

# 基于本体的构件描述和检索

彭鑫<sup>1</sup>, 赵文耘<sup>1</sup>, 肖君<sup>2</sup>

<sup>1</sup> 复旦大学计算机科学与工程系软件工程实验室, <sup>2</sup> 上海远程教育集团.

**摘要:** 构件的描述和检索是软件复用和构件库研究的重点.在基于刻面的描述方法基础上引入本体作为用户复用需求和构件描述的公共基础,从而使二者能够在语义层面上进行匹配.为了支持构件领域特性的描述,还引入了上层本体和领域本体分别作为构件公共特性和领域特性描述和匹配的知识基础.语义基础的引入使得构件的复用机会大大提高.

**关键字:** 构件; 描述; 检索; 本体; 刻面; 匹配

可复用构件的描述和检索是软件复用和构件库研究的一个重点[1].不同的构件检索方法是以相应的构件描述方案为基础的,例如枚举法、关键字法、属性-值法,基于构件规约的语法匹配[2],以及基于刻面描述的查询检索[1][3][4].其中基于刻面的构件描述和检索是目前应用最为广泛的一种方法,例如青鸟构件库[5]采用的就是以刻面分类为主多种分类模式相结合的构件描述方案.

复用是一个在复用需求与产品描述之间进行匹配的过程[6].在基于刻面的描述和检索方法中,用户的检索请求被表示成一棵查询树,从而使构件检索转化为查询树与库中构件的刻面描述树之间的匹配问题[1].刻面分类方法从若干不同的维度描述复杂对象[5],提高了描述的精确性,并具有较高的检索效率,但仍然存在着一一定的局限性.需求或产品描述的不完全是构件复用的主要问题[6].在基于刻面的构件检索中,用户查询树代表了用户的复用需求,但在很多情况下都是不完整、不精确的.这一方面是由于用户自身复用需求的不明确,另一方面也是由于提供给用户的查询请求表述手段有限.从构件描述的角度看,刻面方案能够较为全面地描述一个构件,但也存在一些局限性,例如术语间主要是一般特殊关系且无法表达术语间的交叉分类.由此可见,单纯使用基于刻面的描述方案无法让用户充分表述自己的查询请求,同时构件的描述信息也不够完整.

近年来,本体论作为共享知识的表达基础已经被广泛应用于信息科学中,例如软件复用[6]、信息检索、需求获取[7]等.领域本体为领域内的概念以及概念间广泛存在的各种关系提供了共享的描述,因此可以作为领域内构件描述的知识基础.基于领域本体一方面可以更加准确、完整地描述检索要求,另一方面可以为构件的刻面描述提供丰富的语义注解,从而更好地弥合用户复用需求与构件描述之间的“鸿沟”.

本文在基于刻面的描述和检索方法基础上引入本体作为检索请求和构件描述的知识基础,提出了一种基于本体的构件描述和检索方法.本文将首先介绍基于刻面的构件描述和检索方法及其局限性,然后介绍基于本体的构件描述以及检索过程.

## 1. 基于刻面的构件描述检索

基于刻面的描述方案主要由三部分组成[1]: 刻面分类方案、各个构件的刻面描述集合以及刻面描述术语之间的关系,即术语辞典.图 1 是一个构件描述刻面树及术语空间的部分示例,其中方框代表刻面,椭圆代表术语,刻面之间按照组成关系(文献[8]将其定义为 Aspect-of 关系)构成一棵树,而术语按照一般特殊关系构成一棵树.有的刻面方案只包含刻面树,术语直接作为刻面树的叶子节点,例如[1][3][4].而在青鸟构件库[5]中,每一个刻面下的术语按一般特

殊关系构成一棵树.我们认为刻面和术语二者都需要建立层次结构.剖面可以按照观点或特定领域的维度进行组织[9],分层的剖面方案使各剖面的含义更加容易理解,并且使构件描述的维度更为丰富.按一般特殊关系构成的术语空间为构件在某一剖面下的描述提供了多种抽象层次上的选择.因此,无论剖面方案还是术语空间都应该支持层次构造.

基于剖面的方法相比传统的基于关键字的方法有了很大进步,但也存在着一定的局限性,主要体现在:1)剖面树和术语空间的表达能力有限,例如剖面之间主要是组成关系,而术语之间主要是一般特殊关系和同义关系;2)术语空间往往只能体现一种概念分层模式,限制了构件描述的灵活性;3)匹配的灵活性不够;4)剖面及术语方案往往只能满足构件高层描述的需要,无法体现特定领域的描述内容.

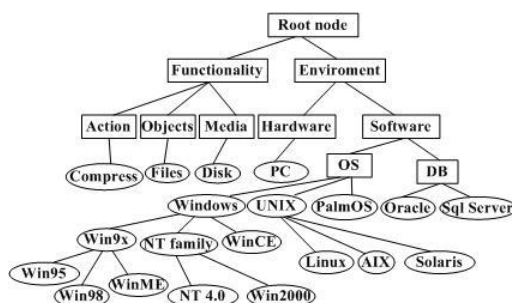


图1 构件描述剖面树及术语空间部分示例

Fig.1 Example of facet tree and term space

## 2. 基于本体的构件描述

文献[6]认为用户的复用需求也隐含着一种基于个人认识的本体,而揭示构件描述和复用需求的真实含义的过程是一个本体协商的过程.由此可见,构件的描述和检索需要一个规范而全面的概念空间作为基础.由于本体在语义层面上详尽描述了领域内的概念以及概念间的关系,因此可以成为用户复用需求表述以及构件描述的公共知识基础,有利于解决上面提到的三个问题.OWL[10]是 W3C 推荐的 Web 本体语言,本文中的本体将主要使用 OWL 来描述.

### 2.1 基于本体的刻面和术语定义

从知识管理的角度看,本体是指某些领域的共享理解,作用在于支持逻辑推理以及知识的共享、复用[11].领域模型与本体紧密相关,文献[9]认为领域模型可以看作一个范围较窄或特定的本体,而二者之间的主要区别在于领域模型的非形式化并且不包含公理.剖面和术语空间同样具有一定的本体特征,每一个术语空间都是从一个描述维度出发的层次概念树.

引入本体后,构件描述仍然以剖面方案为主,不同的是领域本体将作为构件描述和检索的知识基础存在.这些知识将有助于揭示构件复用需求以及构件描述的真实含义,提高构件复用的机会.我们认为领域知识主要体现在叶子剖面下的术语空间上,而剖面树主要体现一系列正交的描述方面的组成关系.我们将叶子剖面和术语定义为本体中的概念,而叶子剖面、术语以及其它概念之间的关系也将在本体中定义.

OWL 中的概念主要由类 (Class)、实例 (Instance) 和关系 (Property) 组成.叶子剖面在本体中定义为类,而术语中的抽象术语和具体术语分别定义为类和实例.同一叶子剖面下定义为类的术语构成一系列继承关系,相应的叶子剖面类是它们共同的父类.定义为实例的术语是上层术语类的实例.术语中类与实例的区分与用户在本体建模时的观点有关,例如图 1 中如果不区分 Oracle 的具体版本那么术语“Oracle”是类“DB”的实例,否则是“DB”的子类.由于本体中允许多继承,因此每个剖面下的取值不再局限于某种分类模式下的术语树.

除了继承关系,本体中还可以定义其它关系,例如类“Action”和“Objects”之间的“Act on”关系,这是一种剖面间依赖关系,或者类“Windows”之间的先后关系,这些关系能够为用户检索请求和构件描述提供附加语义.我们在本体将剖面间依赖关系定义为“FacetDependenceProperty”,具体的依赖关系均从它继承,如“Act on”关系.

## 2.2 面向特定领域的扩展

一个通用的剖面描述方案必然导致领域特征描述能力的不足.为了更加准确地对特定领域构件进行描述,应该允许描述方案以及本体向特定领域扩展.文献[11]在环境建模中引入上层本体和特定领域本体的分层结构,分别描述一般的环境建模概念以及特定领域中的扩展.这种扩展是以本体中的继承方式进行的.与此相似,可以通过以下两种方式扩展本体和剖面描述方案:

- 1) 扩展特定领域的术语.这种扩展只能在叶子剖面或定义为类的上层术语下进行,在领域本体中体现为这些领域术语对于上层本体中类的继承或实例关系.
- 2) 其它知识的扩展.除了领域术语之外,还可以在领域本体中定义新的类、关系和实例.

## 2.3 基于本体的剖面和术语实例

图2是一个财务领域的剖面和本体片段,图中左侧是剖面树的一部分,右侧分别描述了上层本体和财务领域本体的一个片段.从中可以看出类之间的关系可以在同一叶子剖面的术语之间定义,也可以在不同叶子剖面下的术语之间定义.前者体现了同类术语之间的某种关联,例如图2中“OS”实例之间的“following”关系,后者往往体现了不同剖面间的依赖性,例如“Act on”关系.本体中的知识一部分体现为实例知识,另一部分体现为公理.类之间的子类关系就是一种公理.除此之外还可以定义其它公理,例如将关系定义为传递关系.这些知识都可以用来辅助用户的构件检索过程.图3是图2中上层本体 OWL 描述的片段.

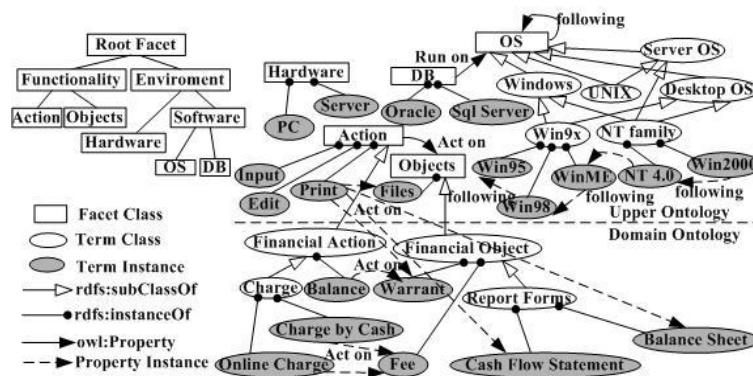


图2 基于本体的剖面和术语定义

Fig.2 Ontology-based definitions of facets and terms

```

<owl:Class rdf:ID="Action"/>
<owl:Class rdf:ID="Objects"/>
<owl:Class rdf:ID="OS"/>
<owl:Class rdf:ID="ServerOS">
  <rdfs:subClassOf rdf:resource="#OS"/>
</owl:Class>
<owl:Class rdf:ID="DesktopOS">
  <rdfs:subClassOf rdf:resource="#OS"/>
</owl:Class>
<owl:Class rdf:ID="Windows">
  <rdfs:subClassOf rdf:resource="#OS"/>
</owl:Class>
<owl:Class rdf:ID="NTfamily">
  <rdfs:subClassOf
rdf:resource="#Windows"/>
  <rdfs:subClassOf
rdf:resource="#ServerOS"/>
  <rdfs:subClassOf
rdf:resource="#DesktopOS"/>
</owl:Class>
<owl:TransitiveProperty
rdf:ID="following">
  <rdfs:domain rdf:resource="#OS"/>
  <rdfs:range rdf:resource="#OS"/>
</owl:TransitiveProperty>
<owl:ObjectProperty
rdf:ID="FacetDependenceProperty"/>
<owl:ObjectProperty rdf:ID="ActOn">
  <rdfs:subPropertyOf
rdf:resource="#FacetDependenceProperty"/>
  <rdfs:domain rdf:resource="#Action"/>
  <rdfs:range rdf:resource="#Objects"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="previous">
  <owl:inverseOf
rdf:resource="#following">
</owl:ObjectProperty>
<Win9x rdf:ID="Win95"/>

```

图3 上层本体的 OWL 描述片断  
Fig.3 OWL fragment of the upper ontology

### 3. 基于本体的构件检索

#### 3.1 检索过程

基于本体的构件检索主要包括以下步骤（图4）：

- 1) 用户检索请求描述及处理：用户以短句的形式表达检索要求,系统首先将词汇映射到基于本体的概念体系中,例如文献[8]通过概念词典完成这种映射.
- 2) 查询树生成及修改：根据本体知识将检索要求映射到查询树上.如果查询树存在不一致则需要提示用户修改查询请求.
- 3) 刻面树匹配：根据生成的查询树进行构件匹配,匹配过程中将按照匹配算法计算匹配度,然后依次返回检索结果.

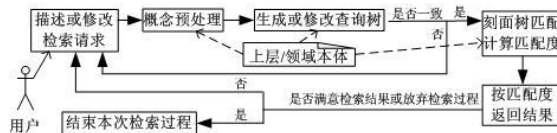


图 4 基于本体的构件检索过程

Fig.4 Ontology-based retrieval process

### 3.2 检索请求处理及查询树生成

使用自然语言表达复用需求对于用户无疑是最方便的,但需要相应的概念预处理过程将用户词汇转换为概念.用户陈述一般由子句组成.这些子句通过句式分析和切分后可以得到词汇流,而词汇可以借助概念词典映射为本体中的概念,包括类、关系和实例.概念词典记录词汇与概念间的对应关系,文献[8]对此有详细介绍.

经过概念预处理后,用户陈述转换为基于概念的规范子句.根据其中概念的组成一般就可以确定目标构件所属的通用或特定领域.这些子句可能包含刻画、术语、其它本体类、实例或关系,每一个有效子句都能确定查询树中某些刻面的取值.根据通用本体或目标领域本体中的知识,可以按照以下步骤生成查询树:

1) 根据子句中的直接表述确定某些刻画下的术语,这种子句一般具有“刻画+术语”或者“术语组合”形式.前者例如“Objects=Warrant”,后者例如“Print Warrant”.

2) 根据子句中的关系表达确定一部分检索要求,一般是同一刻画下的术语关系.例如“following Win98”可以推导出{WinME, NT4.0, Win2000}.图 3 中的本体还定义了“following”的反关系“previous”,因此也可以使用“previous NT4.0”这样的子句.

3) 根据刻画间的依赖关系间接推导出隐含的检索要求.例如“Action”刻画取值“print”,那么在财务领域中依据“Act on”关系,“Objects”刻画取值可以为{Files, Report Forms, Warrant}.

4) 综合各项限制条件对同一刻画下的术语进行取舍.步骤 1-3 都可能对某一个刻画下的术语进行限制,那么需要选择满足各条限制的公共子集.例如“OS”刻画下用户指定了“Server OS”,而根据与其它刻面的依赖关系必须取值“Windows”,那么综合后取值为“NT family”.如果多个检索要求间存在不一致性,则需要列出导致矛盾的那些查询子句并显示导致不一致性的原因,提示用户修改检索要求.

步骤 2 和 3 需要基于本体的推导过程,推导可以基于公理和推导规则.例如“following Win98”的推导过程需要用到“following”是“TransitiveProperty”,即传递关系的公理,以及 OWL 中关于“TransitiveProperty”的推导规则.除此之外,用户还可以在本体中自定义推导规则.

### 3.3 刻画匹配度公式

如果用户的检索请求带有领域性(包含特定领域术语),那么匹配将在具有相同领域属性的构件中进行.查询树中同一刻画下各术语间是并的关系,而多个刻画之间是交的关系[5].刻画匹配以叶子刻画为单位进行,对于查询树  $Q$  以及构件  $C$ ,匹配度  $M$  的计算公式为:

$$M = \sum_{i=1}^n \alpha_i m(Q_{t_i}, C_{t_i}), \text{ 其中 } n \text{ 表示叶子刻画数, } \alpha_i \text{ 表示第 } i \text{ 个刻面的权重, } Q_{t_i} \text{ 和 } C_{t_i} \text{ 分}$$

别表示  $Q$  和  $C$  在第  $i$  个叶子刻画上的术语集合,  $m(Q_{t_i}, C_{t_i})$  代表它们的匹配度.

$$S_q = \{ins \mid ins \in termIns \wedge \exists t \in Q_{t_i} : ins = t \vee ins \pi t\},$$

$$S_c = \{ins \mid ins \in termIns \wedge \exists t \in C_{t_i} : ins = t \vee ins \pi t\},$$

其中  $termIns$  表示本体中的实例术语集合,  $\pi$  表示  $instanceOf$  关系,

于是,  $m(Q_t, C_t)$  可以定义为  $m(Q_t, C_t) = \frac{|S_q \cap S_c|}{|S_q|}$ .

叶子刻面的权重体现了用户的复用需求中各方面因素的相对重要性,其取值与构件类型有关.例如功能性构件的功能、运行环境等刻面权重较高,资源类构件与内容相关的刻面权重较高.而 Q 与 C 在每个叶子刻面上的匹配度 m 则取决于本体中该刻面下二者公共实例术语数量的比例.这种计算方法较适用于本体中术语分类方案较为匀称的情况下,这对于一个构造良好的本体模型而言是可以满足的.如果术语分类方案很不均匀,例如,如果图 2 中“UNIX”术语作为实例存在,则与“Windows”术语下的分类层次有较大差异,不适合用该公式计算刻面匹配度.此时需要引入术语层次作为一个计算因子进一步细化刻面匹配度公式,但复杂度也将大大增加.

#### 4. 总结和展望

基于刻面的构件描述和检索方法的一个重要不足是概念表达的不充分性.本文在基于刻面方法的基础上引入本体作为构件描述和用户复用需求表述的公共知识基础,从而进一步弥合二者之间的表达差异.这种方法的匹配过程是基于本体知识的概念匹配度计算,因此能够在很大程度上实现模糊匹配.另一方面,用户不需要基于特定的刻面方案表达检索要求,查询树可以在领域本体的支持下从用户的查询表述中得出.由此可见,只要共享相同的本体知识,用户的检索请求就能被采用不同刻面分类方案的构件库所理解,因此还特别适用于分布式异构构件库环境下的构件检索.进一步的研究工作主要包括以下两个方面:

1) 引入文本和图像检索中相关反馈技术[12],建立基于反馈的交互式构件检索方法.引入交互能力的主要目的是借助用户反馈以及本体知识,通过交互不断地校正并挖掘用户的检索要求,从而使构件检索的结果更加符合用户的真实需求.

2) 引入更加形式化的逻辑语言,例如与 OWL DL 描述能力相当的描述逻辑,作为本体的描述手段,从而使推理过程自动化.

#### 参考文献:

- [1] WANG Yuan-feng, ZHANG Yong, REN Hong-min, et al. Retrieving Components Based on Faceted Classification. *Journal of Software*, 2002, 13(8): 1546~1551. (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(8): 1546~1551).
- [2] MA Liang, SUN Jia-su. Component Retrieval Based on Specification Matching. *MINI-MICRO SYSTEM*, 2002, 23(10): 1153~1157. (马亮, 孙家骅. 基于规约匹配的构件检索. *小型微型计算机系统*, 2002, 23(10): 1153~1157).
- [3] WANG Yuan-Feng, XUE Yun-Jiao, ZHANG Yong, et al. A Matching Model for Software Component Classified in Faceted Scheme. *Journal of Software*, 2003, 14(3): 401~408. (王渊峰, 薛云皎, 张涌, 等. 刻面分类构件的匹配模型. *软件学报*, 2003, 14(3): 401~408).
- [4] JIA Xiao-Hui, CHEN De-Hua, YAN Mei, et al. Research on Matching Model and Algorithm for Faceted-Based Software Component Query. *JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT*, 2004, 41(10): 1634~1638. (贾晓辉, 陈德华, 严梅, 等. 基于刻面描述的构件查询匹配模型及算法研究. *计算机研究与发展*, 2004, 41(10): 1634~1638).
- [5] CHANG Ji-chuan, LI Ke-qin, GUO Li-feng, et al. Representing and Retrieving Reusable Software Components in JB (Jadebird) System. *ACTA ELECTRONICA SINICA*, 2000, 28(8): 20~23. (常继传, 李克勤, 郭立峰, 等. 青鸟系统中可复用软件构件的表示与查询. *电子学报*, 2000, 28(8): 20~23).

- [6] Sidney C. Bailin. Software Reuse as Ontology Negotiation. Proceedings of the 8th International Conference on Software Reuse (ICSR 2004). 2004: 242~253.
- [7] JIN Zhi. Ontology-Based Requirements Elicitation. Chinese Journal of Computers, 2000, 23(5): 486~492. (金芝. 基于本体的需求自动获取. 计算机学报, 2000, 23(5): 486~492) .
- [8] Li Zhendong, Fei Xianglin. Research on the Concept-based Information Retrieval Model. Journal of Nanjing University(Natural Sciences), 2002, 38(1): 99~109. (李振东, 费翔林. 基于概念的信息检索模型研究. 南京大学学报 (自然科学), 2002, 38(1): 99~109) .
- [9] Prieto-Diaz, R.. A faceted approach to building ontologies. Proceedings of IEEE International Conference on Information Reuse and Integration (IRI 2003). 2003: 458~465.
- [10] Sean Bechhofer, et al. Owl Web Ontology Language Reference”, <http://www.w3.org/TR/owl-ref/>, 2004-02-10.
- [11] Xiao Hang Wang, Da Qing Zhang, Tao Gu, et al. Ontology based context modeling and reasoning using OWL. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW 2004). 2004: 18~22.
- [12] Tan Xiao-Yang, Sun Zheng-Xing, Zhang Fu-Yan. Relevance Feedback in Content-based Image Retrieval: the State of the Art. Journal of Nanjing University(Natural Sciences), 2004, 40(5): 211~218. (谭晓阳, 孙正兴, 张福炎. 交互式图像检索中的相关反馈技术研究进展. 南京大学学报 (自然科学), 2004, 40(5): 639~648) .