

青鸟系统中可复用软件构件的表示与查询

常继传,李克勤,郭立峰,梅宏,杨芙清

(北京大学计算机科学技术系,北京 100871)

摘要: 可复用构件的表示和检索是构件库和软件复用研究的重点,它伴随复用实践的深入和新技术的出现而不断取得进展.青鸟构件库系统以青鸟构件模型为基础,采用以刻面分类为主、多种分类模式相结合的方法对构件分类索引,支持多种构件查询方式的组合使用.本文介绍了青鸟型构件库系统中可复用软件构件的分类表示模式,以及构件检索机制的体系结构、设计依据和系统特性.

关键词: 软件复用;软件构件;构件库;构件表示;构件检索

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2000) 08-0020-04

Representing and Retrieving Reusable Software Components in JB (Jadebird) System

CHANG Ji-chuan, LI Ke-qin, GUO Li-feng, MEI Hong, YANG Fu-qing

(Department of Computer Science and Technology, Peking University, Beijing 100871, China)

Abstract: How to represent and retrieve reusable software component is always of interest to the reuse community. As a software development environment supporting systematic reuse, JB system chose faceted classification as the primary classification method, with several other methods constituting a combined solution, which supports multiple retrieval methods and their combination. This paper discusses the representation, classification and retrieval of reusable components in JB system, and introduces our assumptions, design rationale, system architecture, data model and features of related subsystems.

Key words: software reuse; software component; reuse library; component representation; component retrieval

1 引言

对软件复用和软件构件的研究始于 60 年代后期,30 年来人们一直关心的两个问题是:(1)如何有效地表示可复用软件构件?(2)如何在已有的大量构件中找到所需的构件?

与前者相关的研究涉及到构件模型(什么是构件?构件的基本属性是什么?构件之间如何交互?)、构件规约(如何用形式化方法规约构件的功能和其他属性?)、构件库数据建模(构件库的元模型和数据模型是什么?构件库采用何种分类索引模式?)等问题;而后者涉及到构件库查询接口(SQL、OQL 或者与复用相关的构件库查询服务)、构件的匹配和检索算法、以及自动工具支持(如构件查询工具、认证工具和评价工具)等问题.构件库互操作也对构件表示和检索提出了新的要求,必须考虑如何在不同构件库之间共享和交换构件信息(这涉及到库访问接口、数据模型、传输格式和传输协议的标准化),以及如何在分布的多库系统(如 Internet 上的多个构件库)中表示和检索构件(这涉及到构件的命名、版本、配置和分布检索算法等问题).

JBCL 构件库系统^[1]是青鸟型系统的重要组成部分,它

能够保存和管理多形态、多层次的 reusable 软件构件,联结面向复用的开发与基于复用的开发.本文将主要讨论青鸟系统中的构件的描述和分类模式及其对构件检索的支持,其他方面的问题请参看相关技术报告,在此并不详述.

2 模型和系统概述

2.1 数据建模

一个完整的构件可能包括构件实体表示、构件目录信息和元数据(Meta-Data)三个部分,分别对应于构件库数据建模中的数据、数据模型和元模型三个层次^[2].

构件实体表示可以有结构或无结构的,对有结构的构件实体可以进行基于内容的检索(如正文检索和规约匹配)和版本控制.构件目录信息包括描述性信息和组织性信息;描述性信息包括构件的尺寸、版本、关键词和发布时间等属性;组织性信息包括构件类之间、构件与构件类之间以及构件之间的关系.在描述性信息中,构件 ID 用于在库中唯一标识构件,UID 在分布的多库环境中全局唯一标识构件,它们可以作为构件实体的访问句柄.元数据主要包括数据模型信息,用于构

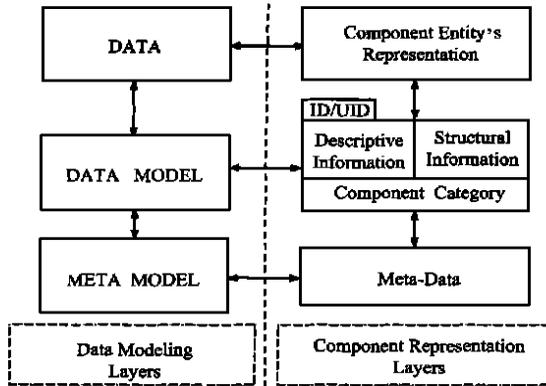


图1 数据建模与构件表示

件信息的交换和传输以及构件库数据模型的定义和扩展。

下图为青鸟构件库的数据模型：

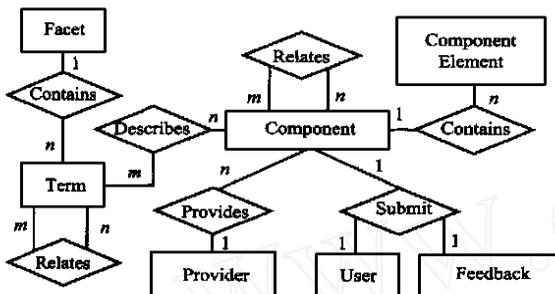


图2 青鸟构件库实体关系模型

2.2 构件库的角色

由于各个组织的规模、目标领域、组织模式和复用成熟度不同,导致构件库系统在组织中的角色和作用也不相同。可以归纳出以下 4 种典型的构件库:(1)面向项目的构件库、(2)面向领域的构件库、(3)面向构件共享的构件库和(4)面向构件市场的构件库。从(1)到(4)标志着构件库规模越来越大、目标领域越来越清晰和广泛、复用成熟度也越来越高。

上述因素将直接影响到构件的表示和分类,以及相应的检索和交换方式。例如:面向项目的构件库需要突出构件的项目背景;面向领域的构件库需要以特定领域的系统构架为骨架来组织构件,并表示出领域之间的相关关系;面向共享的构件库建立在若干相对独立的子库之上,需要为分布在各地的用户群提供灵活统一的查询界面,实现对不同应用领域、形态、位置和表示方法的构件的透明访问;面向构件市场的构件库在更大范围内提供复用支持,通过商业手段可以使构件的认证、度量与质量保证等技术难题得到较为满意的解决^[3]。设计构件库时,应该首先根据复用组织的现状和复用目标确定构件库的角色,在此基础上选择适当的分类和检索方案。

2.3 基本假定

青鸟构件库系统的需求分析和设计实现基于以下基本假定:

⑧ 构件:构件是软件开发过程中不同阶段(如分析、设计、编码、测试等)生成的不同形态(如类、框架、构架、模式等)、不同表示(如图形、伪码、语言等)的软件实体。

⑧ 构件库角色:青鸟构件库目前定位为面向共享和面向市场的构件库,目标是建立灵活的系统构架和数据模型,通过定制和裁剪满足不同构件库角色的需求。

⑧ 构件库用户:构件库用户特指从构件库获取复用服务的用户。构件库用户所关心的应用领域和构件种类不同,复用经验不同,所处地域不同,使用不同的工具访问构件库。用户的数量很大,访问构件库频繁。

⑧ 复用成本:对复用者而言,单个构件的复用成本由检索成本、获取成本、理解成本、修改成本和组装成本构成。对构件库而言,单个构件的复用成本可能包括获取成本、评价与认证成本、分类与存储成本。由于对构件库访问次数非常多,使得建立和维护构件分类索引的成本远小于由此节省的用户检索和理解成本。

⑧ 构件库数据模型:不同构件库的数据模型不同,同一构件库的数据模型也可以变化。需要建立灵活的系统构架和统一的元模型,来满足构件库间的互操作需求和数据模型的演化需求、屏蔽数据模型变化对复用库访问接口的影响。

⑧ 构件类型和数量:库中构件的生命周期、形态和构件实体的表示在很大范围内变动;构件数量很大,而且不断更新和演化,需要使用(半)自动工具辅助人工进行构件的分类和检索。构件可能保存在分布的异质的子构件库中。

2.4 分类模式

对大量构件进行分类有三个好处:便于组织管理、方便查询和辅助理解。构件分类是建立(包括生成)和维护构件目录信息的活动,构件的检索方法依赖于构件的表示和分类。文献[4]对各种构件分类与检索方法进行了综述,通过比较,青鸟系统选择以刻面分类为主,关键词、属性-值、关系索引等多种模式相结合的方法对构件进行分类。刻面分类方法从若干不同的维度描述复杂对象,在刻面中的术语之间可以定义关系(如一般特殊关系和同义词关系),具有枚举分类、属性-值分类和关键词分类几种方法的优点。在同一系统中支持多种分类模式和检索方法的组合使用,可以满足不同复用组织和不同用户群体的需求。初级用户可以选择使用自己熟悉的方法,高级用户可以综合使用几种方法。实践证明各种方法互相补充,能够提高检索效率^[5]。

2.5 刻面分类

每个术语 T 对应于一个谓词 P_T , 根据术语 T 检索将得到所有满足 P_T 的构件集合 C_T 。称术语 T 是术语 T^* 的上层术语当且仅当对任意构件 C 有 $P_{T^*}(C) \rightarrow P_T(C)$ 。构件 C 被索引为术语 T 当且仅当对于所有以 T 为上层术语的术语 t , 都有 $C \in (C_T - C_t)$ 。可见建立索引的算法是一个对术语空间的深度优选搜索。同一刻画中的术语被组织成层次结构,上层与下层术语之间是严格的一般特殊关系。在刻面 F 中选择多个术语 T_A, \dots, T_K 时,检索得到的构件集合为: $C_F = C_{T_A} \dots C_{T_K}$ 。查询条件中有多个刻面时,各刻面检索到的构件集合之交构成最终结果集合。在青鸟系统中根据术语之间的一般特殊关系对刻面中的术语进行编码,提高了检索算法的效率。

一般特殊关系的引入能够描述不同构件在通用性与专用性方面的差异,并隐含地表达构件之间的相似关系。复用者可

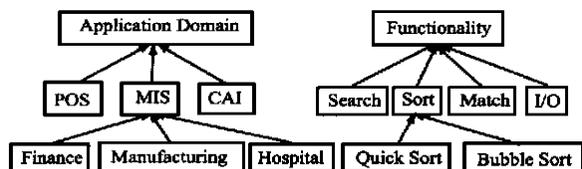


图3 刻面和术语间的一般特殊关系

以在术语空间中周游,对所需构件的抽象层次权衡取舍。在术语之间可以建立同义关系,同义术语之间不可以有一般特殊关系。定义术语的同义词词典可以扩展术语空间的词汇表,以此解决术语的一词多义和多词同义的问题。

青鸟构件库中定义了“使用环境”、“应用领域”、“功能”、“层次”和“表示方法”5个刻面^[1]。构件的计算语义在跨领域复用时非常重要,但复用者经常从应用领域出发构造查询条件,使用“应用领域”和“功能”两个刻面的术语描述构件在应用域和计算域的语义可以兼顾两者的需求。

对刻面分类法的主要批评是需要人工建立和维护术语空间和构件索引。但是术语空间所表达的语义信息,能够准确地描述构件属性,有助于复用者理解构件所处的上下文。基于复用成本的假定,对构件进行刻面分类的工作量将随检索效率的提高而被合理化。当构件数量增加到非常大,对构件进行刻面分类的绝对工作量超出人工所能处理的范围时,可以用自动分类工具辅助构件分类。使用术语空间的管理工具也可以减轻构件库管理员的工作。

2.6 构件间关系

构件之间存在着各种关系,典型的构件间关系包括:继承关系、精化关系、版本关系、包含关系、协作关系等。构件和构件间关系构成了超文本的节点和超链,以某个构件为起点可以沿关系链在构件库中周游。构件间关系的类型和性质由构件库数据模型定义。使用预定义的关系类型可以赋给关系以特定的语义;允许用户自定义关系类型则具有更大的灵活性,适于描述构架、框架、设计模式和软件生命周期各阶段产物之间的复杂关系。青鸟构件库系统预定义了上述6种关系,并提供扩充关系类型的手段和建立构件间关系的工具。

3 系统体系结构

青鸟构件库系统构件表示和检索子系统由多库分布处理机制(包括事务处理服务器、构件命名服务器、位置服务器和特定构件库的包装层)、复用库服务、集成的复用支持工具(数据建模工具、构件分类工具、构件查询工具、构件组装工具、构件评价工具等)组成。多库分布服务机制建立在构件库互操作协议的基础上,向上屏蔽多个分布构件库的位置差异,实现构件共享和分布查询,并处理构件命名、版本、传输和安全问题以及事务处理细节。复用库服务建立在分布机制之上,向上提供标准的复用库访问接口,屏蔽语言和平台差异。集成的复用支持工具通过复用库服务访问复用库,直接向用户提供构件库管理、构件分类、查询、评价和组装的功能。系统整体结构如图4所示,下面将分别介绍各个子系统。

数据建模工具辅助管理员创建、维护和扩展构件库系统

的数据模型(比如实体关系模型中的实体及实体间关联、实体属性及其类型、属性之间的约束等)和建立在数据模型上的分类模式与检索算法。该工具目前仅实现了一个最小功能集,能够建立和维护术语空间。

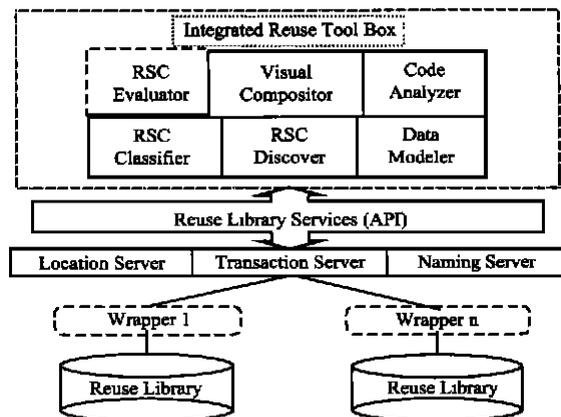


图4 构件表示和检索子系统的体系结构

(图中的虚线表示尚未完全实现)

构件查询工具支持刻面、关键词、属性等多种查询方法的组合,支持以构件间关系为超链的超文本浏览和以术语间层次关系为线索的构件浏览。查询得到的结果是一组符合条件的构件,用户可以查看每个构件的详细信息,并将感兴趣的构件标记为候选,进一步比较、评价或者浏览与之相关的构件。

查询工具中实现了可视化的查询条件构造器,支持对构件集合的并交运算。青鸟构件查询工具对查询条件执行“交”运算实现逐步求精的构件查询,逐渐缩小检索范围。当发现条件过于苛刻时可以回溯到先前的条件。借助构件间关系和术语的层次浏览,用户得到以某个构件为中心的关系闭包,熟悉构件和相应构件库的上下文并比较和理解候选构件的区别。所有这些特性都有助于降低构件的查询和理解的成本。

构件分类工具辅助管理员建立和维护构件目录信息,支持多种模式相结合的分类方法。在对多个分布异质库中的构件建立全局目录时,必须处理数据模型集成、版本控制和命名模式等问题,排除重复的目录项,这些都依赖于底层的分布处理机制,工具的当前版本尚不能处理异质多库的问题。

构件组装工具辅助用户用 CDL 描述构件,可以根据构件的 CDL 描述生成相应的 C++ 代码骨架,在代码级或目标级(在 CORBA 或 DCOM 的支持下)进行组装^[6]。

代码理解和构件评价工具是表示与检索子系统的重要外围工具。构件评价工具对候选构件进行评价,辅助用户选择最符合需求的构件。代码理解工具采用增量分析技术静态分析程序源代码,用 EER (Enhanced Entity-Relationship) 为 C++ 程序建立概念模型并抽取程序信息,按照不同的用户需求组织程序信息,辅助用户理解 C++ 程序^[7]。

复用库服务建立在软件工程环境等底层服务的基础上,以元模型定义下的数据组织技术为依据,向上提供定义、管理和查询构件目录所需的服务,高层工具和用户通过这些服务访问构件目录并间接访问构件。目前实现的服务有库管理服务、数据模型服务、构件描述服务、查询服务、构件处理服务和

访问控制服务.

青鸟复用库服务参考 STARS 的 ALF 框架^[2],屏蔽不同构件库设计和实现的差别,为创建和维护构件库系统提供了统一的编程接口.建立在服务接口上的复用工具可以用一致的方式访问符合同一元模型的异质构件库,构件库系统也可以用所有符合复用库服务的工具进行扩展.

多库分布处理机制负责在分布环境中处理构件的分类和查询,向上屏蔽构件库分布细节.特定构件库的包装层在本地数据模式与统一数据模式之间、复用库查询服务请求与本地构件库的查询语言之间进行转换.事务处理服务器负责保证分布构件库的数据完整性和分布查询的原子性;位置服务器处理构件的定位、副本和缓存,向上提供全局物理存储虚像;命名服务器负责处理构件命名、重复构件排除和版本控制,向上提供全局逻辑命名虚像.随着 Internet 技术的发展,分布多库处理已经是现实存在的问题.

4 结束语

青鸟构件库系统的原型实现采用客户-服务器的体系结构,目前的版本部署在局域网上.青鸟构件库系统 JBCL 在概念上基于 JBCOM 构件模型,在构件模型之上建立的构件库数据模型参考了 RIG 组织的 BIDM 模型(IEEE 标准 1420.1).JBCL 采用以剖面分类为主、多种分类方法相结合的分类模式,通过独立于底层平台的应用编程接口提供构件的组织 and 查询服务,并提供查询和分类工具辅助用户和管理员完成各自的任务.

目前青鸟构件库中保存了来自 CASE 和 POS 领域的分析、设计和代码构件,实际验证了以剖面模式为主、多种分类方法相结合的分类方法和查询工具的有效性.在现有构件库的基础上,今后将重点研究 Internet 上的构件库服务.

参考文献:

- [1] LI Ke-qin, GUO Li-feng, MEI Hong, and YANG Fu-qing. An Overview of JB (Jade Bird) Component Library System JBCL [A]. Proc. TOOLS Asia '97 [C]. Beijing, China, Sep. 1997.
- [2] STARS Project. Asset Library Open Architecture Framework Version 1.2 [R]. Informal Technical Report, STARS-TC-04041/001/02, Aug. 1992.
- [3] Shirley V. Browne and James W. Moore. Reuse library interoperability and the world wide web [A]. Proc. of Symposium on Software Reuse Joint Session with ICSE '97 [C]. 1997:183 - 189.
- [4] 常继传,郭立峰,马黎.可复用软件构件的表示和检索 [J]. 计算机科学,1999,26(5):45 - 49.
- [5] W. B. Frakes and T. P. Pole. Proteus: a reuse library system that supports multiple representation methods [J]. ACM SIGIR Forum, 1990, 24:43 - 55.
- [6] WU Qiong, CHANG Ji-chuan, MEI Hong, and YANG Fu-qing. JBCL: An object-oriented component description language [A]. Proc. TOOLS Asia '97, Beijing, China Sep. 1997.
- [7] MEI Hong, YUAN Wang-hong, WU Qiong, and YANG Fu-qing. BD-ConnC++: A C++ program understanding system [J]. Chinese Journal of Electronics, April 1997, 6(2):64 - 69.
- [8] William B. Frakes and Thomas P. Pole. An empirical study of representation methods for reusable software components [J]. IEEE Trans. on Software Engineering, August 1994, 20(8).
- [9] IEEE Standard for Information Technology-Software Reuse-Data Model for Library Interoperability: Basic Interoperability Data Model (BIDM) [S]. IEEE Std 1420.1, 1995.
- [10] C. Lillie and J. Moore. Asset source for software engineering technology (ASSET) software reuse and reengineering [J]. In NISQP Software Reuse and Reengineering Conference, Alexandria, Virginia, Apr. 1991.
- [11] Hafedh Mili, Fatma Mili, and Ali Mili. Reusing software: issues and research directions [J]. IEEE Transactions on Software Engineering June 1995, 21(6).
- [12] Ruben Prieto-Diaz and Peter Freeman. Classifying software for reusability [J]. IEEE Software, 1987, 1:6 - 16.
- [13] Ruben Prieto-Diaz. Implementing faceted classification [J]. Communication of ACM, May 1991, 34(5):88 - 97.

作者简介:



常继传 1975年1月出生.北京大学计算机科学技术系硕士研究生.主要从事软件工程、软件复用和软件构件技术等方面的研究工作.



李克勤 1973年3月出生.北京大学计算机科学技术系博士研究生.主要从事软件工程和软件工程环境、软件复用和软件构件技术等方面的研究工作.