

文章编号: 1000-5013(2012)03-0269-06

面向复用的软构件信息系统的设计与实现

陈杜英, 刘韶涛

(华侨大学 计算机科学与技术学院, 福建 厦门 361021)

摘要: 基于软件体系结构,从需求分析出发,讨论适合此系统的体系结构设计决策,分析系统设计的实现过程,设计并实现一个面向复用的软构件信息系统. 该系统能使管理人员有效地组织和管理各类软构件实体和相关角色信息等,开发者可以在此高效检索到需要的软构件,构件生产厂商可以方便提交自己生产的构件,有效管理软构件以提高软件开发效率.

关键词: 软构件; 信息系统; 软件体系结构; Wright 语言; 复用技术

中图分类号: TP 311.52

文献标志码: A

软件复用技术是解决软件危机的有效途径,而软构件技术是实现成功软件复用的关键因素之一^[1]. 为了提高软件复用的效率,需要有效的组织和管理软构件. 软构件信息系统能有效地组织和管理大量的可复用软构件,并提供相应的工具,支持在软件开发过程中方便的查询、理解、选取和使用软构件,使基于软构件复用的软件开发成为现实. 软件体系结构描述语言 Wright 语言是美国卡耐基梅隆大学的 R. Allen 和 D. Garlan 提出的一种高级分布式学习(ADL),它为体系结构中的连接提供了形式化基础^[2]. Wright 语言的主要特点是,对体系结构和抽象行为的精确描述,定义体系结构风格的能力,以及一组对体系结构描述进行一致性和完整性的检查. 体系结构描述通过构件、连接件,以及他们之间的组合来描述,抽象行为通过构件的行为和连接件的胶水来描述. 本文结合软件体系结构策略,设计并实现了一种具有复用价值的软构件信息系统.

1 软件体系结构级别的设计策略^[3]

基于体系结构的软件分析设计过程(SADPBA)是一个迭代的过程,项目的每个阶段都是以上一个阶段的工作为基础. SADPBA 将过程分解为 3 个动作:需求分析、软件体系结构与系统设计. 在 SADPBA 中,设计空间被扩展并应用到设计过程的 3 个不同的问题,即关注系统功能的功能设计空间、构件组织的体系结构设计空间,以及设计细节与算法的系统设计空间. 设计空间的形式化定义为

$$DS = \{d_1, d_2, \dots, d_n\}.$$

其中: d_i 是设计空间的维度,是对系统某一特性或设计决策的描述, d_i 中枚举出的可能方法称为范畴.

在面向领域的设计过程中,使用 3 个空间意味着设计过程将从一个空间映射到另一个空间,从需求分析到体系结构设计再到系统设计. 该过程执行从需求到设计细节的提炼. 功能设计空间关注需求,尤其是那些功能属性. 体系结构设计空间可以用体系结构描述方法进行维度的度量,如构件、连接件、配置、体系结构风格与模式等. 系统设计关注更多的细节,包括构件与连接器的内部结构及关键算法.

2 需求分析

将系统中的参与者分为用户和管理员两类,其用例图如图 1 所示. 用户包括客户、软构件供应商和客户端用户. 客户需要实现的功能,有输入信息登陆系统、输入检索内容返回相应的检索结果、对检索结

收稿日期: 2011-11-12

通信作者: 刘韶涛(1969-),男,副教授,主要从事软件体系结构与软件复用的研究. E-mail:shaotaol@hqu.edu.cn.

基金项目: 国务院侨办科研基金资助项目(09QZR02)

果进行处理, 当未检索到所需要构件时, 可以发布生成新软构件请求. 软构件供应端用户提供的功能, 主要有登录系统、提交构件、查看构件发布信息. 管理员的任务包括软构件管理、软构件入库管理、软构件信息管理、角色管理、软构件信息发布平台管理、软构件评价管理.

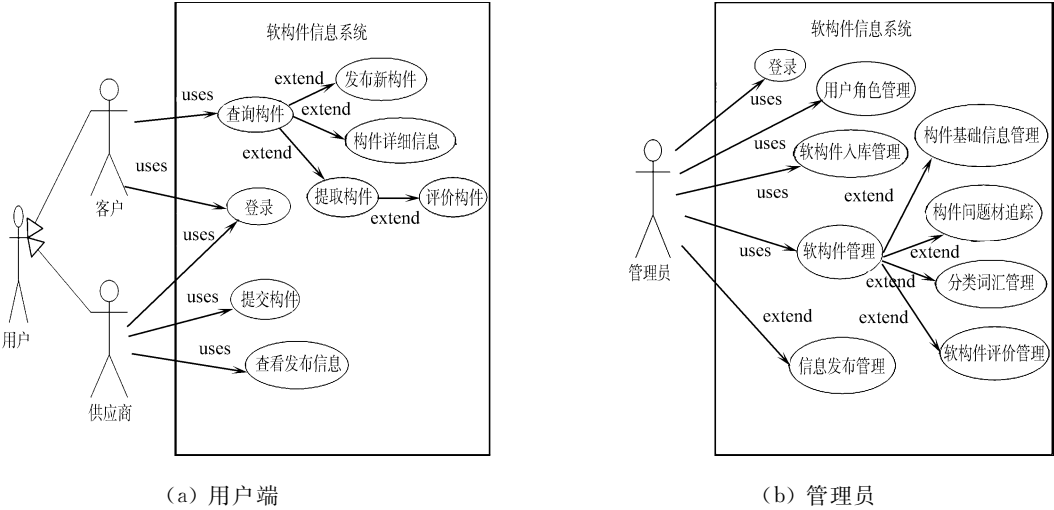


图 1 系统参与者用例图

Fig. 1 Use case diagram of participants in the system

3 体系结构设计

3.1 总体结构

该软件项目采用的是典型的 3 层 B/S 模型的体系结构, 如图 2 所示. 该模型在逻辑上将应用功能分为 3 层: 表示层、功能层、数据层. 1) 表示层. 对应上面的用户端的显示, 主要通过浏览器实现用户端的显示, 完成用户的请求操作. 2) 功能层. 对应代理层, 完成客户请求的功能实现. 3) 数据层. 主要是软构件数据库管理系统, 由管理员进行维护. 在体系结构阶段, 将整个系统分为 3 个子系统: 用户端子系统、代理层子系统和数据层子系统.

3.2 用户端

用户端表示层子系统, 关注表现层功能, 供用户使用. 用户端的体系结构如图 3 所示, 包含 7 个主要构件:

- 1) 界面, 负责用户端界面的构建与显示;
- 2) 查询, 负责提取客户输入的查询构件的信息;
- 3) 提取, 负责获取客户输入的需要提取的构件信息;
- 4) 发布, 负责为客户填写需要生成的新构件的内容, 以及为供应商提供当前存在的需要的新构件发布信息;
- 5) 评估, 负责处理客户的评价构件信息的输入;
- 6) 提交, 负责处理供应商的提交构件处理;
- 7) 构件存储中心, 负责用户端与外面的数据通信. 在这里, 后台构件存储中心采取两个独立的线程进行处理, 以存放接收的消息和存放将要发送的消息.

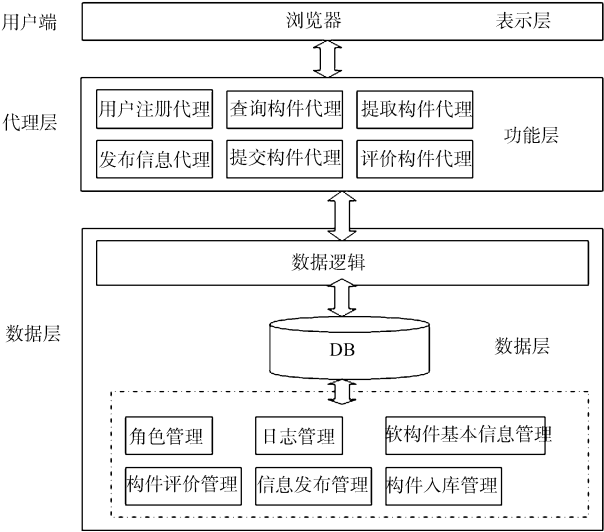


图 2 软构件信息系统的体系结构图

Fig. 2 Architecture diagram of software component information system

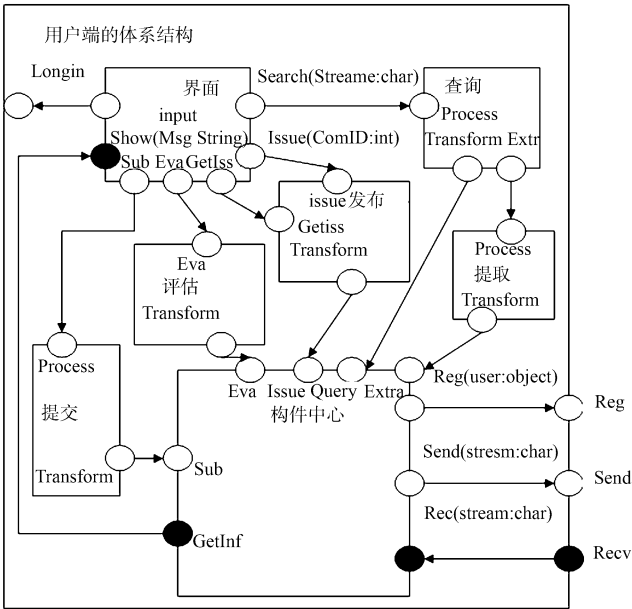


图 3 用户端的体系结构

Fig. 3 Architecture of client

用户端体系结构支持 6 个过程:生成灵活的用户接口、查询构件、提取构件、评价构件、处理发布构件信息和提交构件. 用户端接口依如下步骤生成:用户输入登录信息,通过验证,登录系统,依据用户角色的不同,分为客户和供应商两种角色,分别返回相应角色登录成功后的界面.

3.3 代理层子系统

主要负责处理用户的请求,向数据层子系统传送请求,并将从数据层子系统返回的结果传送给用户端. 代理层构件负责用户对象的建立、处理用户消息并给出反馈. 它有 3 个端口:Reg,Send 和 Recv. Reg 负责用户注册与注销,Send 与 Recv 是负责消息交互的端口. 代理层有 5 个顶层构件:处理中心、查询与提取代理、提交代理、评估代理与发布代理. 其中:处理中心是由构件存储中心构件、代理管理系统与目录功能组成的. 代理层的体系结构图,如图 4 所示.

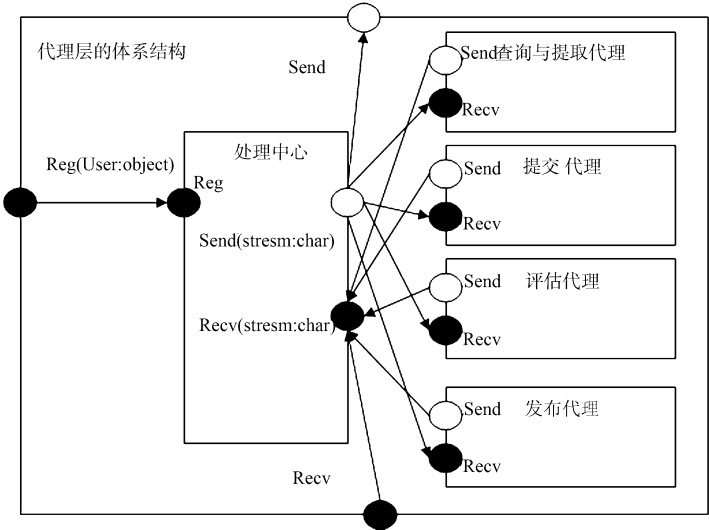


图 4 代理层的体系结构图

Fig. 4 Architecture of the agent layer

3.4 数据层子系统

主要负责处理 SCISPAP 端的消息请求,软构件相关信息的管理,维护系统的运行. 数据层负责软构件信息系统数据库的维护,由管理员进行操作,包括软构件用户的角色管理、软构件的入库管理、软构件的基本信息维护管理、软构件发布信息管理、软构件评价管理. 这些管理都将使得软构件数据库中心

受到相应的影响,从而改变数据库中心数据.
图 5 为数据层的体系结构.

4 系统设计

对于管理软构件的信息系统,必须具有存储数据量大,满足使用方便、操作灵活和安全性好等特点. 系统在设计时应该满足以下 6 个目标: 1) 搭建网络软构件信息系统,系统可以进行大量构件的存储和管理操作,并保证安全性; 2) 提供软构件信息的高效提交、检索和提取等操作; 3) 提供为构件评价管理功能; 4) 提供构件信息发布有效管理; 5) 可以进行日志管理操作,方便事后查看; 6) 系统运行稳定高效安全.

将体系结构转化为实现相关模型,使用 Wright 语言来描述构件间消息传递关系及其空间分布,揭示构件之间的交互过程^[4]. 以客户查询软构件和供应商提交构件为例,用户与系统的交互流程:

- 1) 客户向系统发送查询构件的请求→系统向构件存储中心发起查询的请求→构件存储中心返回查询结果给系统→系统返回查询结果或查询失败信息;
- 2) 供应商向系统发送提交构件的请求→系统向构件管理者发送提交构件的请求→构件处于审核状态→构件管理者审核提交构件→审核成功后则将构件录入构件存储中心→构件存储中心返回结果给系统→系统将结果返回给管理者→管理者将结果返回给系统→系统提交成功或审核失败,管理者通知系统提交失败,系统将处理返回提交成功或失败结果.

使用 Wright 语言,描述以上 2 个交互过程^[5].

Configuration SubmitQueryFlow

// 查询与提取构件流

InterfaceType SendT=Send! x→ SendT II §

// 发送接口

InterfaceType ReceiveT=Receive? x → ReceiveT □ §

// 接收接口

Connector Interact

// 定义交互构件

Role From=ReceiveT

Role To=SendT

Glue=From. receive? x →T0. send! x→ Glue □ §

Component Client

// 客户构件

Port Ssystem=SendT

Port Rsystem=ReceiveT

/* 客户发起触发发送消息,系统监听是否接受客户此类消息 */

Computation=SClient. Send! x→ RSystem. Receive ? x → Computation II §

Component Provider

// 供应商构件

Port Ssystem=SendT

Port Rsystem=ReceiveT

/* 供应商触发发送消息,系统监听此类消息 */

Computation=SProvider. Send! x→ RSystem. Receive ? x → Computation II §

Component System

// 系统构件

Port SClient=SendT

Port RClient=ReceiveT

Port SProvider=SendT

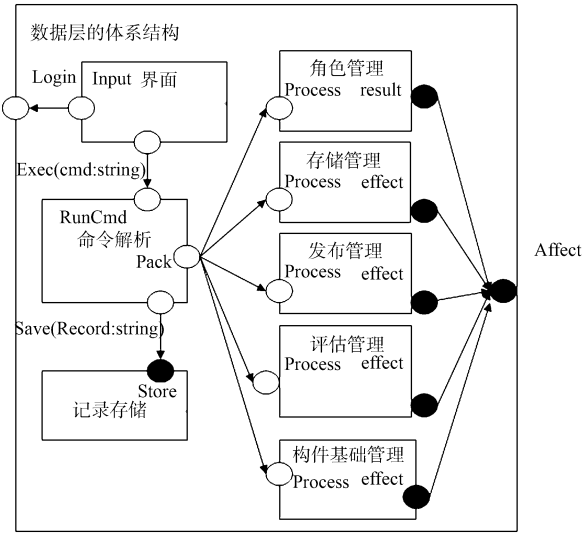


图 5 数据层的体系结构图
Fig. 5 Architecture of the data layer

Port Rprovider = ReceiveT
Port SCmpStCen = SentT
Port RCmpStCen = ReceiveT
Port SManager = SendT
Port RManager = ReceiveT

// 客户发送查询消息，系统接收此消息，构件存储中心接收系统查询消息，构件存储中心再返回结果

// 供应商发送提交消息，系统接收此消息，构件管理中心接收系统消息，构件管理中心再返回处理结果

Computation = RClient. Receive? x \rightarrow $\overline{\text{SSystem. Send! x}}$ II
RcomStCen. Receive? x \rightarrow $\overline{\text{SCmpStCen. Send! x}}$ \rightarrow Computation II §
RProvider. Receive? x \rightarrow $\overline{\text{SSystem. Send! x}}$ II
RManager. Receive? x \rightarrow $\overline{\text{SManager. Send! x}}$ \rightarrow Computation II §

Component Manager // 定义管理处构件

Port SSystem = SendT
Port Rsystem = ReceiveT
Port SCmpStCen = SendT
Port RcmpStCen = ReceiveT

// 系统接收提交构件消息，触发审核处理，审核处理完，向构件存储中心发起处理结构，构件存储中心将返回结果，系统将处理完发送审核处理结果。

Computation = RSystem. Receive? x \rightarrow Verfing \rightarrow Computation II §
Verfing = RcmpStCen. Receive? x \rightarrow $\overline{\text{SSystem. Send! x}}$
 \rightarrow Computation \rightarrow Computation II §

Component CmpStCen // 定义构件存储中心构件

Port SSystem = SendT
Port RSystem = ReceiveT
Port SManager = SendT
Port RManager = ReceiveT

Computation = RSystem. Receive? x \rightarrow Computation II §
RManager. Receive? x \rightarrow Computation II §

Insatances // 实例

C : Client
P : Provider
S : System
M : Manager
CM : CmpStCen
I_{1...5} : Interact

Attachments // 配置行为描述

S. SClient = I₁. Receive
C. RSystem = I₁. Send
S. SProvider = I₂. Receive
P. RSystem = I₂. Send
CM. SSystem = I₃. Receive
S. RcmpStCen = I₃. Send
M. SSystem = I₄. Receive

```
S. RManager    =  I4. Send
CM. SManager   =  I5. Receive
M. CmpStCen    =  I5. Send
End SubmitQueryFlow
```

5 结 束 语

基于构件的软件开发是一种高效的软件重用技术,通过结合软件体系结构技术,研究了软件体系结构的整体设计方案. 该设计行为被抽象成设计空间,把它分为功能设计空间、体系结构设计空间与系统设计空间. 这是 SADPBA 的核心概念,即以体系结构为中心的设计过程. 然而,原型软构件信息系统仍存在一些问题,下一步工作主要是进一步优化系统性能,努力提高系统的查全率、查准率,以及提高系统响应时间,同时考虑人工智能方法改进软构件的检索效率,进一步提高系统检索的能力.

参考文献:

[1] 彭博,王晋,李亚芬. 面向 Web 领域的构件库系统的设计与实现[J]. 计算机应用技术,2009(20):1-3.
[2] 冯冲,江贺,冯静芳. 软件体系结构理论与实践[M]. 北京:人民邮电出版社,2004:32-36.
[3] 覃征,邢剑宽,董金春,等. 软件体系结构[M]. 北京:清华大学出版社,2008:139-140,148-153.
[4] 张友生,李雄. 软件体系结构原理、方法与实践[M]. 北京:清华大学出版社,2009:112.
[5] 王烨. UML 与 WRIGHTADL 在企业设备管理系统中得应用[D]. 苏州:苏州大学,2006:41-45

Design and Implementation of Software Component
Information System Oriented Reuse

CHEN Du-ying, LIU Shao-tao

(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

Abstract: A software architecture suitable for the information system is discussed, the implementing procedures of the system design are analyzed based on software architecture and requirement analysis of the system and a software components information system oriented reuse designed and implemented. The system can facilitate the managers to effectively organize and manage all kinds of software components, its related roles' information, and can help the developers effectively retrieve the components they need, can facilitate the providers of components to submit the components they developed and can effectively manage software components to improve the efficiency of software development.

Keywords: software component; information system; software architecture; Wright language; reuse technology

(责任编辑: 陈志贤 英文审校: 吴逢铁)