基于功能语义的构件描述和检索研究①

Component Description and Retrieval Based on Functional Semantics

范　菁　杨　冰　熊丽荣　(浙江工业大学 软件学院 浙江 杭州 310023)

摘　要： 现有的构件描述和检索技术大多没考虑语义描述能力，其查准率和查全率往往无法令人满意。因此，如何通过引入本体来弥补这种语义缺失成为新的研究热点。本文在刻面描述的基础上引入本体技术，提出利用功能刻面的分类模式建立构件本体的方法，并讨论了该本体对构件语义检索的支持。

关键词： 刻面 关本体 功能语义

随着软件开发规模的增大，以软件构件化为基础的共享和复用技术越来越得到业界的关注，软件构件的描述方法已经成为海量构件检索的研究重点。现有的构件描述和检索技术中比较有代表性的包括信息科学编目查询技术、基于框架、基于演绎和基于刻面的构件描述与检索方法，其中刻面分类以其较强的描述能力和高效的检索效率成为构件描述的主流方法。但这些方法都没有考虑语义描述能力，在处理查询请求时只能进行机械的符号匹配而不能进行相关语义信息的关联和扩展，构件描述和检索的灵活性受到限制。

通过引入本体来弥补现有构件描述和检索技术中的语义缺失已成为新的研究热点，例如文献[1-3]等都对此做了深入的研究。本文以刻面描述技术为基础，分析已有刻面分类的特点，选择具有丰富语义信息也最为常见的功能刻面作为切入点，将其映射成本体模型，通过把术语空间中蕴含的语义信息转化为本体中的概念和关系，将层次结构的刻面描述树改造成网状结构，增强其描述能力，为构件检索提供语义支持。该方法具有刻面概念正交、层次清晰的优势，同时弥补了刻面描述缺乏语义的不足，能够支持构件的语义检索。

本文内容组织如下：第一部分简要介绍刻面分类方法，分析各类刻面特征，提出将本体引入功能刻面；第二部分以信用构件库的功能刻面为例，具体介绍基于功能语义的构件本体的建立过程，并给出该本体的形式化描述；第三部分讨论了基于功能语义的构件本体对语义检索的支持；最后是结论和展望。

1　基于刻面的构件描述和检索

构件的刻面描述方法[4]的基本思想是抽取构件的某些本质特征属性(刻面)来描述构件。每一个刻面与一个术语空间相关联。术语空间由一组术语构成，是有限的不定空间，可动态地增加和删除术语，任意两个术语空间正交。一般在每个刻面下选择术语形成构件描述树。用户通过选择子刻面或术语作为查询条件，形成查询树。检索过程即树匹配的过程。

本文根据刻面在检索中的作用大小将其分成两类：检索匹配刻面和检索约束刻面。前者描述构件的高级特性，如功能、应用领域等，因此也可以称之为高级特性刻面。这些往往是用户在检索构件过程中最关心的特性，也是检索构件的出发点。后一类型的刻面则描述了构件的基本特征信息，如构件类型、实现语言、应用环境等，也可称之为基本特征刻面。这些信息一般不会成为用户的先决查询条件，而是作为在已有检索结果集中进一步“筛选”符合具体需求的构件的约束条件。

大多数基于刻面分类的构件库系统的刻面可归结为这两大类。高级特性刻面下的术语对构件检索贡献较大，并蕴含丰富的语义信息，基本特征刻面下的术语往往是特征种类的枚举集合，术语间鲜有或没有关

① 基金项目:浙江省科技计划面上项目(2007C21011)
收稿时间:2008-09-18

系。在高级特性刻面中，都有一个刻面是用来描述构件功能的，虽然名称会因系统的具体需求而各不相同，但该刻面的术语空间必然蕴含着丰富的语义信息。本文提出一种基于功能语义的构件本体(下文简称为“构件本体”)，该本体将刻面语义与构件语义结合起来作为构件描述和检索的基础。

2　基于功能语义的构件本体

2.1 本体的定义

“本体是共享的概念模型的明确的形式化的规范说明”[5]。本体由概念类、关系、函数、公理和实例(或个体)五种元素构成，其中概念可形成一个分类层次，并通过关系、函数、公理来表达概念之间或函数之间的关联、约束。因此本体能够明确地描述领域概念的定义，通过概念之间的关系反映概念的语义信息，并为简单的术语赋予明确的背景知识，从而使隐含的关系明晰化，保障语义的一致性。本体能够在语义层次上描述信息，其本身具有良好的概念层次结构，支持逻辑推理，因此特别适合于智能搜索中对概念及其语义的处理。

2.2 基于功能语义的构件本体的建立

2.2.1 概念类及其关系的建立

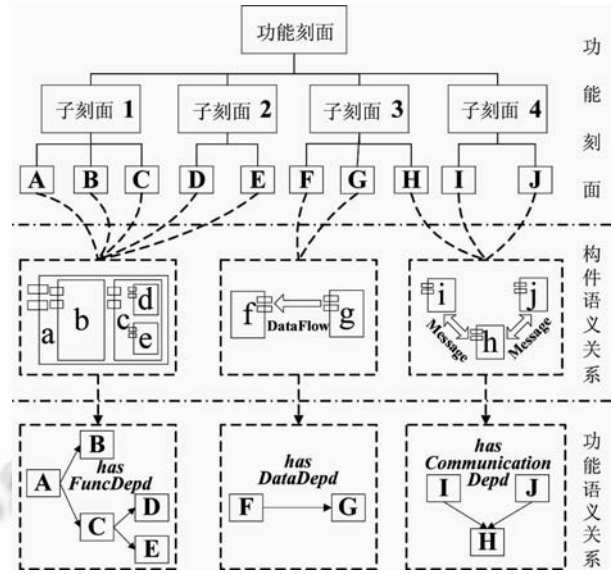
文献[6]提出了一种由刻面辅助建立本体模型的通用方法，但该法既要自顶而下建立本体框架，又要同时自底而上构建刻面分类，最后通过双向修订形成最终的本体。这种方法虽然严谨、精确，但在实际应用中比较耗时耗力。本文对其做了改进，将建立刻面与建立本体的过程相结合，形成本文的本体构建方法。对于已有的功能刻面，也可以按照本文的方法提取语义，建立构件本体。下文将以信用构件库的构件为例，为其建立功能刻面，并将其映射成构件本体。

(1) 功能刻面的建立

功能刻面采用自底向上的建立方法。选择信用领域构件中的典型构件作为建立刻面的样本集合，以构件功能为界限，将样本分解成各种粒度的构件。本文将能作为独立的软件单元提供一定功能，但无法再分解的称为原子构件，可再分解为原子构件的称为复合构件。

图 1 构件语义关系层(a)中所示 a、b、c、d、e 是粒度各不相同的五个构件，其中 a、c 属于复合构件，提供功能 A 和 C；b、d、e 是原子构件，提供 B、D、E 三种功能。(b) (c)所示的是具有各种语义关系的构件 f、g、h、i、j，他们分别具有 F、G、H、I、J 这几个功能。提取描述这些功能的术语作为叶子刻面。然后，将叶子刻面进行分类，形成上层刻面，最终构造出一棵功能刻面树，如图 1 中的功能刻面所示。然后在每个叶子刻面下加入同义词术语，用于在该刻面下描述构件。

图 1 的功能刻面层中的叶子术语 A~J 在刻面分类中的地位和作用完全相同，但在功能语义中却可能处于不同地位，具有不同的依赖关系。我们在提取构件功能特征构建功能刻面的同时，提取构件在功能层面上的依赖关系，将其记为一个二元关系集合，形如 $I=\{Depd|Depd_i(f_m,f_n)\ i,m,n=1,2,3\cdots\ m\neq n\}$，fm 和 fn 表示功能，$Depdi(fm,fn)$表示 f_m 对 f_n 有某种依赖关系 $Depd_i$，I 表示依赖关系集合。



(a) 功能依赖　(b)数据依赖　(c)通讯依赖

图 1　功能刻面的依赖语义

(2) 功能刻面树到本体概念类的映射

本文选用 OWL 中 DL 子语言作为本体描述语言，它是 W3C 推荐的语义网描述语言，具有成熟的语法和广泛的应用。将功能刻面映射为构件本体中的根类，子刻面和叶子刻面作为其子类，他们之间由<rdfs:subClassOf>构成构件本体的概念层次结构。在完成树形结构的映射后将构件本体中根类名称改成 Component，其它概念类的名称都加上后缀“Comp”表示构件类，例如“DataProcess(数据处理)”子刻面，

将改为“DataProcessComp(数据处理构件)”。这就形成了按功能分类的构件本体。

对于同义词的映射有两种方案。一种是将同义词映射为本体中的概念类，用<owl:equivalentClass >表示类间的等价关系。一种是表示成概念类的数据属性，即<owl:DatatypeProperty>，将其值域定为String。在推理中，前者会造成一个实例被推理出同属于多个类的结论，增加事实断言库的信息量，增加推理负担，因此将同义词映射为数据属性hasSynonyms更为合理。

(3) 概念间关系的建立和权重的设置

本体是以关系为中心的，没有关系或者关系很少很弱的本体不能起到应有的作用。文献[7,8]等都对构件关系的做了深入的研究，但都针对具体的构件模型，也没有考虑功能间关系和关系的紧密程度。本文将构件的关系归结为三类：依赖关系、层次关系和相似关系。其中依赖关系表示构件在功能层次上对其他构件的依赖，也可称之为功能关系，是功能刻面中蕴含但因无法表达而遗失的语义信息。层次和相似关系则是表达构件间的非功能关系。关系的紧密程度和重要性也是一种重要语义信息，本文将其量化成具体的权重值，作为关系的属性。

我们用对象属性<owl:ObjectProperty>和子属性<rdfs:subPropertyOf>来描述关系的从属结构，用 ObjectProperty 的 Annotaiton 中的<rdfs:comment>来标注关系的权重。本文对权重大小做如下量化处理：设某功能与其自己的关系称为相等关系，关系权重为 1；两个功能完全没有关系称为零关系，关系权重为 0，这两个关系均不用来描述功能关系，仅作为关系权重的基准。因此，其它所有关系的权重在 0 和 1 之间。

hasDepdRelation——依赖关系，该关系提取自功能刻面，即(1)中提取的二元关系集合 I。构件依赖可以非正式地定义为一个构件对另一个构件的功能性、配置性、数据性、控制性的依赖[7]，这种依赖一般体现在构件功能上。本文根据依赖关系的强弱程度将其分为主动依赖、被动依赖和相互依赖。

主动依赖：假设存在关系 $Depd_i(f_m, f_n) \in I$，但 $Depd_j(f_n, f_m) \notin I$，则称 f_m 主动依赖于 f_n。

被动依赖：在上述条件下，f_n 被动依赖于 f_m。

相互依赖：假设存在关系 $Depd_i(f_m, f_n) \in I$，但 $Depd_j(f_n, f_m) \in I$，则称 f_n 和 f_m 互相依赖。

三者的关系权重为被动依赖 < 主动依赖 < 互相依赖的关系权重。

根据以上定义，本文将集合 I 中依赖归为以下几类：hasControlDepd(控制依赖)、hasDataDepd(数据依赖)、hasFuncDepd(功能依赖)、hasCommuni-cationDepd(通讯依赖)、以及它们的反向关系isControlDepdBy、isDataDepdBy、isFuncDepdBy、isCommunicationDepdBy，还有依赖性很强的hasMutualInitiativeDepd(相互主动依赖)。它们都被定义成 hasDepdRelation(依赖关系)的子关系。

前面四种依赖属于主动依赖，表示较强的依赖关系，可设置较大的权重。后面四者属于被动依赖，是前四者的反向关系，定义时用<owl:inverseOf>来标注，它们的依赖程度较弱，可以设置较小的权重。这八种依赖关系具有传递性，都用<owl:TransitiveProperty>表示。最后一种互相主动依赖属于互相依赖，表示更强的依赖关系，可设置更大的权重。该关系具有对称性，用<owl:SymmetricProperty>来表示。

图 1 功能语义层所示即为从构件语义关系层中提取的三类典型的依赖关系。它们是 a~j 这 10 个构件间这三类依赖在功能层上的抽象表示。

hasSimilarRelation——相似关系。该关系用来描述构件实例间的非功能关系，是用户在构件入库时指定的。其子关系有 hasSmlrInterface(接口相似)和hasSmlrAlgorithm(算法相似)。前者表示二者提供相似接口，例如两个遵循业界标准 API 开发的同类功能构件。后者表示二者实现某一功能时采用相同的算法。这类关系往往比较紧密，具有对称性，权重可设为较大值。

hasHierarchicalRelation——层次关系，也是非功能关系。其子关系有 hasPart(包含关系)和hasUpdateVersion(版本升级)。前者表示一个构件拥有子功能，一般用来描述复合构件和原子构件，或复合构件之间的关系。后者表示构件拥有升级版本。他们都有对应的反向关系 isPartOf 和isUpdateVersionOf。这些关系紧密程度适中，具有传递性，权重可设为中间值。

(4) 其他刻面的映射

在一个基于刻面的构件库系统中，往往还有其它

刻面，如本文的信用构件库中还有“应用领域”、“构件类型”、“应用环境”刻面。这些刻面与功能刻面正交，是构件在其他刻面的描述，可视为构件的一种属性，用<owl:DatatypeProperty>表示，我们设置 hasDomainArea、hasCompType、和 hasEnviron-ment 与其余刻面一一对应[9]。

2.2.2 实例和公理的建立

实例表示某个特定概念类的具体化的值，在构件本体中即为具体的构件，建立实例的过程就是构件入库的过程。功能刻面的选择决定了实例所属的构件类别，其它刻面的选择决定了相应数据属性的值。

实例类的建立过程也是公理建立的过程。公理代表永真断言，可以用来约束信息、证明正确性或者推导新信息。在本体中加入公理意味着能表达更为丰富的概念间的关系。

公理包括 Tbox 和 Abox[10]。Tbox 引入本体中的术语表；Abox 包含对个体的断言和关系断言。

Tbox 包含了本体描述领域的内涵知识，通常以术语公理的形式描述概念和任务的事实。一般术语公理有两种形式：蕴含和等式。本文中，构件本体中的概念类及其关系即为 Tbox。

Abox 包含了外延知识，通常以实例化公理断言描述个体和个体间关系的事实。一般实例化公理有两种形式：概念断言(形如 C(a)) 和关系断言(形如 R(a,b))。在构件入库时选择功能刻面下术语就对应一个概念断言的生成；而用户指定该构件与其他构件的关系就对应一个关系断言的生成。

规则是增加陈述的束缚机制，是逻辑的核心形式的扩展[10]，能更好的表达知识并协助推理。一部分规则是 OWL 语言中的隐含的语义称为内建公理，另一部分规则由用户定义称为用户自定义公理。表 1 和表 2 分别列举了这两类规则。

2.3 本体的形式化描述

本体的形式化定义五花八门，但普遍认同本体的五种基本元素，本文参照 OWA[11]的定义，将本体定义为一个五元组，O ={ C , A^C , R , A^R , X }：

C：表示概念集合，包括所有父类和子类构件概念类。

AC：表示在概念集 C 上的属性集，即概念类的 DatatypeProperty；

表 1　OWL 内建公理示例

subClassOf	(?A rdfs:subClassOf ?B)∧ (?B rdfs:subClassOf ? C) → (?A rdfs:subClassOf ?C)
DisjointWith	(?C owl:disjointWith ?D)∧(?x rdf:type ?C)∧ (?y rdf:type ?D) → (?x owl:differentFrom ?y)
SymmetricProperty	(?P rdf:type owl: SymmetricProperty)∧ (?a ?P ?b)→ (?b ?P ?a)
inverseOf	(?P owl:inverseOf ?Q)∧(?x ?P ?y) → (?y ?Q ?x)

表 2　用户自定义公理示例

UserRule 1	(?x hasPart ?y)∧(?y hasSmlrInterface ?z) → (?x hasPart ?z)
UserRule 2	(?x hasFuncDepd ?y)∧(?y hasPart ?z) → (?x hasFuncDepd ?z)
UserRule 3	(?x hasDataDepd ?y)∧(?y hasSmlrInterface ?z) → (?x hasDataDepd?z)

R：表示概念间关系和实例间关系的集合，包括所有用户定义的 ObjectProperty 和 OWL 内建的关系如 subClassOf 等；

A^R：表示关系的属性集，本文为 ObjectProperty 定义了权重属性；

X ：表示公理集合，包括 Tbox、Abox 和规则集合。

这个形式化描述作为本体的核心结构，被普遍接受并易于用现有的本体语言来进行描述，如本文采用的 OWL。下面是由信用构件库的功能刻面映射而成的构件本体的部分形式化表述：

C = { 构件类 | 数据管理构件、图表显示构件、数值统计构件、均值计算构件、表格显示……}

A^C= { 属性|A^C(同义词)，A^C(应用领域)，A^C(构件类型)，A^C(使用环境)……}

R = {概念从属关系和构件关系 | subClassOf(数据处理构件，构件)，subClassOf(元数据采集构件，数据处理构件)，hasDataDepd(具体构件 1，具体构件 2)，hasUpdateVersion(具体构件 3，具体构件 4)……}

A^R= {关系属性|A^R(关系权重) }

X = {hasDataDepd(具体构件 1，具体构件 2)∧(hasDataDepd owl:inverseOf isDataDepdBy)

→isDataDepdBy(具体构件 2，具体构件 1)、

hasFuncDepd(具体构件 3，具体构件 4)∧
hasPart(具体构件 4，具体构件 5)
→hasFuncDepd(具体构件 3，具体构件 5)、
hasDataDepd(具体构件 6，具体构件 7)∧
hasSmlrInterface(具体构件 7，具体构件 8)
→hasDataDepd(具体构件 6,具体构件 8)……}

用本体来描述构件及其关系，就将构件概念置身于一个多维的描述空间，从纵向(层次关系)、横向(功能依赖关系)和平行(相似关系)这三个维度来描述构件，使刻面下本来只有抽象和具体的层次描述结构关系扩展成具有多种关系的网状描述结构，增加了语义信息，增强了描述能力。图 2 为基于功能语义的构件本体的示意图。

3　构件本体对构件检索的支持

构件本体一方面可以帮助用户更加准确、完整地描述自己的检索要求，另一方面可以为构件描述提供丰富的语义注解，从而更好的弥合用户需求和构件描述之间的“鸿沟”，提高了检索的查准率和查全率。基于功能语义的构件本体可在以下几个方面支持构件检索：

(1) 构件入库

新入库构件的信息除存到数据库外，还将构件作为概念的实例加入到本体库中。新入库构件的信息(包括构件所属类别和与指定库中构件关系)将作为新的断言加入 Abox。然后用新断言和已有断言(已入库构件信息)以及预定义规则一起推理出隐含的关系断言(即新构件与除用户指定相关构件外的其它构件的关系)，并再次将其加入 Abox。随着构件数量的增多，Abox 会不断增大，影响相关功能推荐中的查询效率，因此本系统将 Abox 中的关系断言进行转化存储到关系数据库中。虽然这会造成数据冗余，但提高了检索效率，是以空间换时间的策略。

(2) 关键字查询的语义扩充

在关键字查询中，用户给出原始关键字集合作为初始查询条件，系统查找本体中是否具有关键字的同义词，若有则计算其同义词与其他概念的语义匹配度，将匹配度大于某一阀值的概念加入初始查询条件；若没有则在通用本体知识库(如知网)查询相关词汇加入初始查询条件。这种查询扩充发挥了本体作为构件库知识基础的作用，帮助用户更好的描述查询要求，提高查全率。

(3) 刻面语义匹配查询

刻面检索是一个树匹配的过程，一般查询树与跟它有父子关系和包含关系的描述树具有较高的匹配度[4]。语义匹配则既有树匹配的特点又考虑概念的语义相似度。例如在构件本体中柱状图显示构件与数据转换构件这两个概念虽然有不同的父类，即在概念上没有公共祖先节点，但具有权重较高的 hasDataDepd 依赖关系，因此这两个概念的构件实例也会有较高的语义相似度，当用户在查询柱状图显示构件时，数据采转换类构件也会被加入结果集返回给用户。

(4) 相关构件推荐

当用户查看具体构件时，系统将查询由 Abox 断言转化得到的实例构件关系表，根据构件类的语义相关度和关系权重的大小计算构件相关度，将相关度高的构件推荐给用户。依赖本体的 Abox 推理，系统能找到构件间的隐含语义关系，为用户提供潜在匹配构件。

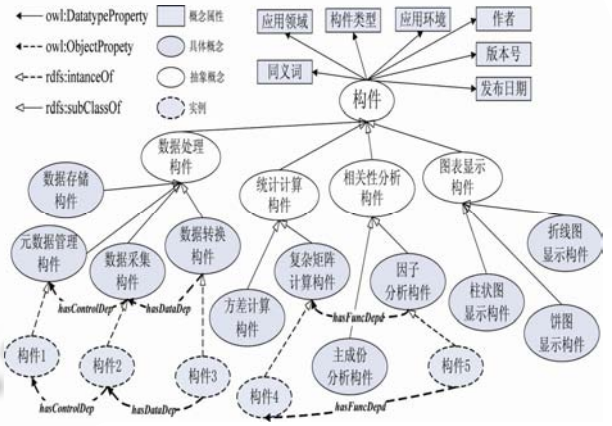


图 2　基于功能语义关系的构件本体模型

综上所述，检索系统能够借助构件本体“全面了解”构件描述中的语义信息，同时“智能理解”用户的查询需求，为用户提供人性化的检索服务。

4　总结与展望

传统的构件描述方法具有较大的语义缺失性，无法很好的满足用户的构件检索需求。本文分析了目前主流的刻面分类描述方法的优缺点，对其进行改进，对信用评估构件库的功能刻面建立基于功能语义的构件本体，将构件功能描述中丰富的语义信息纳入构件本体中，提高了构件描述的精确性和完整性，也提高

了检索的查准率和查全率。

本文所提出的本体建模方法仅针对常见的功能刻面做语义分析和本体建模，对其他刻面则是简单的将其作为数据属性处理。在目前基于刻面分类的构件库系统中，除了功能刻面外还有其他刻面也含有丰富的语义信息，例如信用构件库中的应用领域刻面就可以映射成一个领域构件本体，描述构件在领域中的相关性。但应用领域刻面的建立需要领域专家的参与，与功能刻面的建立完全不同，因此将其映射成领域构件本体也无法套用本文所述方法。如何同时将多个刻面下信息同时映射成构件本体，并支持高效的语义检索，将是本文需要进一步研究的课题。

参考文献

1 贾育,顾毓清.基于领域特征空间的构件语义表示方法.软件学报, 2002,13(2):311－316.

2 杨明华,钱乐秋.基于本体的构件描述关键技术研究[硕士学位论文].上海:复旦大学, 2006.

3 陈颖,沈军.基于本体的构件描述与检索.计算机应用与软件, 2007,24(7):30－32.

4 王渊峰,张涌,钱乐秋.基于刻面描述的构件检索.软件学报, 2002,13(18):1546－1551.

5 Borst WN. Construction of Engineering Ontologies for Knowledge Sharing and Reuse [Ph. D. Thesis]. University of Twente, Enschede, 1997.

6 Prieto-Diaz R. A Faceted Approach to Building Ontologies. Proceedings of IEEE International Conference on Information Reuse and Integration(IRI 2003), 2003:458－465.

7 Kon F, Campbell R H. Dependence management in component-based distributed systems. IEEE Concurre-ncy, 2000,8(1):26－36.

8 Zhao JJ, Ushij Ima K. A dependence-based representation for concurrent object-oriented software maintenance. Proceedings of the Second Euromicro Conference. Washington DC: IEEE Computer Society Press, 1998:60－66.

9 范菁,刘涛,熊丽荣. 信用构件的刻面分类及检索方法研究.计算机系统应用, 2008,17(6):52－56.

10 Baader F, Nutt W. Basic Description Logics. Baader F, McGuinness, Nardi D, et al. eds. The Description Logic Handbook. Cambridge Univ Press, 2003.

11 Naing Myo-Myo, Limy Ee-Peng, Hoe-Lian DG. Ontology-based Web Annotation Framework for HyperLink Structures. Proc. of the Third Intl. Conf. on Web Information Systems Engineering (WISE). Singapore, 2002.