

文章编号:1001-9081(2010)01-0018-04

## 构件库语义描述和检索技术研究

牛志一<sup>1</sup>, 杨俊强<sup>2</sup>, 杨宁<sup>3</sup>

(1. 解放军理工大学 指挥自动化学院, 南京 210007; 2. 西安通信学院, 西安 710106;

3. 北京信息高技术研究所, 北京 100085)

(niuniu99414@163.com)

**摘要:**传统的构件库描述和检索方法无法对构件的语义关系进行描述,阻碍了用户对构件的应用。采用本体论的方法建立构件属性的描述模型,实现构件查询基于本体的语义扩展。给出构件属性与用户需求之间相似度的计算方法,帮助用户迅速准确找到需要的构件。

**关键词:**本体;语义;构件描述;构件检索;相似度

**中图分类号:** TP39 **文献标志码:** A

## Research on semantic description and retrieval of component database

NIU Zhi-yi<sup>1</sup>, YANG Jun-qiang<sup>2</sup>, YANG Ning<sup>3</sup>

(1. Institute of Command Automation, PLA University of Science and Technology, Nanjing Jiangsu 210007, China;

2. Xi'an Communication Academy, Xi'an Shaanxi 710106, China;

3. Beijing Institute of Advanced Information Technology, Beijing 100085, China)

**Abstract:** The traditional methods of description and retrieval of component database cannot describe the semantic relation of components, obstructing their application. Therefore, a description model of components was founded by means of ontology, in order to carry out semantic extension of component retrieval. The method that works out the similarity value between function of component and requirement of users was presented. It helps users to quickly and precisely find the components in need.

**Key words:** ontology; semantic; component description; component retrieval; similarity

## 0 引言

近年来随着计算机应用的迅速扩展,软件规模日益扩大,带来了软件复杂程度的增加和程序代码的几何级增长,最终导致软件开发成本增加,开发周期延长,产品质量下降,软件复用技术正是为了解决这些问题而出现。软件复用是指重复使用“为了复用目的而设计的软件”的过程。通过软件复用,在应用系统开发中可以充分利用已有的开发成果,提高了软件开发的效率。同时,通过复用高质量已有的开发成果,并且重新开发可能引入的错误,从而提高了软件的质量,降低软件开发和维护成本。

软件构件技术作为支持软件复用的核心技术,近几年来迅速发展并受到高度重视。基于构件的软件开发,是软件复用的一种实践方法,它在软件开发过程的各个阶段尽可能地利用可复用的构件,组装成新的应用软件系统。随着对软件复用应用研究的深入,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<sup>[1]</sup>。

构件的描述和检索是构件库研究的重点。传统的构件库描述和检索方法无法描述构件的语义信息和它们之间的关系,这使得对需求的表达程度完全依赖于用户的技能和经验,面对日益增长的构件数量和构件库规模,用户很难迅速准确找到所需要的构件,这大大妨碍了软件复用得进一步发展和应用。因此,本文提出了一种构件库语义描述和检索方法,利用本体模型对构件属性进行描述,充分体现构件的语义信息

和相互关系;在用户查询时,通过推理机制对用户需求的概念进行语义扩展和匹配。

## 1 国内外研究现状

目前,国内外对基于本体的构件库语义描述和检索的研究工作仍然处于起步阶段。在文献[2]中探讨了如何用语义网技术来对软构件进行检索和注册,提出了基于语的软构件语义检索框架;在文献[3]中,从本体元建模的角度,提出了一种软构注册的本体元模型,旨在从概念上综合已有的各种注册模型,从而实现多种基于互联网的软构件注册技术的互操作;在文献[4]中,提出了基于本体技术的构件信息集成层次架构,使用本体技术来描述构件信息,从而提高构件的检索效率,更好地满足客户需求;在文献[5]中,提出了用于程序挖掘的分布式构件库系统框架结构,将构件语义网络应用到该框架之中,并实现一个原型系统,能够实现基于语义导航的构件搜索。但是在当前的研究中,本体主要是利用在某个领域或者构件的某些术语,本体的语义完整性不足影响了本体应用的有效性,从而影响了构件检索的效率。另外,当前本体的语义检索在语义扩展方面做得比较简单,不能满足用户的检索需求。

## 2 构件库的语义描述模型

本体的概念最初起源于哲学领域,20世纪60年代人们开始将本体的概念和方法应用于计算机领域,用于知识表示、知识共享和知识重用。直观地讲,本体是一个实体,是对某领

收稿日期:2009-07-13;修回日期:2009-09-07。

**作者简介:**牛志一(1981-),男,河北秦皇岛人,博士研究生,主要研究方向:服务描述; 杨俊强(1981-),男,陕西西安人,博士研究生,主要研究方向:服务描述; 杨宁(1981-),男,河北秦皇岛人,助理工程师,主要研究方向:服务描述。

域应用本体论的方法分析、建模的结果,即把现实世界中的某个领域抽象为一组概念以及概念之间的关系。本体描述模型能明确地形式化定义术语的含义及术语间的关系,因此可以用来明确地定义构件服务的语义;有了明确定义的构件服务语义,就可以运用构件本体实现构件检索的推理扩展和结果排序,为用户需要提供语义上“匹配”的构件。

综合现有构件描述模型的内容,可将构件的属性分成功能无关属性和功能相关属性两部分。

### 2.1 功能无关属性

功能无关属性主要描述构件在产生和管理过程中生成和积累的信息,是由构件生产者和管理者提炼归纳的基本信息术语。由于是实践中长期积累的基本术语,所以没有严密的逻辑组织结构;因为进行了简要的归类而具有一定的层次性,但层次关系一般比较简单,不具有任何其他复杂关系和语义信息;这些属性不涉及构件功能信息的描述,也不受领域演变的影响而相对固定。因此,这些功能无关属性在构件本体中可以作为“构件”概念简单的数值属性,而无需为其增加新的对象属性和类。功能无关属性的具体内容如图1所示。

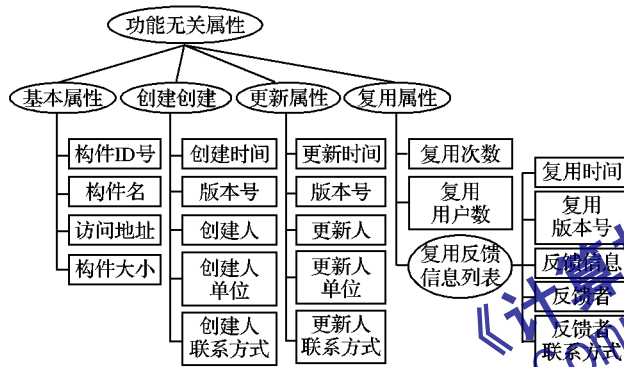


图1 功能无关属性

### 2.2 功能相关属性

功能相关属性是与构件实现的功能、服务相关的构件描述信息,通常用来帮助用户了解构件提供的服务等信息,是用户检索构件凭借的最主要因素,包括领域、功能、接口和环境等,具体内容如图2所示。

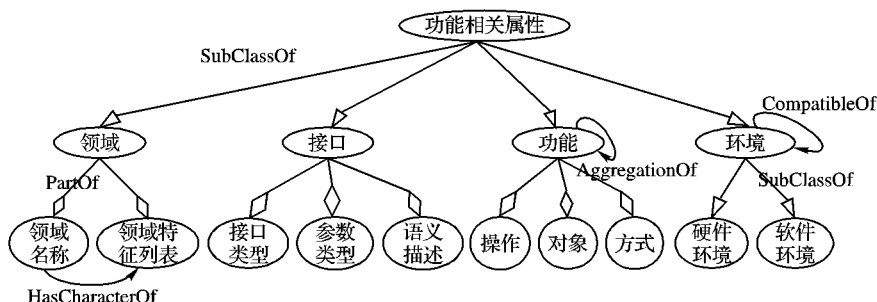


图2 功能相关属性

功能相关属性每一部分的内容都比较复杂,包含了丰富的语义信息和复杂的关系,其内容也会随着领域的演变和服务需求的变化而产生相应的变化,因此,在构件本体中必须为功能相关属性声明对应的类,通过和“构件”概念之间建立属性关系成为对象属性,从而满足对它们语义和关系描述的需要。

#### 2.2.1 领域属性

通常用户对构件需求的描述总是和他所在的领域密切相关,同一个描述词语在不同的领域中的意义、关系和逻辑常常会有很大差异,从而导致用户对需求的描述的区别,因此需要

对构件所属的领域进行描述,更精确地表达用户的需求。

对领域的描述包括领域名称和领域特征,领域名称确定了构件属于哪一个领域,要在其领域本体中完成语义推理。每一个领域都拥有一组相对应的与构件描述相关的领域特征,从而能够对本领域的构件进行更加深入细致的描述,例如“军事”领域具有“军兵种”、“业务”和“密级”等领域特征,构件这些特征的确认对用户需求的准确描述具有很重要的作用。

#### 2.2.2 接口属性

构件接口的描述对于构件功能实现和用户使用都具有重要的意义,模型中对接口的描述包括接口类型、参数类型和语义描述三个部分。其中接口类型用来区分是输入接口还是输出接口;参数类型表示这一接口的参数是何种类型,如数值、字符、字符串等,用户可以据此来定义和使用接口;语义描述是本体模型区别于传统描述方法最主要的方面,描述了接口具体的内容,例如“机票查询”构件,输入接口参数类型为字符串,语义描述可能为“航班号”,也可能是“出发地”和“目的地”的组合。

#### 2.2.3 功能属性

对构件功能的直接描述是用户选择构件的最主要依据,也是构件本体中最核心的内容。构件的功能的描述包括三个要素:操作、对象和方式。操作和对对象联合起来表述了构件“做什么”,而方式是对操作的进一步说明和限制。一般来说,对一个功能的描述必然具有操作,而对对象和方式有时是可以缺省的,例如“分布式计算”这一功能描述中,就只有操作“计算”和方式“分布式”。如果一个构件功能的操作、对象和方式都和用户的需求相吻合,就可以认为该构件能够满足用户的需求。

另外在本体模型中功能自身还具有聚合(Aggregation Of)关系,即某些功能可以通过另外一些功能以一定形式聚合在一起来实现,从而支持对构件功能多层次、多粒度的描述。

#### 2.2.4 环境属性

环境属性描述的是构件功能实现所需要的软件和硬件资源。软件环境主要包括操作系统、数据库、平台和容器四个方面;硬件环境是指构件完成功能所需的硬件的性能,分为硬件类型(CPU、内存等)、比较符和性能指标三个部分,即每一个硬件环境可以通过一个不等式来描述。环境属性具有兼容(CompatibleOf)关系。

## 3 构件库的语义扩展检索

传统的检索方法在处理查询时仅进行关键字级的概念匹配,很多和用户需求相关但表达不相同的概念就无法被检索到,因此,本文利用本体模型对构件查询中的概念进行语义扩展。

用户需求的概念集合对应到构件本体的语义网络,根据构件本体和领域本体中概念之间的相互关系,在语义层次上对用户需求的各个方面进行扩展,形成一个扩展的需求集合;将这一扩展的需求集合利用传统研究的匹配算法与构件属性进行匹配,选出符合条件的构件构成结果集合;用户的功能需求经过语义扩展之后,将大大增加符合条件的构件的数量,往往导致返回的结果集和过于庞大,使用户很难确定哪些构件

最适合自己的需求,因此根据匹配需求的程度对结果集中包含的构件进行排序,形成一个有序的结果集合,将最有价值的检索结果以一种用户容易识别和发现的形式表示,这也是传统方法的检索系统难以做到的一个方面。构件库语义扩展检索的主要过程如图3所示。

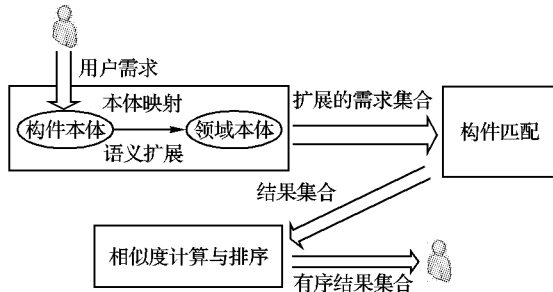


图3 构件库语义扩展检索

## 4 相似度计算

### 4.1 领域属性和功能属性相似度计算

本体中概念相似度计算的方法多种多样。较为常用的方法是将本体映射为一张树(或泛树)形图,图中的节点即本体中的概念,而边则是概念间的层次关系。两个概念间的相似度可用这两个概念对应的节点在该图中的最短路径长度和概念层次的深度来衡量,所以可由如式(1)<sup>[6]</sup>计算。

$$\text{sim}(c_1, c_2) = \begin{cases} e^{\alpha l} \times \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}, & c_1 \neq c_2 \\ 1, & \text{其他} \end{cases} \quad (1)$$

其中,  $\alpha \geq 0, \beta > 0$  为系数,  $l = \text{dist}(c_1, c_2)$  为两概念间的最小距离,  $h = \text{length}(\text{root}, \text{lcs})$  为  $c_1, c_2$  的最小公共父概念的深度。式中,左侧  $e^{-\alpha l}$  部分计算了概念间距离对相似度的影响,而右侧  $\frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$  部分计算了概念深度对相似度的影响。另外要注意的是,如果两个概念之间还有横向的关系(如等价关系、重叠关系)会对它们的最小距离产生影响,例如有等价关系,两者间的距离即为0。

计算领域相似度首先要判断领域名称是否相同,如果不相同说明需求和构件不属于同一领域,则领域相似度为0;若相同,再根据式(1)计算各个领域特征间的相似度,然后通过加权相加得到领域相似度。功能相似度也可以根据同样的公式计算得到。

### 4.2 接口属性相似度计算

一般来说每个构件都会具有多个输入和输出接口,因此可以采用基于二部图最佳匹配的方法<sup>[7]</sup>来计算接口属性的相似度。将用户需求描述  $Q$  中的输入或输出接口和构件属性描述  $A$  中的输入或输出参数分别映射为二部图的两部分,如图4所示。

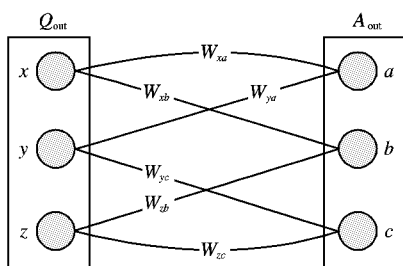


图4 输入输出接口的二部图匹配

基于本体模型的支持,使用式(1)计算每个节点与另一

部中其他节点间的相似度,作为二部图中边的权值,然后调用匈牙利算法计算加权二部图的最佳匹配,对选择出的最佳匹配将其权值归一化并相加,作为输入接口相似度  $\text{Sim}_{\text{input}}$  或输出接口相似度  $\text{Sim}_{\text{output}}$ 。

接口属性和用户需求的相似度为:

$$\text{Sim}_{VO} = \alpha \text{Sim}_{\text{input}} + \beta \text{Sim}_{\text{output}}$$

其中  $\alpha + \beta = 1$ , 且  $0 < \alpha < 1, 0 < \beta < 1$ 。

### 4.3 环境属性相似度计算

如前文所述构件的环境属性分为软件环境和硬件环境两部分。其中,软件环境的相似度可以通过和上文领域属性等相同的方法来计算。而硬件环境以不等式的形式存在,其相似度计算可转换为判断现有构件的硬件环境描述是否能满足用户需求。具体计算过程分为如下几个步骤:

1) 类型配对。根据硬件环境中的“硬件类型”对用户需求和构件硬件环境属性进行配对。

2) 单位转换。配对完毕后,需要进行单位转换,避免计量单位不一致造成的匹配错误。

3) 判断构件的硬件环境属性是否能够满足用户需求。构件的硬件环境属性与硬件环境需求都是由不等式表示,假设其表达式分别为  $A$  和  $B$ ,则判断约束需求是否能满足就等价于判断  $A \Rightarrow B$  是否成立。假设共有  $n$  条硬件环境需求,而硬件环境属性能够使其中的  $m$  条得到满足,则硬件环境相似度为  $m/n$ 。

最后,将领域相似度、功能相似度、接口相似度和环境相似度加权相加,即可得到构件相似度,如图5所示。

下面以一个具体的例子来说明构建相似度的计算过程。

假设一个用户需求  $A$  所属的领域为“军事”,军事领域的具有“军兵种”、“军事业务”和“密级”三个领域特征,其中  $A(\text{军兵种}) = \text{“陆军”}$ ,  $A(\text{军事业务}) = \text{“弹药保障”}$ ,  $A(\text{密级}) = \text{“机密”}$ ;

功能描述上  $A(\text{操作}) = \text{“查询”}$ ,  $A(\text{对象}) = \text{“信息”}$ ,  $A(\text{方式})$  为缺省;输入接口有两个,  $A(\text{输入1})(\text{参数类型}) = \text{“字符型”}$ ,  $A(\text{输入1})(\text{语义描述}) = \text{“弹药类型”}$ ,  $A(\text{输入2})(\text{参数类型}) = \text{“字符型”}$ ,  $A(\text{输入2})(\text{语义描述}) = \text{“弹药库编号”}$ ,输出接口有一个,  $A(\text{输出1})(\text{参数类型}) = \text{“数值型”}$ ,  $A(\text{输出1})(\text{语义描述}) = \text{“弹药库存”}$ ;硬件环境属性包括两条:“CPU 主频大于 1 GHz”和“内存容量大于 1 GB”;软件环境属性是  $A(\text{操作系统}) = \text{“Windows 98”}$ ,  $A(\text{数据库}) = \text{“Oracle”}$ 。

同时存在一个构件描述  $B$  也属于军事领域,  $B(\text{军兵种}) = \text{“海军”}$ ,  $B(\text{军事业务}) = \text{“交通运输保障”}$ ,  $B(\text{密级}) = \text{“秘密”}$ ;功能描述上  $B(\text{操作}) = \text{“查询”}$ ,  $B(\text{对象}) = \text{“信息”}$ ,  $B(\text{方式})$  为缺省;输入接口有一个,  $B(\text{输入1})(\text{参数类型}) = \text{“字符型”}$ ,  $B(\text{输入1})(\text{语义描述}) = \text{“油料类型”}$ ,输出接口有一个,  $B(\text{输出1})(\text{参数类型}) = \text{“数值型”}$ ,  $B(\text{输出1})(\text{语义描述}) = \text{“油料数量”}$ ;硬件环境属性包括两条:“CPU 主频大于 1.2 GHz”和“内存容量大于 512 MB”;软件环境属性是  $B(\text{操作系统}) = \text{“Windows XP”}$ ,  $B(\text{数据库}) = \text{“Oracle”}$ 。相关的部分本体语义网络如图6所示。

1) 计算领域相似度和功能相似度。

由于  $A(\text{军兵种}) = \text{“陆军”}$ ,  $B(\text{军兵种}) = \text{“海军”}$ , 所以两概念间的最小距离  $l = 2$ , 最小公共父概念的深度  $h = 0$ , 根据式(1), 设  $\alpha = \beta = 0.5$ , 可求得  $A, B$  间“军兵种”属性的相似度  $Sim(A, B)(\text{军兵种}) = 0$ , 根据同样方法可以求得

$Sim(A, B)(\text{军事业务}) = 0.103$ ,  $Sim(A, B)(\text{密级}) = 0.280$ 。平均分配权重, 可求得  $Sim(A, B)(\text{领域}) = 0.128$ ; 功能相似度的计算方法与领域相似度完全相同, 本例中  $A$  和  $B$  功能属性的值完全相同, 所以  $Sim(A, B)(\text{功能}) = 1$ 。

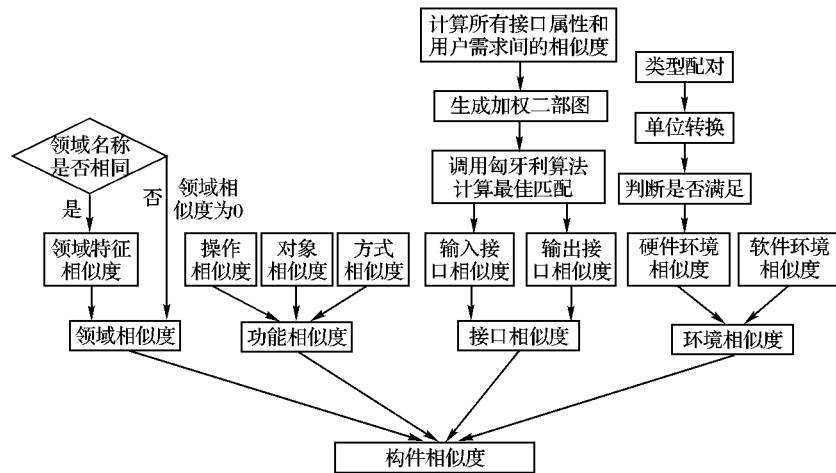


图5 构件相似度计算

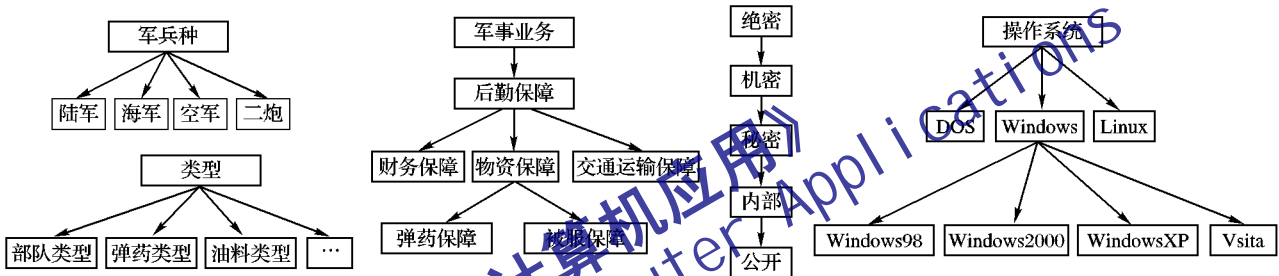


图6 部分本体语义网络

## 2) 计算接口相似度。

在  $\alpha = \beta = 0.5$ , 平均范围分配权重的情况下, 根据式(1), 可得到二部图匹配如图7所示。

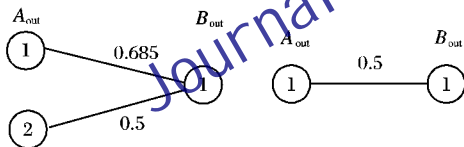


图7 输入输出接口的二部图

根据匈牙利算法得到最佳匹配可计算  $Sim(A, B)(\text{输入接口}) = 0.685$ ,  $Sim(A, B)(\text{输出接口}) = 0.5$ 。那么平均分配权重, 可得  $Sim(A, B)(\text{接口}) = 0.593$ 。

## 3) 计算环境相似度。

根据  $A$  和  $B$  的硬件环境描述可知  $A$  有两条硬件环境需求, 而  $B$  的硬件环境描述可以满足其中一条, 所以  $Sim(A, B)(\text{硬件环境}) = 1/2 = 0.5$ ; 在  $\alpha = \beta = 0.5$ , 平均范围分配权重的情况下, 根据式(1) 可计算得到  $Sim(A, B)(\text{操作系统}) = 0.170$ ,  $Sim(A, B)(\text{数据库}) = 1$ , 那么  $Sim(A, B)(\text{软件环境}) = 0.585$ ; 则  $Sim(A, B)(\text{环境}) = 0.543$ 。最后, 仍然是平均分配权重,  $A$  与  $B$  之间的相似度  $Sim(A, B) = (0.128 + 1 + 0.593 + 0.543) = 0.566$ 。

## 5 结语

随着软件复用和构件技术的迅速发展, 大量构件被生产出来, 构件库的规模越来越大。传统的构件库描述和检索方

法已经不能满足用户的需求。本文通过建立构件库语义本体模型, 对构件属性的语义和关系进行描述, 在此基础上提出了构件库的语义扩展检索和相似度计算方法, 帮助用户全面准确找到所需要的构件, 为软件复用技术的进一步发展和应用奠定基础。

## 参考文献:

- [1] 赵俊峰. 软件构件标准概述[J]. 信息技术与标准化, 2006(6): 10-13.
- [2] YAO HAINING, ETZKORN L. Towards a semantic-based approach for software reusable component classification and retrieval[C]// Proceedings of the 42nd Annual Southeast Regional Conference. New York: ACM, 2004: 110-115.
- [3] 范辉华. 软构件注册的本体元模型研究[D]. 武汉: 武汉大学, 2003.
- [4] 周拥峰. 基于本体的构件检索架构研究[D]. 上海: 复旦大学, 2003.
- [5] 贾成, 陈松乔, 王斌. 基于构件语义网络的分布式构件库原型系统[J]. 计算机工程, 2005, 31(5): 117-119.
- [6] LI YUHUA, BANDAR Z, McLEAN D. An approach for measuring semantic similarity between words using multiple information sources[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 871-882.
- [7] BELLUR U, KULKARNI R. Improved matchmaking algorithm for semantic Web services based on bipartite graph matching[C]// IEEE International Conference on Web Services. New York: IEEE, 2007: 86-93.