面向语义构件检索的交互式查询方案生成

蔡怡峰，彭 鑫，钱乐秋

(复旦大学计算机科学与工程系软件工程实验室, 上海 200433)

摘 要：基于语义的构件检索过程中存在的一个突出问题是用户对问题域的理解以及所熟悉的自然语言表达方式与构件的规范语义描述之间存在差异，从而导致用户很难准确刻画自己的检索请求. 针对这一问题，本文提出了一种交互式的查询方案生成方法. 该方法通过概念映射将用户的自然语言查询转换为本体描述，从而确定用户的查询方案. 其中，概念映射需要通过与用户的交互式会话过程完成. 该方法可以充分挖掘潜在的用户查询语义，对于提高构件检索的查准率和查全率、减少查询努力具有十分重要的作用.

关键词：构件检索；语义；本体；会话；交互；查询生成

中图分类号： TP311 文献标识码： A 文章编号：0372-2112（）071476

An Interactive Query Generation Method for Semantics-based Component Retrieval

CAI Yi-feng, PENG Xin, QIAN Le-qiu

(Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

Abstract: One major problem in semantics-based component retrieval process is the discrepancy between nature language which users are familiar with and formal semantic descriptions of components, which makes users difficult to describe retrieve queries precisely. An interactive query generation method is proposed to solve this problem. The method uses concept identification process to map users' nature language to ontology descriptions of components, in which concept identification uses conversational process to interact with users. This method extracts implicit semantics in user queries so that it leads to improve recall and precision rates, reduce query effort in component retrieval.

Key words: component retrieval; semantic; ontology; conversation; interaction; query generation

1. 引言

基于构件的软件开发（CBSD）提出复用软件构件，而不是采用一切“从零开始”的方式，来组装特定领域的应用系统. CBSD可以充分利用已有的开发成果，减少重复劳动，提高软件的开发效率和质量. 然而当存在大量构件时，在构件库中检索符合特定复用需求的构件变得十分困难. 因此可复用构件的描述和检索一直是CBSD的一个研究重点[1].

基于语义的构件检索在各个步骤都考虑到用户查询的语义以及相关领域的知识，因而具有较高的查准率和查全率[2]. 文献[3]对基于语义的构件描述和检索方法的总体框架进行了阐述，认为基于语义的构件描述和检索方法应该包含以下三方面的优势：（1）将软件构件视为领域本体基础上的语义描述体；（2）允许用户使用自然语言表达检索请求，然后将其自动转化为语义描述框架；（3）在领域本体基础上通过用户查询语义表示与构件语义描述之间的语义匹配来改进检索效果. 该框架概括了基于语义的构件描述与检索的基本特征，为以后的工作指明了方向. 文献[4]对信息检索领域进行了研究，认为查询扩展技术以及通过会话获得用户反馈来优化用户查询，能够取得更好的检索效果. 我们认为这种思想同样适用于构件检索. 文献[5]在基于案例的推理（Case-Based Reasoning）方法基础上，提出通过会话的方式优化用户查询，但假设用户十分了解领域本体的表示方式，并遵从系统的要求输入检索请求，这显然限制了检索系统的实际应用. 而本文所提出的方法在用户的自然语言表述基础上通过概念映射获取初步的语义查询方案，因此可以解决这一问题.

从当前一些研究工作看，基于语义的构件检索存在的一个突出问题，即用户对问题域的理解以及所熟悉的自然语言表达方式与构件的规范语义描述之间存在差异. 针对这个问题，本文提出了一种面向语义构件检索的交互式查询方案生成方法. 该方法通过概念映射将用户的自然语言查询转换为本体描述，从而确定用户的查询方案. 其中，概念映射充分利用了用户自然语言中的上下文（主要由谓语、宾语、

状语和定语组成）信息以及领域本体中的语义上下文之间的相似度信息，同时辅以交互式会话过程. 该方法可以充分挖掘潜在的用户查询语义，对于提高构件检索的查准率和查全率、减少查询努力具有十分重要的作用.

2. 基于本体的构件的描述

在基于语义的构件描述和检索中，本体（Ontology）往往作为知识基础存在，相关工作如文献[2，5，6]等. 本体是对特定领域的显示概念化，并提供了对该领域普遍和共享的知识理解[7]. 对于语义构件检索来说，本体可以为构件以及查询的描述提供明确的语义词汇表，并且可以支持基于公理和语义规则的推理. 此外，W3C所推荐OWL (Web Ontology Language)[8]等本体描述语言也使得基于本体的构件描述和检索更加容易. 本文的方法同样采用OWL作为领域模型和构件语义的描述基础，并定义了相应的本体元模型. 在此基础上，我们定义了构件的语义描述框架.

2.1 功能本体元模型

构件的功能语义是复用者了解并判断构件可复用性的首要依据[9]. 因此本文主要关注于构件的功能语义描述. 我们在前期基于本体的构件语义描述工作[9]基础上，定义了面向构件语义描述的领域本体元模型. 元模型主要由构件的功能操作（Operation）、操作对象（Object）、功能操作和操作对象的语义刻面（Facet）、刻面取值术语（Term）以及功能操作和操作对象之间的操作关系（ActOn）组成. Operation和Object分别表示构件所提的功能操作和所操作的业务对象. Facet用于精确描述Operation和Object某个方面的语义特征，其值域为术语（Term）或者普通数据类型.

在这个本体元模型的基础上，我们可以定义特定领域的本体模型. 图 1 给出了多媒体处理领域本体模型的一个片断，其中处理（Process）操作有四个子功能操作，分别是读取（Read）、缩放（Resize）、旋转（Rotate）、剪辑（Edit）；媒体（Media）对象有三个子操作对象，分别是图像（Image）、视频（Video）、音频（Audio）. Read 操作在 Image、Video、Audio 上，表示读取一个多媒体对象，同时还有一个语义刻面 HasLocation，表示读取的位置. Resize 和 Rotate 操作在 Image 上，表示改变图像大小和旋转图像角度. Edit 操作在 Video 和 Audio 上，表示剪辑视频和音频. Image 有两个语义刻面，分别是图像格式（HasFormat）和图像类型（HasType）. Video 和 Audio 的语义刻面 HasFormat 的含义与 Image 相同.

2.2 构件语义描述框架

构件语义描述框架是构件语义的规范表示，同时也是用户构件查询方案的描述模板. 在我们的方法中，查询用户不需要熟悉此描述框架以及所使用的领域本体，其查询方案将通过交互式会话进行启发并根据语义关系自动进行推理. 本文所使用的构件语义描述框架以构件的功能语义作为主要的描述内容. 下面首先定义本体关系表示法和概念约束集合，并在此基础上给出本文所使用的构件语义描述定义.

定义 1（本体关系表示法）：对于∀prop∈owl:Property，∀subj∈domain(prop)（prop 的定义域），∀obj∈range(prop)（prop 的值域），如果领域本体中 subj 和 obj 之间存在 prop 关系，则表示为三元组<subj, prop, obj>.

定义 2（概念约束集合）：对于 ∀ concept ∈ Operation ∪ Object，concept 上的约束集合 FTs(concept)={<facet, facetValue> | facet ∈ Facet ∧ facetValue ∈ Term ∪ xsd:Boolean ∪ xsd:Integer ∧ 满足<concept, facet, facetValue>}.

定义 3（构件语义描述）：构件语义描述 CSD=<operation, FTs(operation), object, FTs(object)>. 其中 operation∈Operation∧object∈Object∧（∃actOn∈ActOn，满足<operation, actOn, object>）.

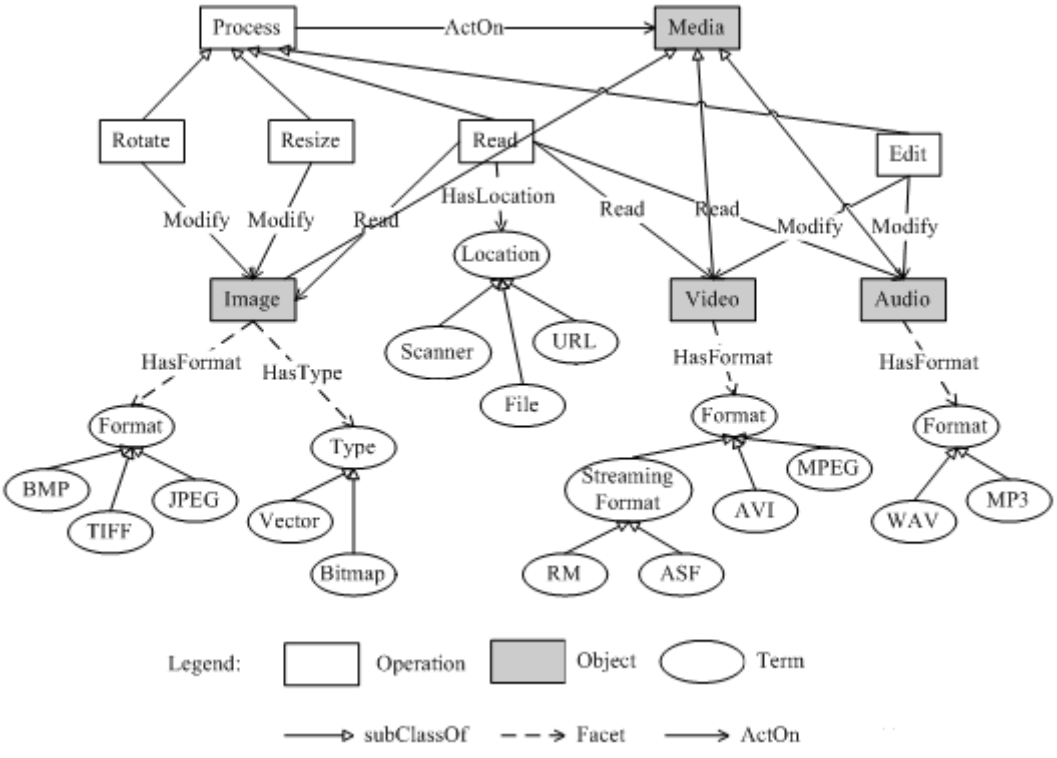


图 1　多媒体处理领域的本体模型

3 概念映射

生成构件查询的主要步骤是概念映射. Bailin等[6]认为用户的复用需求隐含着基于个人认知的领域本体，而揭示复用需求和构件描述的真实含义的过程是一个本体协商的过程. 他们在文献[6]中提出了一个语义构件检索的理想场景，即通过语义推理和一系列问答式的会话过程，将一个本体中的概念映射到另一个本体中的概念，就如同构件用户和构件提供者直接面对面地进行探讨并达成共识那样，但并没有实现. 而本文在概念映射这一步骤中就是要通过用户与构件库系统之间的会话式过程完成这一本体协商任务.

为了完成自然语言词汇到领域本体概念之间的转换，可以使用预先建立的概念词典[10]. 在语义信息缺乏的情况下，概念词典只能实现词汇到本体概念的多对一映射从而解决自然语言中的多词同义现象，而自然语义的一词多义问题就比较难解决了，例如Java一词可以同时表示Java编程语言（Programming Language）和一种咖啡（Coffee）. 因此，语义概念映射应该考虑自然语言上下文与本体语义上下文之间的关系. 例如一词多义问题可以根据用户输入的句子中的所有词汇以及词汇之间的语法关系进行整体的概念映射来解决：当Java和Class、Object等词汇一同出现时，Java被映射为一种编程语言；而当Java和Drink、Cup等词汇一同出现时，Java被作为一种咖啡. 如果有多个候选取值，系统可以通过会话的方式，征询用户的意见，得到准确的映射. 同时系统也可以根据反馈结果不断丰富概念词典.

3.1 概念映射算法

本文假设用户输入若干英语的自然语言动宾短语，描述所需构件的各个功能. 系统分析每一个动宾短语，把它们映射为构件的规范语义描述（定义 3），从而使构件库系统能够识别并与已有的构件描述进行匹配. 在映射过程中，用户句子中的动词和宾语将被映射为构件的功能操作概念和操作对象概念，而状语和定语则分别映射为功能操作和操作对象的语义刻面描述. 例如用户输入构件查询“Input a JPEG image from scanner”，系统将其映射为 CSD=<Read, {<HasLocation, Scanner>}, Image, {<HasFormat, JPEG>}>.

首先我们给出用户查询短句的结构定义，这种结构可以使用语法分析工具（例如MINIPAR[11]）对用户输入句子进行切分得到.

定义 4（动宾短语）：用户输入的英语动宾短语可以描述为四元组 NL=<P, O, ADVs, ATTs>，其中 P 是谓语，O 是宾语，ADVs 是状语（描述谓语）词汇集合，ATTs 是定语（描述宾语）词汇集合.

定义 5（概念和词汇的词根集合）：对∀concept∈owl:Class∪owl:Property，R(concept)表示 concept 名字标签的词根集合；对任何自然语言短语 x，R(x)表示 x 包含的所有词汇的词根集合.

定义 6（刻面词汇集合）：对∀concept∈Operation∪Object，Fwords(concept)={x | x∈$\bigcup_i$R($facet_i$)，其中$facet_i$是领域本体中第i个约束描述concept的刻面概念}.

定义 7（术语词汇集合）：对∀concept∈Operation∪Object，Twords(concpet)={x | x∈$\bigcup_i$R($term_i$)，其中$term_i$是领域本体中第i个约束描述concept的术语概念}.

定义 8（约束词汇集合）：对∀concept∈Operation∪Object，FTwords(concept)=Fwords(concpet)∪Twords(concpet).

在此基础上我们给出构件查询方案概念映射算法如图 2.

输入：四元组 NL
输出：构件语义描述 Query_CSD

(1) 根据本文提出的相似度计算公式（见 3.2 小节公式 6）计算<P, O>到每一个<operation, object>的相似度 SIM. 其中 operation∈Operation∧object∈Object∧（∃actOn∈ActOn，满足<operation, actOn, object>）.
(2) 把 SIM 超过阈值 minSIM 的所有映射作为候选集合.
(3) 如果候选集合不空，把候选集合根据相似度 SIM 排序. 假设相似度最大的映射 SIM=k，则取出所有相似度大于（k - maxDIFF）映射作为最佳映射集.
(4) 如果最佳映射集只有一个元素，则本四元组 NL 的映射完成，跳到(6).
(5) 如果最佳映射集不止一个元素，则启动会话过程，把这些映射呈现给用户，由用户确定最佳映射.
(6) 假设最佳映射为<$operation_i$, $object_j$>.
(7) 如果概念词典中没有P到$operation_i$的映射，则把该映射加入.
(8) 如果概念词典中没有O到$object_j$的映射，则把该映射加入.
(9) 对每一个状语词汇 x∈ADVs
(10) 如果x∈Twords($operation_i$)，则在FTs($operation_i$)中加入概念约束二元组<facet, x>，其中facet∈Facet且满足<$operation_i$, facet, x>.
(11) 如果x∈Twords($object_j$)，则在FTs($object_j$)中加入概念约束二元组<facet, x>，其中facet∈Facet且满足<$object_j$, facet, x>.
(12) 对每一个定语词汇 x∈ATTs
(13) 如果x∈Twords($object_j$)，则在FTs($object_j$)中加入概念约束二元组<facet, x>，其中facet∈Facet且满足<$object_j$, facet, x>.
(14) 创建Query_CSD=<$operation_i$, FTs($operation_i$), $object_j$, FTs($object_j$)>.
(15) 返回 Query_CSD，算法完成.

图 2 构件查询语句概念映射算法

本算法分两步完成，首先通过相似度计算和会话，把二元组<P, O>映射到本体中有 ActOn 属性关联的<operation, object>二元组，随后把用户输入的状语集和定语集映射到本体中约束描述操作行为和业务对象的刻面和术语.

算法的第 11 步把状语词汇映射到本体中约束描述操作对象（objcet）的术语词汇，这种情况一般出

现在方向状语包含的名词中. 比如 Convert an image from BMP format to JPEG format，其中的 from BMP format to JPEG format 在语法上属于状语（描述谓语），但 BMP 和 JPEG 描述的却是宾语.

3.2 概念映射的相似度

本文的概念映射主要是根据用户自然语言查询句子与领域本体中相关概念语义上下文的整体相似度来进行判断，包括用户输入的自然语言词汇与领域模型中的概念间的词汇相似度以及上下文信息之间的结构相似度.

NL 中的谓语 P 和领域模型中的某个操作行为 operation 的词汇相似度如下：

$$SIM_L(P \to operation) = \begin{cases} 1 & \text{, 如果概念词典中有}P\text{到}operation\text{的映射} \\ |R(P) \cap R(operation)| / |R(P) \cup R(operation)| & \text{, 否则} \end{cases} \quad (1)$$

类似地，NL 中的宾语 O 和领域模型中的某个业务对象 object 的词汇相似度如下.

$$SIM_L(O \to object) = \begin{cases} 1, & \text{如果概念词典中有}O\text{到}object\text{的映射} \\ |R(O) \cap R(object)| / |R(O) \cup R(object)|, & \text{否则} \end{cases} \quad (2)$$

自然语言的上下文信息主要体现在状语和定语上. 如果状语集合的词汇出现在约束描述某个功能操作概念的刻面或术语词汇集合中，则谓语P和该功能操作operation的结构相似度增加；如果定语集合的词汇出现在约束描述某个操作对象概念的刻面或术语词汇集合中，则宾语O和该操作对象object的结构相似度增加. 另外，我们使用MINIPAR[11]对CodeProject[12]所提供构件的 40 多个功能进行语法分析后发现，定语集合的词汇出现在操作对象的标签词汇中同样会增加宾语O和该操作对象object的结构相似度*. 基于和 3.1 节算法第 11 步相同的考虑，状语集合的词汇出现在约束描述某个操作对象概念的刻面或术语词汇集合中，宾语O和该操作对象object的结构相似度也会增加.

根据以上分析，宾语 O 和某个操作对象 object 的结构相似度、谓语 P 和某个功能操作 operation 的结构相似度分别如下：

$$SIM_C(O->object) = \frac{|\{x \mid x \in ATTs \cap (FTwords(object) \cup R(object))\}| + |\{x \mid x \in ADVs \cap FTwords(object)\}|}{|\{x \mid x \in ATTs\}| + |\{x \mid x \in ADVs \cap FTwords(object)\}|} \quad (3)$$

如果分母为 0，表示用户输入可能没有定语词汇，则SIM_C (O→object)=N/A.

$$SIM_C(P->operation) = \frac{|\{x \mid x \in ADVs \cap FTwords(operation)\}|}{|\{x \mid x \in ADVs\}| - \max_j |\{x \mid x \in ADVs \cap FTwords(object_j)\}|} \quad (4)$$

其中，∃actOn∈ActOn，满足<operation, actOn, $object_j$>. 如果分母为 0，表示用户输入可能没有状语词汇，则SIM_C (P→operation)=N/A.

综合文字相似度和结构相似度，P 和 operation 的相似度公式如下（O 和 object 的相似度公式类似）：

$$SIM(P \to operation) = \begin{cases} SIM_L(P \to operation) & , SIM_C(P \to operation) = N/A \\ (SIM_L(P \to operation) + SIM_C(P \to operation))/2 & , SIM_C(P \to operation) \neq N/A \end{cases} \quad (5)$$

最后，二元组<P, O>到有 ActOn 属性关联的<operation, object>二元组的相似度如下：

$$SIM(<P, O> \to <operation, object>) = (SIM(P \to operation) + SIM(O \to object))/2 \quad (6)$$

3.3 概念映射实例

本节例举一个实例，演示系统生成构件查询的过程. 假设用户检索多媒体处理领域的软件构件，领域模型如 2.1 节图 1 所示，图 2 算法中的 minSIM=0.5，maxDIFF=0.1，概念词典中尚不存在任何映射.

用户输入构件查询“Input a JPEG image from scanner”. 经过 MINIPAR 分析后得到 P=Input，O=Image，ADVs={from, scanner}，ATTS={JPEG}. 概念映射的相似度计算结果见表 1.

---

* 这是由自然语言的特性决定. 自然语言往往会把操作某个事物，说成操作某个事物的属性. 比如Rescale an image表示放大缩小图片，用自然语言可能表达成Rescale the size of an image，其中image成为size的定语. 在计算相似度时，我们需要把size of an image映射到image.

表 1　概念映射的相似度

SIM(P→operation)		operation				
		Process	Rotate	Resize	Read	Edit
P=Input	SIM_L	0	0	0	0	0
	SIM_C	0	0	0	0.5	0
	SIM	0	0	0	0.25	0

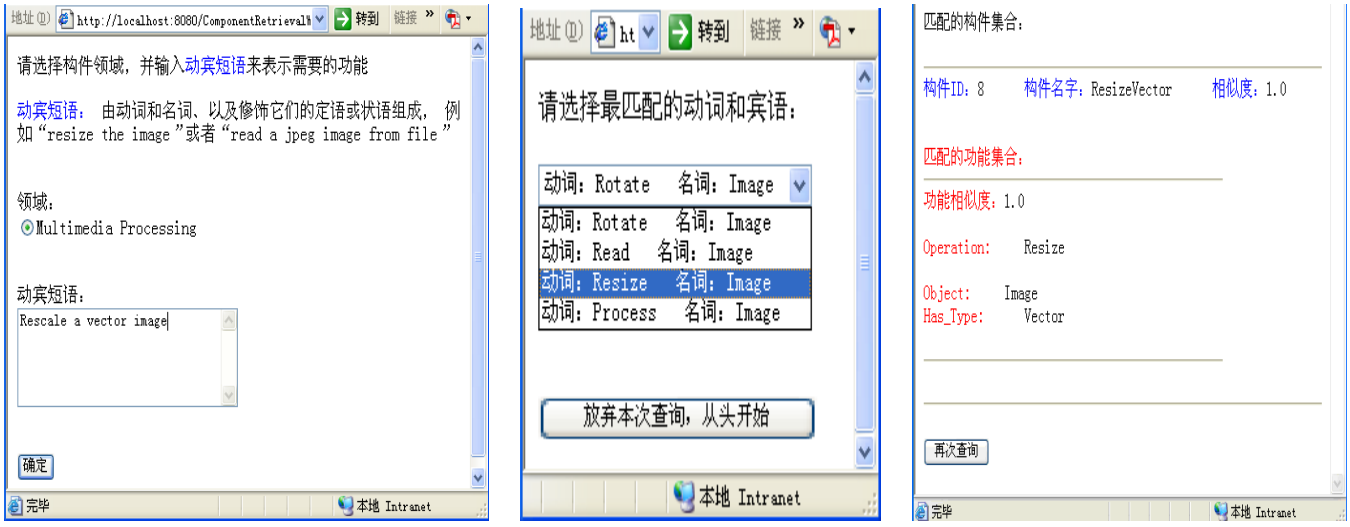
SIM(O→object)		object			
		Media	Image	Video	Audio
O=Image	SIM_L	0	1	0	0
	SIM_C	0	1	0	0
	SIM	0	1	0	0

SIM(<P, O> → <operate, object>)		operation				
		Process	Rotate	Resize	Read	Edit
object	Media	0				
	Image	0.5	0.5	0.5	0.63	
	Video	0			0.13	0
	Audio	0			0.13	0

由于<Read, Image>的相似度超过了其他所有二元组 maxDIFF=0.1 以上，因此被作为最佳映射. 同时，映射 Input→Read 被加入到概念词典中，辅助以后的相似度计算. 由于状语词汇 Scanner 出现在 Twords(Read)中，FTs(Read)中加入约束描述<HasLocation, Scanner>；定语词汇 JPEG 出现在 Twords(Image)中，FTs(Image)中加入约束描述<HasFormat, JPEG>. 因此，创建构件查询步骤生成的构件语义描述 Query_CSD=<Read, {<HasLocation, Scanner>}, Image, {<HasFormat, JPEG>}>.

4. 系统实现与实验分析

本文实现了整个构件检索系统，其中Jena API[13]被用来操作本体，生成构件查询之后的语义匹配算法综合了语义关系[14]和语义距离[15]，得到用户查询与每个构件的相似度. 检索过程中各个步骤的界面如图 3.

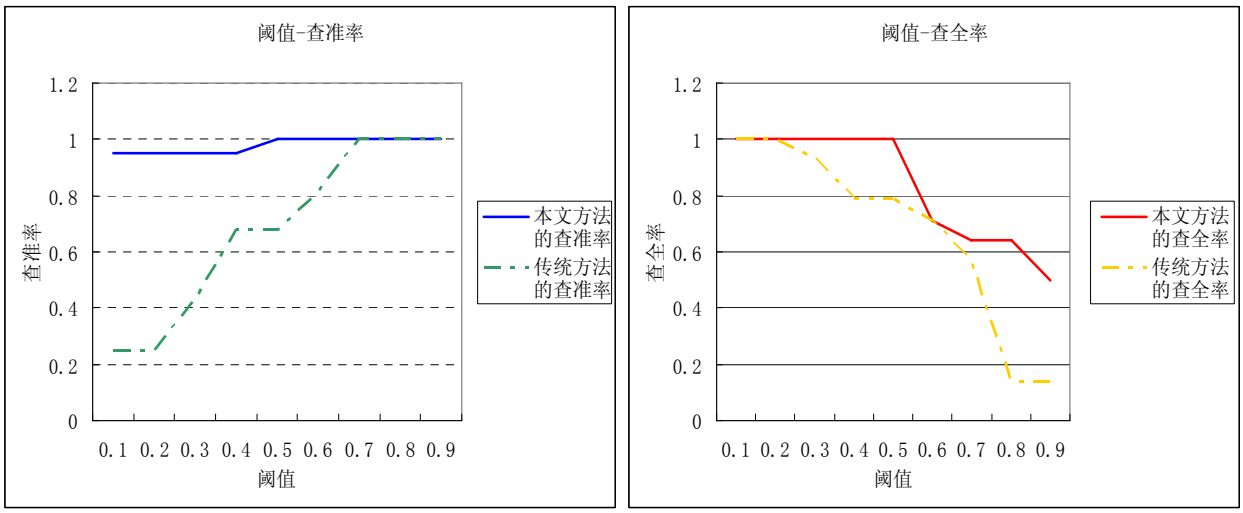


(a) 用户输入自然语言动宾短语　(b)会话过程：用户选择最佳匹配　(c) 系统返回匹配构件

图 3 构件检索系统界面

本文以CodeProject[12]上的开源构件作为数据进行了实验分析，来检验我们提出的概念映射步骤对查准率和查全率的贡献. 作为对比方法（下文称为传统的构件检索方法），传统方法的匹配度计算算法与本文一样，只是跳过了概念映射这步. 由于缺少了概念映射，因此传统方法无法捕获各个词汇之间的关系，只能单独处理每一个单词. 我们以不同的相似度作为阈值，得到查准率和查全率的折线图如图 4. 从图 4(a)可以看出，在阈值比较低的情况下，本文方法的查准率明显优于传统方法，这是因为本文方法经过概念

映射之后，已经缩小了匹配范围. 而传统方法却与一些不正确的构件有相似度，导致低阈值情况下的查准率较低. 当我们提高阈值后，传统方法的查准率会接近并最终达到本文的方法. 但从图 4(b)可以看出，当阈值升高后，传统方法查全率的下降速度明显快于本文的方法. 通过上述分析可以得知，传统方法的缺点在于，它对不正确的构件会有一些相似度（导致低阈值下的查准率低），而对于正确的构件，相似度又不足够高（导致高阈值下的查全率低）. 这表明传统方法的辨别力度不够，而本文的方法恰好弥补了这个缺点.



(a) 阈值-查准率　　(b) 阈值-查全率

图 4 查准率和查全率折线图

5. 讨论和总结

基于语义的构件检索过程中存在的一个突出问题是用户对问题域的理解以及所熟悉的自然语言表达方式与构件的规范语义描述之间存在差异，从而导致用户很难准确刻画自己的检索请求. 本文通过概念映射以及基于语义的交互式会话过程辅助用户生成符合用户复用需求以及构件库构件描述框架的构件查询方案，从而大大提高了构件库中构件的复用机会.

本文方法的限制是对知识工程的依赖，假定领域模型已经被完善定义，但这需要很大的工作量. 然而，如今知识获取和建模领域的发展，例如本体工程，已经提供了系统的方法，使得这个限制得到了很大的缓解[16].

我们开发完成了支持本文方法的原型系统，并根据自由代码网站CodeProject[12]上多媒体领域的构件信息对本文的方法进行了实验（第 4 部分），结果表明本文的方法对于降低用户的检索难度以及提高查准率和查全率有着明显的作用. 我们将继续对构件的语义匹配进行深入研究.

参考文献：

[1] 王渊峰, 张涌, 任洪敏, 等. 基于刻面描述的构件检索[J]. 软件学报, 2002, 13(8): 1546-1551.
Wang Yuanfeng, Zhang Yong, Ren Hongmin, et al. Retrieving components based on faceted classification[J]. Journal of Software, 2002, 13(8): 1546-1551. (in Chinese)

[2] V Sugumaran, V C Storey. A semantic-based approach to component retrieval[J]. The DATA BASE for Advances in Information Systems, 2003, 34(3): 8-24.

[3] Yao Haining, L Etzkom. Towards a semantic-based approach for software reusable component classification and retrieval[A]. Proceedings of the 42nd Annual Southeast Regional Conference[C]. New York: ACM Press, 2004. 110-115.

[4] A F Zazo, C G Figuerola, J L A Berrocal, E Rodriguez. Reformulation of queries using similarity thesauri[J]. Information Processing and Management, 2005, 41(5): 1163-1173.

[5] Gu Mingyang, K Bø. Component retrieval using knowledge-intensive conversational CBR[A]. Proceedings of the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE2006)[C]. Berlin Heidelberg: Springer, 2006. 554-563.
[6] S C Bailin. Software reuse as ontology negotiation[A]. Proceedings of the 8th International Conference on Software Reuse (ICSR 2004)[C]. Berlin Heidelberg: Springer-Verlag, 2004. 242-253.
[7] U Reimer. Tutorial on organizational memories for capturing, sharing and utilizing knowledge[A]. Intemational Conference on Enterprise Information Systems (ICEIS 2001)[C]. Setubal, Portugal, 2001.
[8] S Bechhofer, et al. Owl Web Ontology Language Reference[DB/OL]. http://www.w3.org/TR/owl-ref/, 2004-02-10.
[9] 彭鑫, 赵文耘, 钱乐秋. 基于领域特征本体的构件语义描述和组装[J]. 电子学报, 2006, 34(12A): 2473-2477.
Pen Xin, Zhao Wenyun, Qian Leqiu. Semantic representation and composition of business components based on domain feature ontology[J]. ACTA ELECTRONICA SINICA, 2006, 34(12A): 2473-2477. (in Chinese)
[10] 李振东, 费翔林. 基于概念的信息检索模型研究[J]. 南京大学学报：自然科学版, 2002, 38(1): 99-109.
Li Zhendong, Fei Xianglin. Research on the concept-based information retrieval model[J]. Journal of Nanjing University: Natural Science, 2002, 38(1): 99-109. (in Chinese)
[11] D Lin. Dependency-based evaluation of MINIPAR[A]. Proceedings of the Workshop on the Evaluation of Parsing Systems[C]. Springer, 2003. 317-329.
[12] The Code Project. http://www.codeproject.com/[CP/OL]. 2007-07-28.
[13] Jena – A Semantic Web Framework for Java. http://jena.sourceforge.net/[CP/OL]. 2007-07-03.
[14] M.Paolucci, T.Kawamura, T.Payne, and K.Sycara. Semantic matching of web services capabilities[A]. Proceedings of the First International Semantic Web Conference[C]. Berlin Heidelberg: Springer, 2002. 333-347.
[15] Hwayoun Lee, Ho-jin Choi, In-Young Ko. A semantically-based software component selection mechanism for intelligent service robots[A]. 4th Mexican International Conference on Artificial Intelligence[C]. Berlin Heidelberg: Springer, 2005. 1042-1051.
[16] A Aamodt. Modeling the knowledge contents of CBR systems[A]. Proceedings of Workshop Program at the Fourth International Conference on Case-Based Reasoning[C]. Vancouver, 2001.

作者简介：

蔡怡峰　男，1983 年生于上海. 复旦大学计算机科学与工程系硕士. 研究方向为软件工程、构件技术. Email: 052021118@fudan.edu.cn

彭 鑫　男，1979 年生于湖北省黄冈市. 博士. 复旦大学计算机科学与工程系讲师. 研究方向为领域工程、软件产品线以及软件维护和再工程.

钱乐秋　男，复旦大学计算机科学与工程系教授，博士生导师. 研究方向为软件工程，软件复用，构件技术.

通信作者姓名：蔡怡峰，联系电话：021-55895879，Email：052021118@fudan.edu.cn，通信地址：上海市虹口区东体育会路 408 弄 2 号 304 室，邮编：200083