基于语义的构件描述与检索的研究*

边小凡[1]，惠宝山[2]，王 燕[3]
（1. 河北大学 计算中心，河北 保定 071002；
2. 河北大学 数学与计算机学院，河北 保定 071002；
3. 中国石油大学 计算机系，北京 102249）

摘 要： 根据构件检索技术的研究现状，结合领域本体，对现有的构件描述模型进行了改进，提出了基于语义的构件检索模型及相应的概念语义匹配算法。通过试验分析，此算法提高了构件的查全率和查准率。

关键词： 领域本体 构件描述模型 语义相似度 CBSD

近年来，随着软件复用和构件技术的发展，基于构件的软件开发 CBSD[1] (Component Based Software Development) 被认为是有效提高软件生产率、缩短软件产品交付时间和提高软件质量的新途径。但是在实际的开发中，软件的复用程度并不理想。随着软件复用实践的深入和软件构件库规模的扩大，如何提高构件的查全率、查准率已经成为一个关键的技术问题。本文结合领域本体的特点，对以前的构件描述模型进行了改进，用以支持基于语义的构件检索。本文提出的构件检索模型，目的是找出一种构件需求与构件实体间的检索匹配算法，最终获得理想的构件。

* 基金项目：河北省科技攻关计划基金资助项目（021124059）

1 关于构件描述和检索的研究现状

当前的构件描述和检索技术起源于几个领域，目前比较有代表性的方法包括：传统的信息科学编目查询技术和基于框架、基于演绎和基于刻面的构件描述与检索方法。

(1)传统的信息科学编目查询技术

早期比较具有代表性的是使用基于关键字匹配[2]的传统的图书馆及信息科学编目信息查询技术，这种检索方法主要是计算查询关键字与构件描述关键字之间的匹配程度，在应用中也比较容易实现。但基于关键字匹

配的查询无法体现出所查询的关键字之间的逻辑联系，使得构件的查准率受到限制。

(2)基于框架(Frame-Based)的构件描述与检索

在软件代理和分布式计算领域提出的基于框架的构件描述和检索方法中，要求所有的构件和检索查询都用相同的预先定义好的词汇来描述，即使用“属性-值对”(attribute-valuepairs)来对构件进行分类描述。目前大部分商业化的构件服务搜索技术（例如 Jini、eSpeak、Salutaion、UDDI)[3]都使用基于框架的方法。这种方法对于构件描述和检索查询都是使用相同描述的术语来提高查全率和查准率，但它又是以要求所有的构件服务都用框架进行建模为代价的，其灵活性受到制约，在实际应用中比较难以推广。

(3)基于演绎的构件描述与检索

Fischer B.[4] 等人提出的基于演绎的检索方法将基于框架的方法往前推进了一步。该方法首先使用逻辑方法形式化地说明构件服务的属性(例如输入、输出、功能(前置、后置条件、不变量)、性能等)，然后通过证明某个构件是否实现了检索查询所描述的服务属性来进行构件检索。这种方法要求预先定义的逻辑谓词不能存在冗余，并且对所有的构件服务和检索查询都进行完全形式化的规约，才能获得比较理想的查全率和查准率。

(4)基于刻面的构件描述与检索

基于刻面[5]的描述是一种目前正逐步得到重视和应用的描述方法。该方法能够从多个角度、多个方面对构件作出更为全面的描述，在应用中取得了良好的效果。同时，该方法对于术语空间较为稳定的刻面(如使用环境、应用领域等)易描述、易检索，而对于构件的服务和功能等刻面，由于未定义形式化的描述方式，使得其描述内容往往过于自然语言化，从而导致精确性下降，而这样的刻面又正是最具重要性的。而且，传统的刻面描述方法重视构件的静态特征描述而没有提供对构件动态行为和服务的描述机制，因此，对于检索的查全率和查准率都会带来不可忽视的影响。

2 特定领域的本体建模

2.1 领域本体

领域本体是指在一个特定的领域中可重用的本体，它提供该领域特定的概念定义和概念之间的关系，提供该领域中发生的活动以及该领域的主要理论和基本原理。领域本体主要研究与一个特定领域相关的术语或词汇，如医学、企业模拟等。

2.2 构建领域本体

领域本体的构建必须对特定领域的业务知识有着详尽的理解，从而进行概念的获取。目前，尽管存在很多本体构建方法，如 METHONTOLOGY 方法、斯坦福大学的七步法等，但是本体的构建还没有形成一套规范性的指南，只是在其研究环境下能很好地发挥作用就可以。本文的目的主要是满足构件的检索，综合现有的本体构建方法，对已有的构件领域本体方法进行简化，其步骤包括：

(1)建立概念和概念间的关系。

(2)定义概念的属性。

(3)使用概念图和 OWL 语言描述领域业务本体。

利用概念图对领域本体进行建模，使用 OWL 的类、类属性、类公理等对领域本体进行描述。这一过程一般采用本体编辑工具(如比较著名的 Protege)，在图形化的用户界面下对本体进行构建，然后自动生成 OWL 语言的描述。本体的构建是一个反复叠加的过程，必须不断地对领域业务本体进行维护和完善。

3 改进的构件描述模型

根据构件描述和检索的研究现状可以了解到，现有的构件描述方法对构件的静态特征作了较为详尽的考虑，也获得了较好的解决方案。但对于构件的动态特征，则没有提供对构件行为和服务的准确描述，或者进行了尝试但还没有提出有效的方法，而这正是影响检索质量(查全率、查准率、性能)的重要原因之一。由于软件构件本身的特殊性和功能的复杂性，需要提出一种改进的构件描述方法。这种构件描述方法能够在较高层次上理解构件所提供服务的语义、能有效地支持构件检索、适配和验证。为了解决这个问题，在语义级别上应该刻画三部分内容：接口功能的语义、接口中某个操作的语义及操作所期望得到的结果和一个操作的执行所导致的接口中某部分状态的变化。改进的构件描述方法如下：

定义 构件(Component)：构件是指语义完整、语法正确和有可复用价值的单位软件，从组成上看，它是构件基本信息(C_BaseInformation)、构件语义(C_Semantic)、构件接口(C_Interface)、构件实体(C_ Entity)的复合体。

构件模型：:=<构件基本信息，构件语义，构件接口，构件实体 >

Component：:=<C_BaseInformation,C_Semantic,C_Interface, C_ Entity >

定义 1 构件基本信息(C_ BaseInformation)：主要是指构件的静态信息。它由构件标识、构件名称、构件作者、构件功能、构件版本、构件大小、构件发布日期组成。

C_BaseInformation：:=<C_ID,C_Name,C_Author, C_Function,C_Version, C_Size, C_Date>

定义 2 构件语义(C_Semantic)：是指构件的含义、特点和使用方法，是构件可复用价值的关键因素。由领域特征(Domain_Feature)和构件关系(C_Relationship)组成。

C_Semantic：:=<Domain_Feature,C_Relationship>

定义 2.1 领域特征 (Domain_Feature)：是领域空间子集的集合，每个集合的元素是多个特征的逻辑集合。由领域特性(Domain_Trait)、应用技术(Application_Technology)、操作环境(Operation_Setting)组成。

Domain_feature：:=< Domain_Trait, Application_Tech-

nology, Operation_Setting>

定义 2.2 构件关系(C_Relationship):主要包括版本关系 (Version_Relationship)、协作关系(Cooperation_Relationship)、精化关系(Refinement_Relationship)、包含关系(Inclusion_Relationship)、依赖关系(Dependence_Relationship)。

C_Relationship::=<Version_Relationship, Cooperation_Relationship, Refinement_Relationship, Inclusion_Relationship, Dependence_Relationship>

说明:

版本关系:指构件演化中出现的多个版本的构件之间的关系。

协作关系:与某构件相互合作、共同完成一项任务的各构件之间的关系。

精化关系:软件生命周期相邻阶段的构件之间的关系。

包含关系:指形态不同的构件之间的包含关系。

依赖关系:一种使用关系,指一个构件的变化会影响到其他构件的使用。

定义 3 构件接口:构件接口是构件与其他构件或外部环境交互的媒介,它包括服务请求接口和服务提供接口。主要由接口名称(Interface_Name)、接口函数(Interface_Mothod)、接口语义(Interface_Semantic)组成。

C_Interface::=< Interface_Name, Interface_Mothod, Interface_Semantic >

定义 3.1 接口函数 (Interface_Mothod): 由函数名(Mothod_Name)、函数功能(Mothod_Funtion)、函数参数(Mothod_Parameter)组成。

Interface_Mothod::=<Mothod_Name, Mothod_Funtion, Mothod_Parameter>

定义 3.2 接口语义(Interface_Semantic):它可分为接口功能(Interface_Function)、接口操作(Interface_Option)、接口参数(Interface_ Parameter)三部分描述。

Interface_Semantic::=<Interface_Function, Interface_Option, Interface_ Parameter >

说明:

接口功能的描述对应领域本体中的业务功能。

接口操作的描述对应领域本体中的动作概念。

接口参数的描述与领域本体中信息资源相对应。

定义 4 构件实体是满足接口规范和语义描述的实例。构件实体在构件模型中具体表示为:

构件实体::=<索引,实现体>

C_Entity=<Index, Implementation_Body>

4 构件检索模型及匹配算法

构件的检索与匹配是实现软件复用的关键技术之一。随着构件库构件数量的不断增加,如何在构件库中找到合适的构件是一个难点。本文提出了这样的检索过程:首先用户使用自然语言确定需要的构件需求,根据需求得出用户初始查询,然后利用领域本体中的领域知识来检索上下文,对初始查询进行服务求精,这一检索过程如图 1 所示。

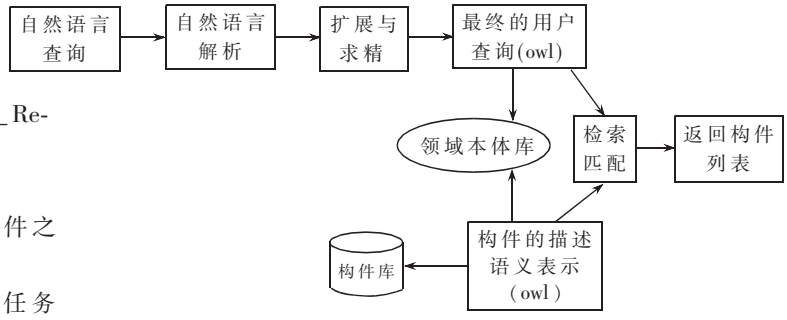

图 1 基于语义的构件检索模型

检索方法主要包含以下步骤:产生初始查询;查询扩充和求精;检索构件。具体描述如下:

(1)产生初始查询:用户使用自然语言确定检索需求,通过自然语言解析产生用户初始查询,即把用户查询转换为概念图,由 OWL 语言表示。该步骤涉及到自然语言理解和自然语言到 OWL 语言的转换等技术。此功能模块借鉴了基于实例的学习方法[6],首先建立了一个实例库, 存储查询语句及其相应的被转换的 OWL 实例的集合,通过在实例库中查找相似的查询语句,新的查询语句能够被转换为相似的 OWL 实例。此实例库初期相对较小,自然语句解析的好坏与实例库的规模大小有直接的联系,需要不断的完善实例库。具体内容可参见参考文献[4]。

(2)查询扩充和求精:把初始查询概念图中用到的和业务功能相关的关键字和概念与领域本体中的概念进行语义匹配,语义相似度满足一定域值的概念被作为相关的概念,显示给用户,用来扩充查询。下面给出了概念间的语义相似度算法。

定义 1 概念间的语义距离 Distance($C1$、$C2$),表示两个概念在语义树最短路径上 n 条边权值的总和。用公式表示为:

$$\text{Distance}\ (C1,C2)=\sum_{i=1}^{n} Weight_i$$。式中,$Weight_i$ 是连接 $C1$、$C2$ 的最短路径上第 i 条边的权值。

由于领域本体是以倒立的树结构存储的,从主观判断看,处于层次树中离根较远的概念间的相似度要比离根近的概念间相似度大些。因此概念在树中所处的深度是另一个需要考虑的因素,处于树中不同深度的边应该赋给不同的权值。

定义 2 概念 C 在树中的深度 $D\text{epth}(C)$:指概念与树根的最短路径所包含的边数 n。

$D\text{epth}(C)=n$,n 是该最短路径包括的边数。

定义 3 概念间的宽度 $Width(C)$:指同一深度下该节

点的数目。

定义4 概念C的权值 $Weight(C)$：由于从概念C引出的边具有相等的权值，本文规定 $Weight(C)$ 是指从概念C引出的边的权值，它与概念的深度及相同深度概念间的宽度成反比关系。用公式表示为：

$$Weight(C)=\frac{1}{Width(C)}\times\frac{1}{Depth(C)+1}$$

根据上述定义，可得语义相似度与语义距离的转换公式：$\mathrm{Sim}(C1,C2)=1-\sqrt[\alpha]{Distance(C1,C2)}$。式中，参数 $\alpha\in(0,1)$，经过多次试验可以确定 $\mathrm{Sim}(C1,C2)\in(0,1)$。

(3)检索构件：在查询扩充和求精的过程中，通过语义匹配对用户查询进行相关业务功能的扩充，得到了由OWL语言表示的构件最终查询条件，把最终查询条件区分为功能需求和非功能需求（如语言、运行环境等）。利用功能需求对接口列表进行检索。接口支持功能需求的百分比表示为行为相关度，把行为相关度满足一定域值的接口集合按其相关度排序返回给用户。通过接口可以映射到相关的构件，最终获得所需求的理想构件。其匹配算法如下：

输入：语义扩展后的基准本体：$\mathrm{Ontology}_1$，待评估本体：$\mathrm{Ontology}_2$。

输出：结果向量$(R_1,R_2,\ldots,R_m)$。

解析 $\mathrm{Ontology}_1$ 生成概念词汇组$(P_1,P_2,\cdots,P_m)$，解析 $\mathrm{Ontology}_2$ 生成概念词汇组$(C_1,C_2,\ldots,C_n)$。

根据映射规则生成概念权向量 $(t_1,t_2,\ldots,t_m)$ 和$(r_1,r_2,\ldots,r_n)$。

顺序选择$(P_1,P_2,..,P_m)$中的每一个词汇 P_i，比较 P_i 和$(C_1,C_2,\ldots,C_n)$中词汇 C_j，如果 P_i 等于 C_j，则 $R_i=t_i\times r_j$，否则 $R_i=0$，选择下一个 P_i+1。

输出结果向量$(R_1,R_2,\ldots,R_m)$。

结果向量$(R_1,R_2,\ldots,R_m)$中分量 R_i 表示待评估本体某一概念对基准 $\mathrm{Ontology}_1$ 中第 i 个概念的语义影响。结果向量各分量的和值 $\sum_{i=1}^{m}R_i$ 可以反映待评估本体 $\mathrm{Ontology}_2$ 同基准本体 $\mathrm{Ontology}_1$ 的语义相似度，在此作为语义相似度因子值。

利用上述算法，用户能计算出每个接口某方面的概念及与它的语义相关度，进行排序后供用户选择，然后通过接口和构件实体间的映射，找到理想的构件。

5 实验与结果分析

为了测试该检索方法的可行性，作者通过基于刻面的构件检索和基于本体的语义构件检索两种检索方法对现有的网上信息收集构件库进行了测试。该构件库共包含构件152个，测试人员分别从查准率和查全率两个方面进行了验证。

定义1：查准率：指检索到的符合条件的构件与检索到的全部构件的比率。

定义2：查全率：指检索到的符合条件的构件与构件库中实际符合条件的构件数量的比率。

两种检索方法的验证结果如表1所示。通过比较这两种检索方法，在查全率和查准率方面，基于本体的语义构件检索方法明显优于基于刻面的构件检索方法。以上测试验证了本检索方法的可行性和有效性。

表1 两种检索方法验证结果

领域构件库	检索方法	功能	相关构件(个)	多余构件(个)	错过构件(个)	查全率(%)	查准率(%)
网上信息收集构件库	基于刻面的构件检索	问卷管理	6	2	1	85.7	75
		模版管理	8	1	2	80	88.9
		信息查询	8	2	2	80	80
		评估统计	7	1	1	87.5	87.5
	基于本体的语义构件检索	问卷管理	7	1	0	100	87.5
		模版管理	9	1	1	90	90
		信息查询	9	0	1	90	100
		评估统计	7	1	1	87.5	87.5

随着软件复用实践的深入和软件构件库规模的扩大，迫切要求更精确、更完善的构件检索方法。针对这一现状，本文结合领域本体，提出了基于语义的构件检索模型，通过用户需求和构件实体间的匹配算法，提高了构件的查准率和查全率。然而，领域本体库的创建是一个长期而艰巨的工作，需要不断的扩充和完善，才能更好地提高构件的检索质量。

参考文献

[1] 梅 宏，陈 锋.ABC:基于体系结构、面向构件的软件开发方法[J].软件学报，2003，14(4):721-732.

[2] 王乐乐.基于特征模式提取的教育信息构件检索.吉林师范大学学报，2002，(3):52-54.

[3] 林清滢.基于UDDI的语义Web服务发现研究.计算机工程与设计，2006，27(12):2215-2217.

[4] FISCHER B, WHITTLE J. An integration of deductive retrieval into deductive synthesis[J]. Proceedings of the 14th IEEE International Conferenceon Automated Software Engineering. October 1999.

[5] 王渊峰，张涌，任洪敏，等. 基于刻面描述的构件检索.软件学报，2002，13(08):1546-1551.

[6] VON Wun Soo，CHEN Yu Lee. Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques[J]. IEEE, 2003.

(收稿日期：2006-11-11)