高软件开发效率和质量的现实可行的途径^[1]. 软件被广泛地应用在各领域中, 而同一领域内的若干系统既具有共性, 又有差异性, 且同一系统在不同阶段、不同环境中均会有所变化. 在软件复用中, 领域共性为应用系统集成组装提供了基础, 变化性则确定了个体系统特性, 在一个特定的系统中领域共性与变化性共生. 所以分析、提炼领域共性, 控制变化性是成功而有效地实施复用的关键.

在基于构件的软件开发方法中, 构件模型是基础, 构件管理和应用系统的集成式组装是关键技术. 现有的比较具代表性的构件模型 Unicon^[2], Wright^[3], C2^[4,5] 等都在共同的基础上, 从不同的侧面(如组装方式、接口时序关系、动态系统配置)描述了构件的特性, 但没有对领域的变化性——这个贯穿基于构件的软件开发全过程的关键特性——进行描述. 为此, 本文在已有的构件模型的基础上加入描述领域变化性的成

分, 提出了抽象构件的概念, 用来表达一族具有共性和变化性的构件, 并就抽象构件的规约描述以及相应的构件管理和构件组装进行了研究, 定义了一种支持变化性的构件模型.

2 构件模型 JBCOM

构件模型是对构件本质特征及构件间关系的抽象描述. 构件模型关注构件的接口, 封装构件的实现细节, 是构件组装的依据和保证, 并为构件的不同生产者 and 使用者提供了一个为大家所接受的一致性描述^[6].

在 Unicon, Wright, C2 以及在分布式对象技术 CORBA、COM/DCOM 和 JavaBean/EJB 的研究中, 都涉及对构件本质特性的抽象描述问题, 即构件模型问题^[7]. 尽管它们对构件模型的定义和描述方法并不相同, 但其核心机制是一致的, 即将构件分为接口和实现两部分, 其中接口用来表示构件与外界的交互信息, 而实现用来表示可被直接复用的构件实体, 实体可以具有源代码、二进制码等多种不同的物理形态.

在青鸟工程的研究发展中,研制了青鸟构件模型 JBCOM 及其构件描述语言 JBCDL. 本文以 JBCOM 构件模型为基础,研究支持变化性的构件模型 JBCOM/E 及其相应的技术问题.

JBCOM 将构件分为原子构件和复合构件两类:

- (1) 原子构件: 不需再分的构件, 是基本的复用单元.
- (2) 复合构件: 由一组紧密协作的成员构件及其关系组成. 它是大粒度的复用单元, 其成员构件既可以是原子构件, 又可以是复合构件. 复合构件显式定义了构件的组成结构, 使之更具构建性.

在 JBCOM 中, 构件的接口不仅描述构件向外界提供的功能, 而且还描述构件需要外界提供的功能. 同时, 为了减少接口的复杂性, 在构件接口描述中还定义了“服务”(service). “服务”是一组紧密耦合的对外提供和对外请求的功能的集合. “服务”加大了接口的描述粒度, 提升了接口复用的层次.

依据以上对构件模型的讨论, 相应的青鸟构件描述语言 JBCDL(Component Description Language) 将构件的描述分为以下四个部分:

- ⊗ 功能列表部分
- ⊗ 服务描述部分
- ⊗ 构件接口规约部分
- ⊗ 构件体规约部分

其中, 构件体规约部分和构件接口部分是一个构件描述的两个组成部分, 将它们单独列出有利于反映一个构件接口规约可以对应多个构件体实现部分的情况. 而功能列表部分、服务描述部分从属于构件接口部分中, 将它们单独列出有利于复用.

3 支持变化性的构件模型 JBCOM/E

3.1 领域工程及其领域特性

领域工程是针对一个应用领域, 对领域中的若干系统进行分析, 识别这些系统共享的领域需求, 设计能够满足这些需求的构架, 并在此基础上开发和组织该领域的可复用构件的过程. 领域工程将主要的领域特性分为以下三种^[8]:

- ◇ 必须的(Mandatory): 即领域中所有系统都必须具有的本质的且共同的需求, 它体现了该领域中系统的共同性.
- ◇ 可选的(Optional): 部分现有系统具有这类需求, 但并非全部系统都具有. 它体现了领域中系统间的变化性.
- ◇ 多选一的(Alternative): 即领域中的系统在多个需求中, 必须选且只能选其中一项的需求. 这是一组互相之间存在着互斥关系的需求. 假设需求 p, q, r 是这样的一组需求, 当单独地考察每项需求时, 它们都是可选的需求, 但是, 一个特定的系统必须具有其中的一项需求, 又只能具有其中的一项, 即, $\langle p, q, r \rangle$ 形成互斥集. 这类需求体现了系统间的变化性.

另外, 领域中具有变化性的需求间还包括依赖、互斥关系.

3.2 构件模型 JBCOM/E

对构件模型进行扩充的方法是将领域工程关于领域特性的研究成果应用到构件模型中, 在构件模型中加入描述领域变化性的描述成分, 提出抽象构件的概念, 使之可以反映领域

的特性, 表达一族具有共性和变化性的构件. 抽象构件的引入, 使构件模型更具灵活性和适应性, 为基于构件的开发提供了一个适应领域变化性的一致性的技术基础.

3.2.1 抽象构件定义

抽象构件(abstract component)是适应领域需求, 对同领域一族具有共性和变化性的构件进行的抽象. 在抽象构件的接口中具有描述领域变化性的成分.

具体构件(concrete component)是相对于抽象构件而言的, 具体构件描述应用系统固定的构成成分, 其接口中不具有描述变化性的成分.

抽象构件可以看成是对领域中若干应用系统的一族具体构件的抽象, 抽象其共性和变化性, 并加以表达. 在规约级, 这样的—个抽象构件可以表达并对应一族具体构件. 当抽象构件的接口中的变化性成分被选定和固定下来后, 抽象构件便成为了具体构件, 我们称这个过程为具体化过程. 如下图 1 所示, 当抽象构件 C_2 从对外所需求的多个功能 f_1, f_2, f_3 中选择一项功能, 并确定是否对外提供其可选功能 f_4 后, 抽象构件 C_2 就被具体化为具体构件 C_2', C_2'' , 包含 C_2 的抽象构件 C 也随之具体化为 C' 和 C'' .

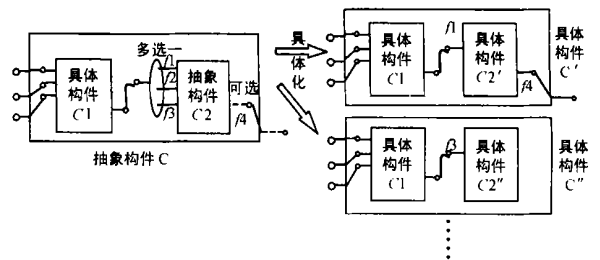


图 1 抽象构件的具体化

3.2.2 抽象构件如何来描述变化性

抽象构件适应领域工程的需求, 对构件模型做了有效地完善和扩充, 不仅体现可选和单选—这样的变化性, 而且也描述变化性间的依赖关系, 为更灵活地构造系统打下了基础.

抽象构件的变化性体现在接口上. 对于可选和单选—的变化性, 可以直接在“功能”或“服务”的前标识“可选”或“单选—”. 而在抽象构件的接口中描述变化性的依赖关系时, 需要注意依赖的顺序问题, 明确是谁依赖于谁. 依赖关系用一个二元组 \langle 被依赖方, 依赖方 \rangle 来表达. 当变化性的被依赖方未被选定时, 其依赖方也就不能被选定. 如果分别用 o 来表示可选, 用 a 来表示单选—, 则 f_o 和 s_o, f_a 和 s_a 分别表示可选的功能和服务、单选—的功能和服务. 下面关于依赖关系的表达都是正确的:

$$\langle f_{1o}, f_{2o} \rangle; \langle s_{1o}, s_{2o} \rangle; \langle f_{1o}, f_{2a} \rangle; \langle f_{1a}, f_{2o} \rangle; \langle s_{1o}, s_{2a} \rangle; \langle s_{1a}, s_{2o} \rangle; \langle f_{1o}, s_{1o} \rangle; \langle s_{1a}, f_{1o} \rangle$$

其中, $\langle f_{1o}, s_{1o} \rangle$ 表示只有在选定一个可选的功能 f_1 的情况下, 才能存在相应的可选服务 s_1 .

当抽象构件是复合构件时, 复合构件除了接口信息之外, 还包括内部成员构件的组成、成员构件的实例化、成员构件之间的连接以及外部接口到内部成员构件接口的映射等四部分内部结构信息. 复合构件并不显式地描述其内部结构信息的

变化性, 因为这些可以通过其内部成员构件的变化性描述得到体现. 当其内部成员构件的变化性及其依赖关系固定下来后, 复合构件的内部结构的变化性也就固定下来.

本文定义的支持领域变化性的构件模型 JBCOM/ E 如图 2 所示.

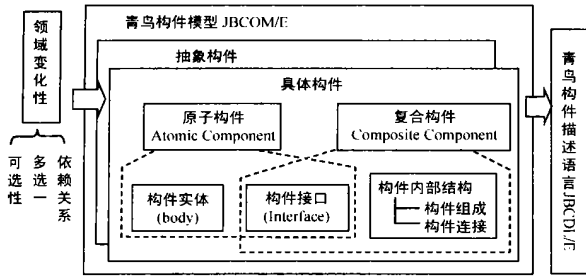


图 2 构件模型 JBCOM/ E

3.2.3 抽象构件规约

抽象构件规约是抽象构件的语言描述 (Abstract Component Description Language, ACDL), 它在原有构件接口规约基础上做了以下三部分的扩充: (用灰底标出)

(1) 定义构件的类型 (component_ type), 用来说明此类构件是抽象构件, 还是具体构件.

Component_ interface ::= Component <component name> is

Component_ Type: [Abstract] Primitivel compound

[Provides: // 提供的功能

... ..

(2) 在构件的对外接口 (对外需求或提供的服务、功能) 中添加了表示变化性的任选项 (optional) 和多选一 (alternative) 的标志.

<function_ spec_ list> ::= <function_ spec> {, <function_ spec>}

<function_ spec> ::= [optional | Alternative] [<return_ type>] <func_ name>

({ [in | out | inout] <type> <parameter_ name> , });

... ..

[services:

[dual] [optional | Alternative] [Service <service_ name>

{, [dual] [optional | Alternative] [Service <service_ name> }

name } }

... ..

(3) 在构件规约描述中加入关于变化性间的依赖关系描述.

... ..

Constraints ::=

Dependence <[Optional | Alternative] [Function_ Spec | Service], [Optional | Alternative] [Function_ Spec | Service]>

Description: // 变化性间的依赖性成分的描述

<text_ docu>

End Constraints;

3.2.4 相应构件管理和构件组装技术

® 构件管理

抽象构件的提出为构件管理提供了良好的支持. 利用抽象构件可将同一领域中一族具有相似功能的构件分层管理. 用户在查询构件时, 可从抽象程度最高的抽象构件开始查找, 并根据需求逐步求精确定位在所需的构件上.

构件库在管理构件时, 可以利用抽象构件的概念对构件进行分层管理, 标定不同层次的抽象构件与其具体构件之间精化映射关系. 这样将一族具有共性的构件聚集在一起, 有利于构件管理的高效性.

® 构件组装

抽象构件可以不具备特定的实现体部分. 抽象构件需要被具体化为特定的具体构件后才能应用到系统组装中去, 其具体化反映了逐步求精的应用系统开发过程. 在构件组装中, 先从抽象构件规约开始, 根据需求, 结合抽象构件具体化指南, 将抽象构件中变化性部分固定下来, 形成具体构件规约, 再由具体构件的实体完成构件组装.

4 应用实例 POS 领域平台

下面以青鸟 POS 领域平台为例, 说明如何用抽象构件的概念及其规约来定义、描述一族构件, 并且如何对此领域中的构件进行管理和组装.

POS 系统是超市、百货商场使用的实时销售管理系统, 被广泛应用于当前诸多的商业领域中, 具有一定的复杂性. 青鸟 POS 领域平台是由 POS 领域构件、青鸟构件库管理系统以及青鸟组装工具搭建而成的, 用户可以利用 POS 领域平台高效、高地开发特定于用户需求的 POS 系统.

POS 领域平台中共提取了 25 个构件, 分为基础服务构件、销售方式构件、POS 信息设置构件、优惠方式构件、商品查询方式构件和用户管理方式构件六大类, 如图 3 所示, 其中可选构件用虚线圈起, 多选一构件用轴虚线圈起, 构件之间的关系用实箭头表示.

下面以抽象构件 SaleMode 为例介绍构件管理和构件组装过程. 抽象构件 SaleMode 代表一族具有共性、又具变化性的销售方式构件, 普通销售、残损销售、退货销售为可选的, 而贴花

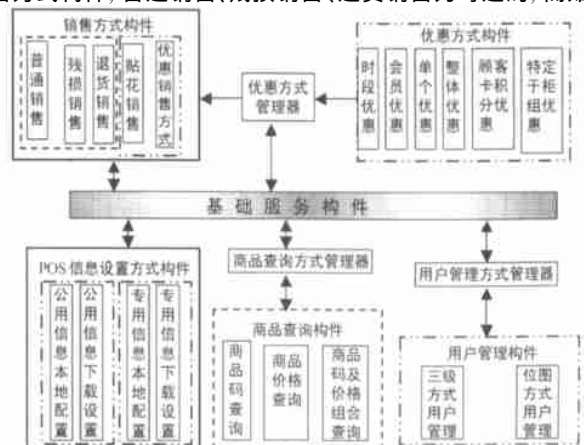


图 3 在 POS 领域开发平台中提取的构件

销售和前台单个优惠销售为多选一,所有其他销售均要依赖于普通销售。

可以利用构件库来管理这族销售构件,首先建立销售构件术语,将抽象构件 SaleMode 的描述、规约入库,接着在抽象构件 SaleMode 层之下,将具体的不同方式的销售构件的规约、描述和实体一一入库,并建立检索关系。

POS 系统进行组装时,首先根据销售功能需求将抽象构件 SaleMode 引入,当功能框架搭好后,就可以根据用户的具体需求,在构件库中精化定位到具体销售方式构件,利用组装工具建立构件之间的连接关系,从而生成特定于用户需求的 POS 系统。

5 结论

在基于构件的软件开发中,构件模型是基础,应用系统的集成式组装是关键技术。领域工程、构件管理与应用工程之间有着密切的联系。本文定义了一种支持变化性的构件模型 JB-COM/E,引入了基于领域特性的抽象构件概念,研究了抽象构件的变化性及其间的依赖关系的规约描述,以及抽象构件的具体化过程,将领域工程、构件管理和应用工程三种软件开发活动有机地结合在一起,并开发了相应的构件组装工具予以支持。

这样,不仅增强了构件模型的描述能力,而且也将相应的构件管理和应用组装提升到一个更高的抽象、复用层次,为基于构件的软件开发提供了一个一致的支持领域特性的技术基础。

参考文献:

- [1] 杨芙清,梅宏,李克勤.软件复用与软件构件技术 [J].电子学报,1999,27(2):68-75.
- [2] Gregory Zelesnik. The Unicorn Language Reference Manual [M]. UK: CMU, Pennsylvania 15213-3890, May 13, 1996.
- [3] Robert J Allen. A Formal Approach to Software Architecture [D]. UK: Carnegie Mellon Univ, CMU Technical Report CMU-CS 97-144, May 1997.
- [4] Nenad Medvidovic. Formal definition of the Chiron 2 software architectural style [R]. Technical Report UCFICS 95-24, Department of Information and Computer Science, USA: University of California, Irvine, August 1995.
- [5] Richard Taylor, Nenad Medvidovic, Kenneth Anderson, E. James Whitehead Jr, Jason Robbins, Kari Nies, Peyman Oreizy, Deborah Dubrow. A component and message based architectural style for GUI

software [J]. IEEE Transactions on Software Engineering, June 1996.

- [6] Johannes Sametinger. Software Engineering with Reusable Components [M]. Germany: Springer Verlag Berlin Heidelberg, 1997.
- [7] 骆华俊,唐稚松,郑建丹.可视化体系结构描述语言 XYZ/ADL [J].软件学报,2000,11(8):1024-1029.
- [8] 李克勤.面向对象的领域工程方法研究 [D].北京:北京大学,2000.
- [9] Hafedh Mili, Fatma Mili, Ali Mili. Reusing software: Issues and research directions [J]. IEEE Transactions on Software Engineering, 1995, 21(6): 528-562.
- [10] OMG. Common Object Request Broker: Architecture and Specification V2.3 [M]. Oct. 1999.
- [11] Guilemo Arango, Ruben Prieto Diaz. Domain analysis concepts and research directions [A]. Domain Analysis and Software System Modeling [C]. Los Alamitos, California: IEEE Computer Society Press, 1991.
- [12] Will Tracz, Lou Coglianesi. Domain Specific Software Architecture Engineering Process Guidelines (Version 2.1) [R]. USA: ADAGE IBM-92-02B, April 1994.

作者简介:



张文娟 女,1969年12月生于河北保定市,博士,高级工程师,2001年获北京大学理学博士学位,主要研究领域为软件工程、面向对象技术、软件构件技术。



赵俊峰 女,1974年8月生于陕西省西安市,1996年获西安交通大学工学硕士学位,现工作于北京大学计算机系并同时攻读计算机软件在职博士研究生,讲师。主要研究领域为软件复用技术,构件技术,大型分布式构件库系统等。

谢冰 男,1970年1月出于湖南湘潭,博士,主要研究领域为软件工程、形式化方法、分布式系统。

杨芙清 女,教授,博士生导师,中科院院士,主要研究领域为软件、软件工程、软件工业化生产技术与系统等。