

一个集成式的软件构件描述框架

张 涌 王渊峰 钱乐秋

(复旦大学计算机科学系 上海 200433)

摘 要 该文提出了一个可扩展、可裁减的集成式软件构件描述框架。该框架采用了相宜的方法对构件的接口、语义、问题空间等方面进行描述,并且通过 XML 提供了可扩展的机制。实践证明,在该框架基础上实现的构件库管理系统体现了较好的构件描述与检索能力。

关键词 构件描述,可复用构件,XML

中图法分类号 TP311

An Integrated Software Component Description Framework

ZHANG Yong WANG Yuan-Feng QIAN Le-Qiu

(Department of Computer Science, Fudan University, Shanghai 200433)

Abstract The description method of reusable software components is an indispensable part of the reusable component repository technique. As the component repository scales up, the components complexity increases and the components' semantics enriches, a single description method has not been competent for the all components in the repository. The author considers that the participation of the component designer and the component repository manager is the precondition of a precise and appropriate component description. An excellent component description should include the description not only for the common information but also for the particular unit. This paper proposes an extensible, tailorable and integrated component description framework to solve this problem, which adopts respective method to describe interfaces, semantics and problem space of a reusable component and applies XML to provide extensibility. This framework includes the following sections: the general description section, the component interface description section, the component's semantics description section, the problem space description section, and intra-component relationship description section. We use natural language to describe the component's common information in the general description section. A suite of pseudo-code is defined to describe its called specification in the component interface description section. Predicate and set theory are adopted to describe the semantics of the component in the component semantics description section. The problem space of the component is described by catalogue language. The organization relationship description section is utilized to describe the relationship between any parts in a component. A management system of reusable component repository has been built based on this description framework, and a better retrieval performance is obtained in some experiments.

Keywords component description, reusable software component, XML

收稿日期: 2001-04-02; 修改稿收到日期: 2001-12-14 本课题得到上海市教委重点学科建设项目的资助 张 涌, 1973 年生, 博士研究生, 主要研究领域为软件复用、软件体系结构、软件测试 E-mail: yongzhang@fudan.edu.cn 王渊峰, 1974 年生, 博士研究生, 主要研究领域为软件复用、构件库管理系统 钱乐秋, 1942 年生, 教授, 博士生导师, 主要研究领域为软件自动化、软件复用、软件自动测试

1 引言

随着对软件复用及应用研究的深入, 可复用的软件构件库作为软件复用的一项重要的基础设施已越来越得到产业界与学术界的重视^[1]. 其中, 对构件的描述方法与检索技术的研究已成为两个主要的研究热点

构件的描述方法是构件检索技术的基础, 3C 模型和 REBOOT 模型^[2]是构件描述中的两个重要的描述模型. Latour 等人提出了 3C 模型^[3], 认为一个理想的构件的描述应当涵盖以下 3 个方面——概念 (concept)、内容 (content)、语境 (context). 其中, 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所刻画的功能; 语境是描述构件和外围环境在概念级和内容级的关系, 语境刻画构件的应用环境, 为构件的选用和适应性修改提供指导信息. REBOOT 模型是在 REBOOT 项目中总结提出的^[4], 它实质上是一个刻面分类模型. 该模型认为: 可以用有限维信息空间的术语组合从若干个刻面的综合角度来刻画一个构件. 具体到 REBOOT 项目, 他们提出的刻面有以下 4 个: 依赖 (dependencies)、抽象 (abstraction)、操作 (operations)、操作对象 (operates on). 另外该模型还提出了用同义词典 (thesaurus) 来提供术语间语义关系的描述手段

在实际使用中, 这些模型需要被转化为具体的语言描述形式, 目前所提出来的语言主要分两类: 规约语言与编目语言. 规约语言不仅可被用于构件的签名匹配和规约匹配, 还可以用于实现自动化的构件验证与组装. 在规约语言的研究中主要经历了操作规约语言 (例如 CSP)、接口规约语言 (例如 DL)、表达式规约语言 (例如 OCL) 等几个发展阶段^[5]. 目前研究的第四代规约描述语言的工作主要集中在构件的语境描述上, 即构件使用的语境信息. 有代表性的工作有 KDL 语言和青鸟构件描述语言^[5,6], 它们不仅能够描述构件, 更提供了对构件子系统的描述能力. 这部分的工作集中于 90 年代中后期, 它们的成熟有待于进一步的实践与推广. 对于编目语言, 根据所使用的技术可分为 3 类^[7]: 图书馆与信息科学方法、人工智能方法和超文本方法. 目前这一领域绝大部分研究工作集中在图书馆与信息科学方法, 所提出的具体方法有枚举、刻面、关键字等等^[7].

构件的检索技术在很大程度上依赖于构件的表示方法, 目前所提出的检索方法, 根据表示方法的不同可分为以下 5 类: 基于表达式规约表示的规约匹配^[8], 基于接口规约表示的签名匹配^[9], 基于编目术语的特征轮廓匹配^[10], 基于构件行为的采样匹配^[11], 基于知识表示的人工智能检索方法^[10,12].

但是, 随着构件库的规模的扩大以及构件的复杂性增加和语义性丰富, 用单一的构件表示方法以及在此基础上的检索算法来胜任整个构件库的检索任务越来越力不从心^[8,10-12].

Frakes 在他的论文^[7]中对几种构件的编目表示方法下的构件库的查准率、查全率与效率作了细致的实验, 得到以下几点值得注意的结论:

- (1) 在查准率、查全率上每一种方法都差不多, 没有明显优劣;
- (2) 每种方法都有一定的固定用户集, 用户对方法的选择没有明显的偏爱与偏弃;
- (3) 每种方法对同一检索得到的结果集都不尽相同, 但没有任何方法的检索结果集是最优的, 它们互有补充

这说明在大规模的构件库中, 要保证有效的查全率、查准率和使用率, 构件的表示方案应设计为一种可扩展、可裁减的表示框架结构. 它应当能提供对现有的表示方法的组合以及对将来的新的表示方法的插入的能力, 否则在需要加入新的描述方法描述的构件进构件库时, 将会给构件库描述系统带来巨大的改动, 甚至无法改动 (因为还涉及到存储方案、检索算法的相应改动).

本文提出了一个集成式的构件描述框架, 它的主要特点是提供了可扩展、可裁减的描述能力. 在该构件描述框架中既使用了自然语言的描述手段又使用了形式化的方法, 使得构件的描述者可以根据构件的具体情况对构件的描述进行组织, 并在此基础上提供了检索工具的有效组合, 以进一步保证构件库的查准率、查全率和使用率

2 集成式的软件构件描述框架

我们认为对一个软件构件的准确、合适的描述既需要构件的设计者的参与也需要构件库管理人员的介入. 既存在公共通用信息的描述部分也存在专向个体的可扩展的描述部分. 本文提出的可扩展的构件描述框架综合了这些考虑, 如图 1 所示

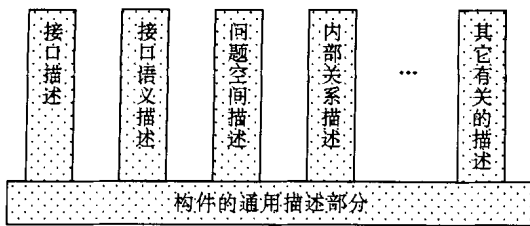


图 1 构件的可扩展描述框架

其中, 构件的通用描述主要是对构件的一些通用信息用自然语言进行描述; 构件接口描述定义了一组伪程序代码以描述其调用规范; 构件的语义描述对构件接口的语义信息用谓词逻辑以及集合论进行说明; 问题空间的描述描述构件的问题领域内一些有关信息, 可以采用一些编目语言对它进行描述; 构件内部组成关系主要描述构件的内部各个组成部分之间的各种关系(这部分描述主要是为构件的自动化组装和验证考虑)。其它描述部分表示该框架提供了可扩展的描述能力。我们对它们依次介绍如下。

2.1 构件的通用描述

对构件的通用描述部分的设计, 我们主要参考了复用库可互操作性组织(RIG)为软件复用库间共享软件构件而制订的UDM 数据模型^[13]以及青鸟

构件模型^[14]。它描述了各种粒度、抽象层次的构件都应具有的一些通用信息, 例如所适用的范围, 软件环境信息等。它的描述采用自然语言进行描述。主要描述字段如下所示:

构件的名称;

构件编写语言;

构件所需软硬件环境;

关键词;

构件中包含的文件;

作者;

发布日期;

使用次数;

构件类型: 需求分析构件、设计构件、代码构件、测试构件。

构件的通用描述部分为各异质构件库间交流构件提供了一个通用的信息模型。

2.2 构件接口描述

构件的接口描述部分描述了构件对外部的接口类型。接口描述部分所使用的参数部分与 C++ 所提供的基本参数类型相同。图 2 列出了接口规约部分的BNF 范式, 其中非终结符用斜体字加“*<*”来表示, 终结符用黑体字来表示, “[]”表示任选。

```

<component interface specification> ::= Interface specification
                                     Interface
                                     <function declaration>
                                     End Interface

<function declaration> ::=
    <function-name> [<para-list>] return <return-type>
<function-name> ::= <function-name> <character> | <function-name> <number> | <character>
<para-list> ::=
    <para-declaration> | <para-list> ; <para-declaration>
<para-declaration> ::=
    <para-name> ; <type>
<para-name> ::= <para-name> <character> | <para-name> <number> | <character>
<type> ::=
    pointer of <prime-type>
    reference of <prime-type>
    <prime-type>
    structure of <prime-type>
<prime-type> ::=
    <built-in type>
<built-in type> ::=
    long | long long | int | short | float | double |
    long double | char | wchar | bool
<return type> ::=
    <type> | void
  
```

图 2 构件接口描述的 BNF 范式

2.3 构件接口的语义规约

构件的接口描述仅提供给用户构件的接口信息, 但这些构件所提供的处理的语义信息对用户来说并不明确, 利用接口的描述, 用户仅知道如何定义和使用接口, 但并不知道构件做些什么。在语义级别

上我们应该刻划构件中某个操作所期望得到的结果。对于构件的语义, 我们主要关心的是构件的初始状态以及一个操作的执行所导致的构件中某部分状态的变化。其中某个状态可以看作是一个命题公式, 它可以递归定义如下:

- (1) T, F 以及原子命题都是公式;
- (2) 若 P, Q 是公式, 则 $\neg P, P \wedge Q, P \vee Q$ 都是公式;
- (3) 若 P 是公式, x 是 P 中的变量, 则 $\forall xP, \exists xp$ 都是公式;
- (4) 公式仅通过有限次使用(1)(2)(3)获得.

由于自然语言表达上的缺点, 在这里我们采用一阶谓词逻辑对接口的功能进行规约. Z 语言是基于—阶谓词逻辑和集合论的规格说明语言, 它具有很强的表达能力, 且它可以对 Z 规格说明进行推理和证明. 本文中的构件语义规约采用类似于 Z 语言的描述机制. 保留一些 Z 语言中的符号, 例如变量 + “?” 表示输入变量, 变量 + “!” 表示输出变量, Z 语言的具体使用可参见文献^[9]; 由于 Z 语言中缺乏一些默认的数据类型, 我们增加一些内建数据类型, 这些内建数据类型与上一部分讨论的接口描述中的内建数据类型相同. 图 3 是对于构件接口语义规约的语法, 其中 *User_Type Definition* 表示用户自定义类型声明, **Declaration** 为声明部分, **Predicate** 是谓词部分.

Semantic-Specification:
 {User_Type Definition}
 Semantic Interface-name
 Declaration
 {Para-name:type}
 Predicate
 {Predicate description}
End Semantic-Specification

图 3 构件接口的语义规约语法

2.4 问题空间描述

我们可以把软件构件看成是某个问题的解决方案, 通常构件的查询者在查询的初始阶段可能对构件的规约或接口还不甚清楚, 有时他们可能主要对构件的整体功能更感兴趣. 这时需要关心构件的编目描述, 即构件所解决的问题空间的描述. 目前提出的编目方法有很多, 实用的有枚举、刻画、受控关键字和非受控关键字等方法^[13]. 在我们的描述框架中这些方法通过容器技术被设计成是可插拔的, 即可以根据实际的描述需要有选择的配置. 我们用一个简化的 XML 的 DTD(文档模板定义)给出我们对该部分描述的设计, 见图 4.

<? XML version="1.0" encoding="GB2312">
<! DOCTYPE 问题空间描述[
 <!ELEMENT 描述方法(枚举|刻画|关键字)+>
 <!ELEMENT 枚举(#PCDATA)>
 <! ATTLIST 枚举 ENTITYS #REQUIRED
 枚举分类方案 ID #REQUIRED>
 <!ELEMENT 刻画(#PCDATA)>
 <! ATTLIST 刻画 ENTITYS #REQUIRED
 刻画分类方案 ID #REQUIRED>
 <!ELEMENT 关键字(#PCDATA)>
 <! ATTLIST 关键字 ENTITYS #REQUIRED
 关键字分类方案 ID #REQUIRED>
>]

图 4 构件问题空间描述 DTD

2.5 构件内部组成部分之间的关系描述
及扩展描述部分

构件内部的组成部分之间存在各种关系, 它们

主要有 send_message, transfer_data, call 等. 这一部分的描述语法如图 5 所示(对该部分的设计主要出于对构件组装的自动化支持的考虑).

Relation_Specification
 function_name[Call|Send_Message|Transfer_data] function_name
Relation_Specification End

图 5

另外, 在往描述框架中添加新的描述方法时, 我们只需要在相应的 DTD 中加入描述模板定义, 就可以有对构件进行该描述方法下的描述, 从而实现描述框架的扩展.

2.6 应用示例

下面我们用一个栈构件的示例说明如何使用以

上讨论的构件描述框架进行构件的描述, 该构件包括 3 个功能, *pop* 是对栈顶元素出栈操作, *push* 是元素进栈操作, *depth* 是计算栈的元素个数操作, 由于篇幅考虑, 我们只给出了 *Stack* 的接口描述、语义描述、问题空间的描述和构件内部组成部分之间的关系描述, 并且在问题空间的描述中只给出一个包含应用领

域、功能、相关构件 3 个刻面的刻画描述, 另外为了可视方便, 我们删去了其中的 XML 语法标记

Specification Stack

Problem Specification:

刻画描述 刻画分类方案= "Facet_20876" 匹配

算法= "RevivelFor Facet_20876"

应用领域: 这是一种通用的抽象数据结构 (与领域无关), 实现了 FIFO 的原则

功能: ADT, FIFO, Pop, Push, Depth

相关构件: Queue, Tree, List

/刻画描述

Problem Specification End

Interface Specification:

Interface:

Function push (A: integer) return void

Function pop () return integer

Function depth (S: Stack) return integer

Interface Specification End

Semantic Specification:

S: P integer

Semantic push

Declaration:

S1: S

A?: integer

S1: S

Predicate:

A? is element of integer

S1 = S add A? && 1 A

End Semantic

Semantic pop

Declaration:

S1: S

A!: integer

S1: S

Predicate:

depth (S) > 0 | depth (S1) = depth (S1) - 1

&& 1 A! is element of S

End Semantic

Semantic depth

Declaration:

S1: S

A!: integer

Predicate:

A! = # S1

End Semantic

Semantic Specification End

Relation Specification

pop call depth

Relation Specification End

Specification End

3 实现考虑

我们在一个具体的构件库管理系统中的构件描述部分实现了上述描述框架 在具体实现中我们应用了 XML 技术及相关的一些协议标准 图 6 是系统中关于描述部分的一个简化示意图

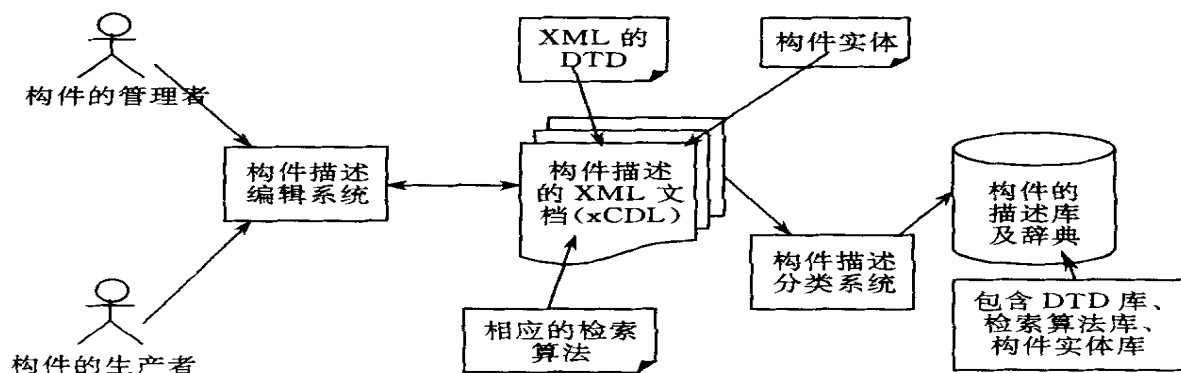


图 6 描述子系统简化示意图

我们之所以选择 XML 作为这种可扩展的构件描述框架的实现技术, 主要出于以下几点考虑:

1. XML 是一种可自定义标记的语言, 它的 DTD 实质是一种元模式的数据描述语言, 即它提供了语法元素、元素属性及语法解析规则的定义方法 这使得它的描述能力具有较大的丰富性和可扩展能力, 这正是我们的描述框架所注重的 我们可以为一个构件描述的每一个描述部分定义各自的 DTD, 然后对每一份 DTD 下的实例 XML 描述在框架下进行组合, 形成一份对该构件而言裁减适宜、表达恰当

的 XML 描述, 我们称这种基于 XML 的可扩展的构件描述语言为 xCDL.

2. XML 中的扩展链接技术 (X Pointer) 和名字空间技术为 xCDL 中各描述部分的组装提供了丰富的支持, 并且为这个框架的可扩展、可裁减机制提供了无缝的实现, 可以较好地保证描述数据的一致性

3. XML 中的转换机制 (XSLT) 为各 XML 描述之间的转换以及 XML 描述的显示提供了较强的控制, 它使得我们有更多的选择、更方便的实现来进

行相关的工作

4 *XML* 不仅可以通过链接技术将每一描述部分与相应的匹配算法链接起来, 实现面向构件不同表示部分的检索算法的有机组合, 而且 *XML* 本身还提供了 *XPATH* 一套查询机制, 有力地支持了检索的实现

5 *XML* 是一种跨平台的开放式的语言, 它的描述必将为构件库在 *WWW* 方向发展奠定坚实的基础

实现了该描述框架的构件库管理系统目前正在原型的基础上进行下一步的实现, 就目前的效果来看, 由于它实现了面向构件的多种表示方法的有机组合, 并能够根据相应的表示部分提供相对应的检索手段, 从而进一步完善了构件库的查询与检索服务

4 总 结

根据一定规模的软件构件库对软件构件描述能力全面、语义丰富和便于检索的需要, 本文提出了一种可扩展的构件描述框架, 它包括了构件的接口描述、语义描述、问题空间描述、构件内部组成关系描述 4 个可裁减的描述部分, 并且根据实际情况的需要它还可以包融进其它的构件描述方法。另外, 我们在一个具体的构件库管理系统中对该框架进行了 *XML* 实现, 并在 *XML* 实现中将各描述部分和与之相对应的匹配算法链接起来, 提供了针对各个构件的匹配算法的有机结合。有理由相信: 实现这一套机制的构件库管理系统相对于传统的基于单一表示方法的构件库而言, 将提供进一步完善的构件表示与检索服务, 从而促进软件复用的进一步深入与发展

参 考 文 献

- 1 Mili H, Mili F, Mili A. Reusing software: Issues and research directions. *IEEE Trans Software Engineering*, 1995, 21(6): 528- 562
- 2 Whittle B. Models and languages for component description

and reuse. *ACM SIGSOFT Software Engineering Notes*, 1995, 20(2): 76- 89

- 3 Latour L, Wheeler T, Frakes B. Descriptive and prescriptive aspects of the 3Cs model: SETA1 working group summary. CASE Centre, Syracuse University, New York: Technical Report 9014, 1990
- 4 Morel J M, Faget J. The REBOOT environment. In: Proc 2nd International Workshop on Software Reuse (REUSE'93), Lucca, Italy, 1993. 80- 88
- 5 Kiniry J R. Leading to a kind description language: Thoughts on component specification. Caltech Computer Science Technical Reports CS-tr-99-04, 1999. http://resolver.library.caltech.edu/caltechCSTR:1999_cs-tr-99-04
- 6 Jade Bird III Project Team of Peking University. The manual of jade bird component description language (JBCDL). Technical Report, 1997 (in Chinese)
(北京大学青鸟III项目组 青鸟构件描述语言 JBCDL 参考手册 技术报告, 1997)
- 7 Frakes W B, Pole T P. An empirical study of representation methods for reusable software components. *IEEE Trans Software Engineering*, 1994, 20(8): 617- 630
- 8 Jeng J J, Cheng B H C. Specification matching for software reuse: A foundation. In: Proc the 17th International Conference on Software Engineering on Symposium on Software Reusability, Seattle, Washington, 1995. 97- 105
- 9 Zaremski A M, Wing J M. Signature matching: A tool for using software libraries. *ACM Trans Software Engineering and Methodology*, 1995, 4(2): 146- 170
- 10 Xu Zhen-Quan. The Methodology and Technology of Software Reuse. Wuhan: Huazhong University of Science & Technology Press, 1998 (in Chinese)
(徐正权 软件复用方法与技术 武汉: 华中理工大学出版社, 1998)
- 11 Podgurski A, Pierce L. Retrieving reusable software by sampling behavior. *ACM Trans Software Engineering and Methodology*, 1993, 2(3): 286- 303
- 12 Eduardo O, James H, Prieto D R, Christine B. Computing similarity in a reuse library system: an AF-based approach. *ACM Trans Software Engineering and Methodology*, 1992, 1(3): 205- 228
- 13 Reuse Library Interoperability Group. Uniform Data Model for Reuse Libraries (UDM). RIG: Technical Report: 0002, 1994
- 14 Jade Bird III Project Team of Peking University. Jade bird component models. Technical Report, 1997 (in Chinese)
(北京大学青鸟III项目组 青鸟构件模型 技术报告, 1997)



ZHANG Yong, male, born in 1973. He is now a Ph.D. candidate of department of computer science, Fudan University. His research interests include software reuse, software architecture and software testing.

WANG Yuan-Fen, male, born in 1974. He is now a Ph.D. candidate at department of computer science, Fudan University. His research interests include software reuse, software component repository.

QIAN Le-Qin, male, born in 1942. He is now a professor of department of computer science, Fudan University. His research interests include software reuse, software architecture, software testing and software automatization.