构件形式化描述与模糊检索研究

陈 文[1] 丁晓明[1,2]
(西南大学计算机与信息科学学院 重庆 400715)[1]
(重庆市智能软件与软件工程重点实验室 重庆 400715)[2]

摘 要 在软件复用研究不断深入的情况下,构件的准确描述与高效检索已成为面向构件的软件复用研究的热点和难点。本文参照3C构件模型,提出一种基于形式化方法的、可扩展的构件描述模型,包括构件的功能描述、接口描述、环境依赖描述等,并保留了构件关键字、非功能属性等描述项。在此描述模型基础上提出了构件的分步检索法,并着重论述了包含四级模糊度的构件形式化检索方法,以提高构件的查找效率并兼顾查全率。

关键词 构件描述,构件检索,形式化,语义

Research on the Formal Description and Fuzzy Retrieving of Component

CHEN Wen[1] DING Xiao-ming[1,2]
(Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China)[1]
(Intelligent Software and Software Engineering Laboratory of Chongqing, Chongqing 400715, China)[2]

Abstract As going deep into the study of the software reuse, the accurately describing of software component and high efficiency of component retrieving are become the hotspot and nodus of the studying of the software reuse. In this article, by the reference of 3C model of software component, we put forward a formable, extensible describe model of component, including function describe, interface describe, environment description, etc, and we keep the keyword list and unfunctional attribute description, by this way, we bring forward a multiple steps searching method of component, and emphasize on the discussion on of the four levels of fuzzy searching of component, in order to raise the efficiency of the component searching and take care of the full-scale of the searching results.

Keywords Component description, Component retrieve, Formal, Semantic

1 引言

构件是具有特定功能、可被复用的软件实体。基于构件的软件开发技术是软件复用的一种重要手段,是近年来软件复用研究的热点。要使用基于构件的软件开发技术,发挥构件在软件重用方面的优势,就必须有支持整个软件生命周期并包含有大量可用构件的构件库系统,来提供有效的构件管理和高效的构件查询,这是基于构件的软件开发(Component based software development)成功的关键。而对构件进行有效的管理和高效的检索,依赖于准确合理的构件描述。常用的构件描述方法有刻面描述、编目描述、关键值描述,以及REBOOT模型等。这些描述方法大部分面向构件的使用和组装,对构件检索支持不够充分,同时缺乏对构件、接口元素之间语义关系的描述。另外,以刻面、关键值、编目等非形式化方法进行构件描述或分类为基础的构件检索方法,构件查全率较高,但不能对构件进行精确描述,难以提高构件的查准率,从而导致查找效率低。而基于形式化的构件描述方法,如基于语义网络的描述方法、基于特征空间的语义定义方法等,对构件的语义描述较为精确,但都基于严密的数理逻辑和大量的专用定义符号,不适合多数构件用户用来构造查询语句,以进行构件查找。且以这类描述为基础的构件检索易忽略与查询构件功能相近的库构件,导致构件的查全率低。

本文研究的重点是以3C构件模型为基础构造一种描述准确、可扩展、易操作的构件形式化描述模型,包括构件的接口描述、功能描述、属性描述、环境依赖描述、附加文档描述5个部分。在描述模型的基础上,本文提出了一种将非形式化检索、形式化检索与手工检索相结合的多步构件检索方法,以提高构件查全率与查准率,并着重论述了包含四级模糊度的构件形式化检索方法。

2 构件的描述模型

本构件描述模型以3C构件模型为基础。3C构件模型是关于构件模型的一个指导性模型,是一些系统工程领域专家在1989年的Reuse in Practice Workshop上提出的。该模型由构件的3个不同方面的描述组成,即概念(concept)、内容(content)、语境(context)。概念是关于"构件做什么"的抽象描述,可以通过概念去理解构件的功能。内容是概念的具体实现,描述构件如何完成概念刻画的功能。语境描述构件和外围环境在概念级和内容级的关系。

根据3C构件模型的特点,并加入了易于构件检索的描述属性,本构件的描述模型包括构件的接口描述、功能描述、属性描述、环境依赖描述、附加文档描述5个部分,即:

Component _ description = {〈Interface _ description〉, 〈function_description〉, 〈Property_description〉, 〈environment _rely_description〉, 〈Attached_file〉}

其中,接口描述是构件描述的主体,对构件各功能项进行

陈 文 硕士生,系统分析师,主要研究领域为软件工程;丁晓明 副教授,主要研究领域为软件工程。

精确描述；功能描述是对构件整体功能的概括性描述，主要用于构件检索中的大类划分；属性描述指与构件的执行过程相关的属性描述，包括功能属性和非功能属性的描述；环境依赖主要对构件的运行实体所必需的软硬件环境进行描述；附加文档是对构件的自然语言描述，可以弥补形式语言描述和自动检索的不足，给复用者提供更完整的信息。上述5部分中，功能描述对应于3C模型的概念部分，接口描述和属性描述对应于内容部分，环境依赖描述对应于语境部分。

2.1 接口描述

接口实现构件的各功能项。任何接口功能的执行必将导致相关状态变量的改变，例如数据的产生、更新、删除等，否则功能项的执行无意义，故接口的功能可以通过其执行前后相关状态变量值的变化对比来体现。相关状态变量既包括构件内部的属性或变量，也包括构件外部环境变量（如数据库记录、磁盘文件、网络数据、指令信号等）。

接口描述的形式化定义如下：

Interface_description={〈Interface_ID〉〈Signature〉〈Pre_condition〉〈Post_condition〉〈Refer_Vairant〉} Interface_ID=string * | number * ,not null;

Signature=Interface_name(Parameterlist) Return_type;

Pre_condition=Restrication_Declarition(Refer_Variant *);

Post_condition=Restrication_Declarition(Refer_Variant' *);

Refer_Variant=Base_Type | UserDefine_Type | Out_Type;

Refer_Variant'= Base_Type |UserDefine_Type | Out_Type;

Restrication_Declarition()=All referred variants' value state;

Base_type=int | uint | float | double | boolean | long | char | string | …;

UserDefine_Tpye=struct | union | enum | class | …;

Out_Type=file | outer | printer | port |…;

Parameterlist=Null | Base_tpye | UserDefine_Type | Out_Type [,Parameterlist] * ;

Interface_name = string * | number * ,not null;

Return_Type=Base_tpye | UserDefine_Type | Out_Type | void;

对各主要定义元素的解释如下。

(1)Pre_condition：接口功能项调用前相关状态变量必须满足的状态取值条件。

(2)Post_condition：接口功能项调用后相关状态变量必须满足的状态取值条件。

(3)Signature：接口签名，指接口函数声明，用于构件调用。

(4)Refer_Variant：指与接口功能函数执行前相关的所有变量，可包括基本变量类型(Base_Type)，用户自定义变量类型(UserDefine_Type)，以及其余非基本变量类型(Out_Type)。Refer_Variant'指功能执行后的所有相关变量，用于与Refer_Variant相区别。

(5)Restrication_Declarition：指对状态变量的取值要求。

(6)"…"表示可扩展，下同。

2.2 功能描述

function_description={〈application_domain〉〈function_type〉〈keyword_list〉}

application_domain=string * ,not null;

function_type=string * ,not null;

keywordlist=string[,string *];

对各元素的解释如下。

(1)application_domain指构件的应用领域，如企业(enterprise)、金融(finance)、网络游戏(network_game)、银行ATM、构件生产者或构件用户可以使用条件选择机制来自动生成领域的关键字描述，以规范领域描述属性值，它保留了可扩展接口，可随着软件应用领域的增加而进行扩展。

(2)Function_type指构件的功能类型，如：Manage_Information_System(管理信息系统)、ERP(企业资源计划)、Real-Time_Control_System(实时控制系统)、Database_Manage_System(数据库管理系统)、构件的功能类型属性值的生成也可以使用条件选择机制进行选择生成，并保留功能类型的扩展接口。

(3)Keywordlist：指构件功能描述的关键字列表，如information_system，network_based、database_management_system、关键字列表用于保存最能反映构件功能特征的关键字组成，用户可以根据关键字列表的值对构件进行基于关键字的构件检索，以提高构件的查全率。

2.3 属性描述

Property_description={〈functional_property〉〈unfunctional_property〉}

functional_property=〈base_attr〉;

unfunctional_property={〈execute_time〉〈security_level〉〈user_interface〉〈reused_times〉…}

base_attr=integer |real | character |boolean |string |arrary| report_paper | data_record |class | struct |file |handle |signal |…[〈expla〉];

user_interface=IE_Graphic | Windows_Graphic | Character |Null…[〈demo〉];

对各元素的解释如下：

构件的属性描述(Property_description)分为功能属性描述(functional_property)和非功能属性描述(unfunctional_property)两部分。功能属性是构件执行过程中的创建、更新、删除等操作的作用对象，如数据库记录(database_record)、指令信号(signal)等。〈expla〉指对属性的简短的自然语言描述，用于帮助构件用户对构件属性的理解。非功能属性是与构件的执行过程相关的属性，如构件的执行时间(execute_time)、构件的安全级别(security_level)、构件执行时的用户界面风格(user_interface)，如浏览器界面(IE_Graphic、Winidows界面风格(Windows_Graphic)、字符界面(Character)、无界面(Null))，等。构件被复用的次数(reused_times)：通常构件被复用的次数越多，表明构件的稳定性越好，通用性越强。〈demo〉指构件演示文件，为用户提供构件运行时的演示图片或图像，使用户对构件的理解更直观、形象，便于用户进行构件选择。

2.4 环境依赖描述

environment_rely_description={〈software_env〉〈hardware_env〉}

software_env={〈operation_system_name〉〈database_name〉〈rely_software_name〉〈other_component〉〈dll_list〉〈file_list〉…}

operation_system_name=string * |digital * ;

database_name= string * |digital * ;
rely_software_name= string * |digital * ;
other_component={〈component_name〉〈component_filename〉}
component_name= string * |digital * ;
component_filename= string * |digital * ;
dll_list={〈dll_name〉〈dll_path〉}
dll_name= string * |digital * ;
dll_path= string * |digital * ;
file_list={〈file_name〉〈file_path〉}
file_name= string * |digital * ;
file_path= string * |digital * ;
hardware_env={〈computer_lowest〉〈network〉
〈usb_list〉…}

环境依赖主要是指构件成功运行必不可少的软、硬件环境。环境依赖描述可为用户的构件选择提供重要的参考信息，如软件兼容性、硬件成本等，并且可以为用户的构件运行环境搭配或构件的适应性修改提供指导标准。

对各元素的解释如下。

software_env：指软件环境、包括操作系统、数据库、依赖软件、相关构件、动态链接库列表、文件列表等；hardware_env：指硬件环境，包括最低计算机配置、网络要求、usb 设备等。软、硬件环境均可根据构件的具体要求而进行描述上的扩展，以保证构件描述的完整性。

2.5 附加文档

Attached_file={〈reused_history〉〈modify_history〉〈version_controlfile〉〈develop_environment〉〈develop_document〉〈source_document〉〈test_document〉〈create_time〉
〈librarity_time〉〈librarity_name〉…}

构件的附加文档用于为构件用户或构件维护者提供自然语言或半形式化语言描述的构件详细信息。

对各元素的解释如下。

reused_history：构件的复用历史，便于用户从以往用户的复用经验上加深对构件的理解；modify_history：构件的修改历史；version_controlfile：版本控制文件；develop_environment：构件的开发环境；develop_document：开发文档以及 source_document(源文件)和 test_document(测试文档)。这 6 个属性文档主要用于构件的维护者对构件的修改和维护，可以通过访问权限限制，只对构件维护者可见。构件的 create_time(生成时间)、librarity_time(入库时间)、librarity_name(入库名称)用于描述构件的生成日期和存储信息。

3 构件的检索匹配

拥有大量可复用构件的构件库是基于构件的软件开发(CBSD)的基础，但海量的库构件易造成数据堆积不易于查找的问题，构件用户往往难以在可接受的时间范围内从构件库中检索到所需的构件。因此，提高构件检索匹配的查全率和查准率是基于构件的软件开发成功的关键。

传统的基于非形式化方法的刻面、关键值、编目等构件检索方法，执行效率低，查准率低，但查全率较高。而传统的基于形式化方法的构件检索，执行效率高，查准率较高，但由于形式化逻辑精确匹配的要求以及用户构造形式化查询语句的能力限制，导致构件的查全率低。构造有效的构件检索方法一直是构件复用研究的热点、难点。本文在基于前述的构件描述模型的基础上，提出一种将形式化检索和非形式化检索相结合的分步构件匹配检索方法。构件检索分为 3 个阶段：首先基于构件功能描述中的应用领域描述(application_domain)、功能类型描述(function_type)、关键字列表(keywordlist)、非功能属性描述中的构件执行时间(execute_time)、安全级别(security_level)、用户界面(user_interface)、复用次数(reused_times)以及环境依赖描述中的各属性值等可进行传统的非形式化构件检索，以提高构件的查全率，并缩小检索范围。在得到第一步检索的结果后，用户既可以直接进入第三步手工筛选阶段，通过构件非功能属性中的构件演示示例描述 demo 以及附加文档中的自然语言描述文档进行手工选择，也可以进入第二步形式化检索阶段，基于属性描述中的功能属性和接口描述中的功能调用前件、后件关系进行构件的形式化检索，以提高构件的查准率，在第二阶段形式化检索的结果基础上，进入第三阶手工筛选阶段。

刻面分类、关键值检索等传统的非形式化构件描述与检索匹配方法研究和应用较广，在文献[14,16,20]等中已有诸多描述，在此不再赘述。下面对构件形式化检索匹配方法进行详细论述。

3.1 基于形式化方法的构件检索

传统的形式化检索方法，查准率较高，但由于精确匹配的要求，导致一些功能与查询构件类似的库构件易被忽略。并且，由于构件用户在查找构件时对查询构件的具体要求和理解程度不同，因此需要设置不同级别的精确查找，引入检索模糊性，以提高构件形式化检索的查全率。参照文献[17,18]中构件查找力度划分的思想，本文将构件查找精确度分为以下 4 个等级：①匹配级；②相容级；③推导级；④松散级。四级构件匹配力度适宜于精确度由高到低的构件查找。

设 Q 为查询构件描述，S 为库构件描述，D 为构件库。引入力度满足函数 Sat(para1, para2)，若在给定的比较域内，para1 的有效力度大于或者等于 para2，则 Sat 为真，否则为假。例如变量精确度方面有 Sat(double, integer)为真，Sat(integer, double)为假，大小值比较方面，Sat(name(I)＞10, name(J)＞7)为真，Sat(name(I)＞7, name(J)＞10)为假。name()为一系列命名替换函数。

(1)匹配级

(∀Q. property∈ base_attr, ∃S∈ D, Sat(S. property, Q. property)) ∧ (∀Q. Interfacei. Precon∈ Pre_condition, ∃S. interfacej, Sat(S. Interfacej. Precon, Q. Interfacei. Precon)) ∧ (∀Q. Interfacei. Postcon∈ Post_condition, Sat(S. interfacej. Postcon, Q. Interfacei. Postcon))

匹配级的构件检索要求构件的功能属性和功能前件、后件都要严格匹配，这一级匹配查准率高，但查全率低，一些通过少量入口参数修改就能实现查询构件要求的库构件容易被忽略。

(2)相容级

(∀Q. property∈ base_attr, ∃S∈ D, Sat(S. property, Q. property)) ∧ (∀Q. Interfacei. Precon∈ Pre_condition, ∃S. interfacej, Sat(S. Interfacej. Precon, Q. Interfacei. Precon)) ∧ (∀Q. Interfacei. Postcon∈ Post_condition, S. interfacej. Postcon=＞Q. Interfacei. Postcon)

相容级的构件检索对构件匹配相似性力度要求有所降低，不要求库构件和查询构件的接口功能后件匹配力度满足函数 Sat，只要求库构件 S 的执行结果能够逻辑推导出查询构件的执行结果。满足相容级匹配要求的构件匹配需要在主调模块对库构件的执行结果进行一定程度的修改或转化，以满

足查询构件的要求。

(3)推导级

(∀Q. property∈base_attr, ∃S∈D, Sat(S. property, Q. property))∧(∃Q. Interfacei. Precon∈Pre_condition, ∃Q. Interfacei. Postcon∈Post_condition, (∃S. interfacej, S. Interfacej. Precon=> S. interfacej. Postcon)=> (Q. Interfacei. Precon =>Q. Interfacei. Postcon))

推导级的构件匹配检索要求通过库构件S的执行逻辑能够推导出查询构件Q的执行逻辑。满足推导级匹配要求的构件匹配需要在构件的主调模块中对库构件接口功能的执行前件进行附加的判断或修改,并对库构件的执行结果进行一定程度的修改或转化,以满足查询构件的要求。

(4)松散级

(∀Q. property∈base_attr, ∃S∈D, Sat(S. property, Q. property))∧(∀Q. Interfacei. Postcon∈Post_ condition, ∃S. interfacej, S. interfacej. Postcon=> Q. Interfacei. Postcon)

松散级的构件匹配检索要求库构件S的执行结果能够逻辑推导出查询构件的执行结果,并且对构件的接口前件不做要求。这一级匹配要求最低,能够提高构件的查全率,多用于用户对查询构件的需求尚不明确,需要进行模糊查找以找出与需求相似的构件再进行手工选择的情况。满足松散级的构件匹配在主调模块中的附加工作量较多,需要对库构件的接口前件进行全面判断及限制并对执行结果进行修改或转化,以满足查询构件的要求。

3.2 应用举例

为简化描述,假设查询构件和库构件都只提供了一个功能接口,使用Z语言进行Q与S的功能属性和接口形式化描述如下:

Q. interface:

a?, b?: integer
c!: Boolean
Pre_condition: a>1, b>1
Post_condition: c! =(a>b)

S. interface:

a?, b?: double
c!: Boolean
Pre_condition: a>10, b>10
Post_condition: c! =(a^2>b^2)

显然,S和Q不满足匹配级要求,Sat((a>b),(a^2>b^2)为假,因为后件中的条件不能确定 a,b 的取值范围,不能满足Sat力度函数为真的要求。

但S和Q满足推导级匹配,因为下式成立:

((a>1,b>1)=>(c! =(a>b)))=> ((a>10,b>10) =>(c! =(a^2>b^2)))

使用带数学公理扩展的一阶谓词证明如下:

①a>1,b>1=>c! =(a>b)	已知前件
②a>10,b>10	引入规则
③a>10,b>10=> a>1,b>1	公理
④a>10,b>10=> c! =(a>b)	1,3及合式公式
⑤a>10,b>10,a>b=>a^2>b^2	公理
⑥a>10,b>10=>c! =a^2>b^2	4,5及合式公式

结束语 本文提出的基于形式化描述的、可扩展的构件描述模型是基于学术界公认的3C构件模型并加入了与构件检索相关的其它重要属性而组成的,使得构件描述较为全面、准确。在描述模型基础上提出了构件分步检索法。本文最后对多模糊级的形式化检索方法进行了论述,提出了四个不同模糊级的形式化检索:匹配级、相容级、推导级、松散级。四级模糊检索能够兼顾形式化检索的查全率与查准率,克服了以往构件形式化检索中查准率高、查全率低的困难。

参 考 文 献

[1] Yourdon P C. Object-orienteddesign[M]. Prentice-Hall, 1991: 1-82

[2] Reuse Library Interoperability Group. RIG Uniform DataModel (BIDM). PRS-0001, April 1993, revised: January 1995

[3] Cohenetal S G. TechnicalReport. CMU/SEI-91-TR-28, ESD-91-TR-28. June 1992

[4] Mili H, Valtchev P, Di-Sciullo A, et al. Automating the Indexing and Retrieval of Reusable Software Components // Proceding of the6th International Workshop NLDB'01. Madrid Spain, 2001: 75-86

[5] Pressman R S. Software Engineering: A Practitioner's Approach. Fifth Edition. China Machine Press, 2002

[6] 杨芙清,邵维忠,梅宏. 面向对象CASE环境JBⅡ型系统的设计和实现. 中国科学,1995(5)

[7] 青鸟III型项目组. 青鸟构件模型. 内部技术报告[R]. 北京大学计算机科学与技术系,1997

[8] 杨芙清. 软件复用及相关技术[J]. 计算机科学,1999,26(5):1-4

[9] IauK-K, Ornaghi M. A Formal Approach to Software Component Specification // Proc. of Specification and Verification of Component-based Systems Workshop at OOPSLA. 2001

[10] 王渊峰,薛去蛟,等. 刻面分类构件的匹配模型 [J]. 软件学报,2003,14(3):402-408

[11] 王渊峰,张涌,任洪敏. 基于刻面描述的构件检索[J]. 软件学报,2002,13(8):1546-1550

[12] 任洪敏,钱乐秋. 构件组装的形式化推导研究[J]. 软件学报,2003,14(6):1067-1073

[13] 贾育,顾毓清. 基于领域特征空间的构件语义表示方法[J]. 软件学报,2002,13(2):312-315

[14] 孟闻天,张维石,等. 一种基于刻面分类描述的软件构件查询方法[J]. 计算机工程与应用,2005(12): 61-64

[15] 徐正权,王家兵,等. 软件构件表示与检索形式化的研究与进展[J]. 计算机科学,2003,30(7):99-102

[16] 刘大昕,赵磊,王卓. 一种基于刻面分类和聚类分析的构件分类检索方法[J]. 计算机应用,2004,24(6):89-90

[17] 李晓博,缪淮扣,刘静. 基于形式规格说明的构件匹配[J]. 计算机应用与软件,2006,23(10):10-12

[18] 徐丽萍,贾红卫,卢炎生. RTSC:一种具有精确语义的实时构件描述机制[J]. 计算机科学,2005,32(8):205-208

[19] 林浩. 一种面向检索的构件描述方法[J]. 山东理工大学学报,2005(3):72-74

[20] 贾晓辉,陈德华. 基于刻面描述的构件查询匹配模型及算法研究[J]. 计算机研究与发展,2004,41(10):1634-1638

[21] 马亮,孙家. 基于规约匹配的构件检索[J]. 小型微型计算机系统,2002,23(10):1153-1157