

基于本体的构件功能检索的设计与实现

闫伟,曹宝香,张永红,李福盛

(曲阜师范大学 计算机科学学院,山东 日照 276826)

摘要:目前,基于构件的软件开发已经成为软件工程研究领域和软件开发实践过程中越来越重要的内容。在构件的描述与检索中引入本体,可以更好地表示构件间的语义信息,同时使得构件过程描述与检索成为可能。在研究基于本体的构件描述基础上,设计并实现了构件功能检索过程,为使用构件进行软件开发的用户提供了一种按照所需功能进行构件检索的方法,同时,由于利用知识语义上的匹配,为查全率和查准率提供了更好的保证。

关键词:本体;功能检索;OWL;Jena推理

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2009)05-0114-05

Design and Implementation of Ontology - Based Component Function Retrieval

YAN Wei, CAO Bao-xiang, ZHANG Yong-hong, LI Fu-sheng

(Computer Science College, Qufu Normal University, Rizhao 276826, China)

Abstract: At present, CBSD(component - based software development) is becoming more and more important in software engineering research and software development practice. By introducing ontology into the field of software - component description and retrieval, the semantic information among the components can be presented more precisely, and at the same time the component represent and retrieval using process property can also be possible. On the basis of ontology - based component represent, designs and implements the process of component functional retrieval, which provides a method of component retrieval in accordance with functions in need. It also improves efficiency and quality in component retrieval since the basis is the matches of knowledge semantics.

Key words: ontology; functional retrieval; OWL; Jena reasoning

0 引言

软件构件化是软件复用的必由之路,构件描述与检索是构件入库、组装的前提条件,基于刻面的构件描述对于反映构件静态信息已经有比较成功的应用,但由于缺乏相关语义信息及动态性,而无法利用推理机制实现具有不同需求的构件检索。将本体引入构件描述,可以表示丰富的语义信息,并通过推理机制,得到隐含的语义信息。在构件中引入本体,可以实现高效的面向服务的构件检索^[1]。

文中所设计并实现的基于本体的构件功能检索,首先创建大量构件的本体库,然后由 Jena 推理机根据所设计的推理规则进行推理。用户只需输入查询的构件功能,就可以得到构件库中具有该功能的所有构件;

再者,用户可以检索某一构件的全部功能;与普通的构件 IOPE 检索相比,还具有较高的查全率和查准率。

1 背景知识

1.1 本体

本体(Ontology)原本是一个哲学中的概念,被哲学家用来描述事物的本质。在计算机领域,Gruber 关于本体的定义得到了业内同行的一致认同,即“本体是概念模型的明确的规范说明”。

本体作为一种能在语义和知识层次上描述信息的概念模型建模工具,在信息检索,特别是基于知识的信息检索中得到广泛应用。基于本体的方法可以描述构件的内在知识及其语义信息,提供检索的知识匹配和推理能力,该方法已经成为构件开发领域最新的研究方向^[2]。同时,如何利用本体知识来描述构件库中的构件,对用户的构件检索需求进行描述,如何高效、准确地从构件库中找出满足用户各种需求的构件,是文中的主要研究内容。

收稿日期:2008-09-01

基金项目:国家自然科学基金项目(60072014);山东省自然科学基金项目(Y2003G01)

作者简介:闫伟(1984-),女,山东济宁人,硕士研究生,研究方向为企业信息化与系统集成、数据挖掘和人工智能。

1.2 OWL

OWL(Web Ontology Language)是W3C组织于2004年2月正式推出的一个专门用来描述本体的语言^[3]。它既有强大的语义表达能力,又可以实现描述逻辑的可判定推理。OWL可以用来存储本体中的概念和关系,按照表达能力由弱到强的顺序分为OWL Lite、OWL DL和OWL Full。OWL本体的最大优点是能有效地支持推理,并且有许多能够对其做推理的工具,这些工具提供了不局限于某个主题领域的通用支持。

1.3 Jena 推理

Jena是由HP实验室开发专门用来构建语义网的应用软件,它为RDF、RDFS和OWL提供了一个可编程实现的环境。Jena是面向语义Web的应用开发包,包含的内容非常全面,推理机只是其中的一部分。Jena提供的推理机也和RACER、FaCT、Pellet等一样,是针对本体的推理机,但是Jena本身并不是“推理机设计专家”,它自身包含的推理机是一种CLISP配合本体领域产生式规则的前向推理系统^[4]。

Jena最大的优点是面向具体的本体语言,针对性强,效率高。文中所设计的基于本体的构件功能检索,运用OWL-DL语言描述构件,形成构件本体库,根据用户功能需求并结合所设计的推理规则,利用Jena推理机进行推理,共同实现高效、准确的构件功能检索。

1.4 功能检索

当前,信息检索技术主要分为三类:全文检索(Text Retrieval)、数据检索(Data Retrieval)和知识检索(Knowledge Retrieval)。全文检索和数据检索的本质是直接基于关键字的检索技术,不能满足用户在语义和知识上的需求。而知识检索强调基于知识的语义上的匹配,因此在查全率和查准率上有更好的保证。

文中所设计的功能检索属于知识检索,它以构件本体库作为知识库,结合设计的推理规则,利用Jena推理机进行各种推理,以得出准确的检索结果。

2 基于本体的构件功能检索

在文中所设计并实现的构件功能检索中,将以OWL-DL作为本体描述语言构建构件本体库,再利用Jena推理机制,并结合所设计的推理规则进行推理。OWL构件本体主要包括按构件级别和功能划分的类(Class)、类的实例(Instance)、表达各个类实例之间和类实例与RDF文字或XML Schema数据类型之间关系的属性(Property)^[5]。

利用OWL本体描述构件的目的是可以利用它进行关于个体的推理,为了在一种有效的方式下做到这

一点,需要一种机制来描述个体所属的类以及这些个体通过类成员关系而继承得到的属性,虽然可以为个体声明特定的属性,但是本体的大部分推理能力在于基于类的推理,因此如何有效规划类的推理是整个推理过程能够顺利、有效进行的关键。

在文中设计的构件功能检索过程中,将构件领域基本概念,即不同构件级别的构件作为各个分类层次树的根,各个类都隐含地是owl:Thing的一个子类,按此方法将创建十个分类层次树的根。然后,对每个分类层次树的根再按照功能划分,生成对应树根的子类,例如,对应二级构件,有注册和登录功能的划分子类。创建的功能子类是作为分类层次树的第一层节点,在此基础上,建立对应于各个功能子类的个体,定义类以及个体间的关系,这一操作主要是通过属性的建立完成。

经过以上步骤,构件本体库基本创建完毕,并且作为知识推理的知识库,结合设计的推理规则和Jena推理机制,完成构件功能检索的推理过程。

图1是构件本体中各组成部分及与推理机关系。

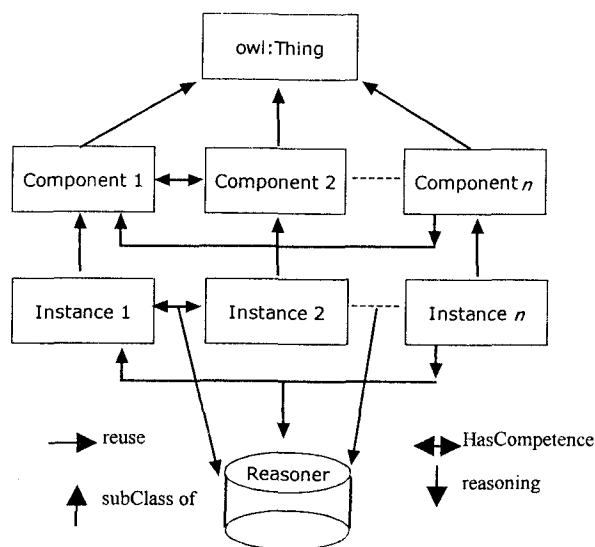


图1 构件本体各组成部分及与推理机关系图

2.1 构件描述类别划分

根据构件表示,构件描述方法分为人工智能方法、超文本方法和信息科学方法,其中,基于信息科学方法的构件描述方法在实际构件库开发的项目中应用较为成功,它又分为基于受控词汇表(如枚举、刻面分类)和基于非受控词汇表(如关键词分类)两种形式。

文中采用枚举分类方法来描述构件,这种分类方法通常将一个被关注的领域严格划分为若干不相关的子领域,构件按照某些性质分为若干大类,每个大类又可分为若干小类,经过若干次分解形成构件的层次结构,而实际的构件位于层次结构的最低层,其他层次则

表示构件所属的父类或者祖先类。该方法对问题域进行了清晰的高度结构化的划分,概念清晰,易于理解和使用。

文中所设计的构件功能检索过程将构件分为十个级别,级别越高,构件粒度越大,构件描述越抽象。一级构件形态为:

$$\langle \%C_x(P_1, P_2, P_3, \dots, P_n)\% \rangle$$

其中, C_x 为构件名称, 由字母、数字和下划线组成, 区分大小写。 $P_1, P_2, P_3, \dots, P_n$ 为构件输入参数, 如显示状态(大小、颜色、位置)、SQL 语句、连接参数、传递数据等。例如, 以下是一级构件输入框的形态表示:

$$\langle \% \text{Input}(\text{name}, \text{left}, \text{top}, \text{length}, \text{value}) \% \rangle$$

从二级构件到十级构件, 在功能上分别代表表单级、过程级、用况级、业务级、业务域级、应用域级、系统级、系统群级和分布式系统群级构件。

2.2 创建构件本体库

构建构件本体库主要包括创建表示不同构件级别和构件功能的子类及个体, 建立反应构件个体实例之间以及构件个体与 XML Schema 数据类型之间关系的各个属性, 例如, 三级构件某个体与它所复用的相关功能的二级构件某个体所建立的复用属性, 该属性为对象属性; 而某二级构件的数据类型属性 Content 则表示该构件的形式化描述, 并且对应于 XML Schema 中的 xsd:String 类型。

2.2.1 构建构件 OWL 类及个体

根据构件描述的不同类别, 按照纵向和横向两个角度创建二维构件本体库。在纵向维度上, 为十个级别构件分别创建十个 owl: Thing 的子类, 分为 FirstComponent, SecondComponent, ThirdComponent, ..., TenthComponent, 其中每个子类按照功能创建下一级子类, 例如在 SecondComponent 中分别创建 RegisterComponent2 和 LoginComponent2, 分别代表在二级构件中, 完善注册和登录功能的构件, 它们是在一定程度上复用一级相关功能构件的结果。在横向维度上, 对已经建立的各个不同级别不同功能的子类, 创建互异的个体实例, 例如, 对于二级构件的完善注册功能的 RegisterComponent2 子类, 扩展其外延, 创建三个互异的个体, 分别是 RegisterInstance21, RegisterInstance22 和 RegisterInstance23, 分别表示在不同程度上完善注册功能的互异的个体实例。

由于文中所设计的是基于功能的构件查询, 因此在设计各类的个体实例时是由初具功能的二级构件, 即表单级构件开始创建, 在实际应用中, 可以根据需要添加一级构件中的个体实例。

同时, 为了构建推理知识库, 还需要创建 owl: Thing 的子类 Function 及个体用来表示构件具有的功能, 例如, 某构件个体具有 register 属性值表示该构件具有注册功能。

构建后的各类关系图如图 2 所示。

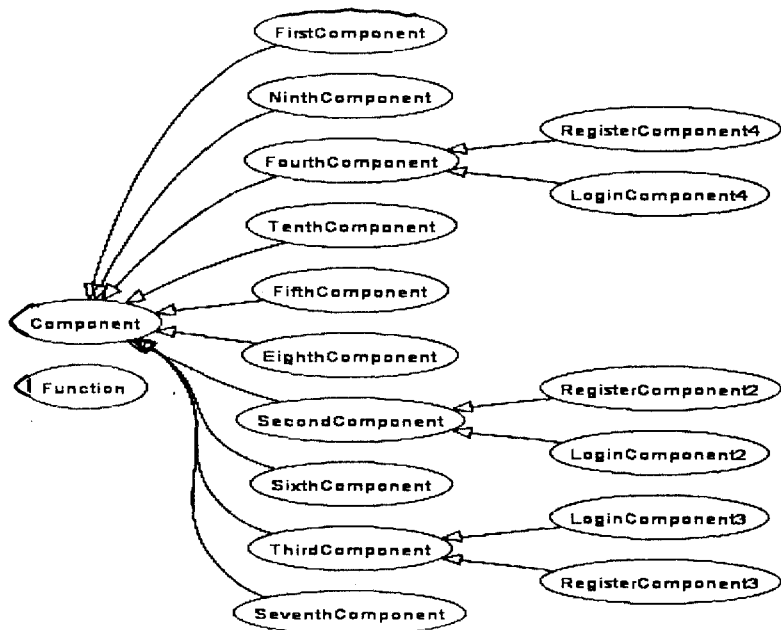


图 2 构件本体库各类关系图

2.2.2 建立 OWL 属性

OWL 属性分为对象属性(Object Properties)和数据类型属性(Datatype Properties)两种, 其中, 对象属性表示两个类的类实例之间的关系, 而数据类型属性则表示的是类实例与 RDF 文字或者 XML Schema 数据类型间的关系^[6]。

在文中所设计的构件功能检索过程中, 主要建立了两个对象属性, 它们分别是 reuse 属性和 hasCompetence 属性, 表示“…复用…”和“…具有…功能”。

根据构件个体间的复用关系, reuse 属性所表示的构件关系属于多对多关系, 即允许一个构件个体复用多个构件个体, 一个构件也可以同时被多个构件复用。与常见的表示单一继承的 inherit 属性相比, reuse 属性定义的多对多关系更能反应各级构件间复杂的复用关系, 为实现具有多种功能的构件检索提供了推理依据。将 reuse 属性设置为传递类型(transitive), 同时, 要求参与属性传递的各个构件个体均属于同一“纵坐标”, 即各个构件个体传递的是同一功能。这些个体还可以同时具有其他功能, 但是必须存在同一“交叉功能”。

reuse 属性对应的规则表示为:

Rule (? a reuse? b), (? b reuse? c) - > (? a reuse? c)

其中, ? a 代表具体的构件个体,这一性质的设定为实现后面关于功能的语义推理奠定了基础。

对象属性 hasCompetence 表示某个构件具有某种功能,它是构件功能检索推理的主体。在实际推理过程中,hasCompetence 属性映射到具体的功能值。

对于数据类型属性,文中主要建立了表示构件具体信息的属言,如 Author、Time 和 Title 等。

2.3 构造推理规则

文中所设计的构件功能检索主要实现以下功能:第一,根据用户输入所需功能,检索具有该功能的构件个体,并给出相关信息;第二,用户输入构件名称,能够检索出其所具备的所有功能,同时给出相关信息。

根据以上功能要求,设计如下规则以实现检索功能:

Rule: (? x hasCompetence ? y), (? z reuse ? x), not Equal(? x, ? z) - > (? z hasCompetence ? y)

Rule 规则说明,如果某个构件个体 x 具有 y 功能,而构件个体 z 复用了 x,并且 x 和 z 是不同的构件个体,则 z 同样具有 y 功能。

将以上规则放入 Jena 推理机,为下一步推理提供了依据。

2.4 编程实现推理过程

在构件本体库和推理规则构建完毕后,即可利用 Java 语言,编写程序调用 Jena 的 ModelFactory 来访问其推理机制,并为相关联的 Reasoner 建立一个新的 Model^[7]。查询这个新的 Model 不但会返回对原来数据的描述信息,还会返回通过 Reasoner 中的规则执行后的附加描述信息^[8]。整个推理机制的核心部分是 InfGraph,因为所有应用的执行都是在 Graph SPI 层进行。Ont/Model API 为用户构建的本体提供了一种便利的方式与合适的推理器连接^[9]。推理结构中的 Graph-base assertions 指的是和 Reasoner 一起绑定的数据(XML 数据-实例),而 Graph-ontology definitions 指的是和 Reasoner 一起绑定的数据结构的限制^[10](也就是 XML 模式-概念和关系)。Reasoner Registry 是一个静态类,它包含了当前用到的所有 Reasoner。Jena 推理机结构如图 3 所示。

其实现代码如下:

```
InfModel infModel;
String file = "E://jena-component//ontology//Component.owl";
Model data = ModelFactory.createDefaultModel();
```

```
Model model = ModelFactory.createDefaultModel();
InputStream in;
try{
    in = FileManager.get().open(file);
    data.read(in, "");
}
catch (Exception e){

}

Resource configuration = model.createResource();
configuration.addProperty(ReasonerVocabulary.PROPRuleMode,
"forward");
configuration.addProperty(ReasonerVocabulary.PROPRuleSet,
"E://jena-component//rules//Component.rules");
Reasoner reasoner = GenericRuleReasonerFactory.getInstance().
create(configuration);
infModel = ModelFactory.createInfModel(reasoner, data);
```

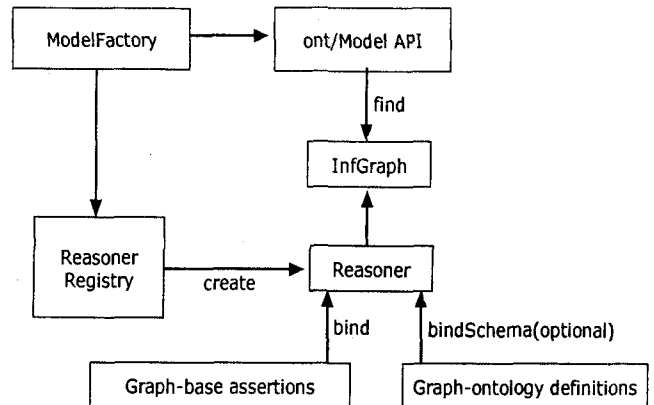


图3 Jena 推理机结构

上面操作输出结果的形式化描述为(以查询 Register 功能为例):

```
-(eg: RegisterInstance21 hasCompetence eg: Register)//构件个体 RegisterInstance21 具有 Register 功能
-(eg: RegisterInstance31 reuse eg: RegisterInstance - 21)//构件个体 RegisterInstance31 复用 RegisterInstance - nce21
-(eg: RegisterInstance31 hasCompetence eg: Register)//构件个体 RegisterInstance31 具有 Register 功能
```

在文中所实现的构件功能检索演示系统中,用户可以选择检索项进行查询,除了可以选择构件功能,还可以选择构件名称、构件创建者和构件版本进行检索。

图 4 为用户选择 register 注册功能系统显示检索构件的截图。

3 结束语

文中所设计并实现的基于本体的构件功能检索过程,将本体应用于构件描述,构建了关于构件的本体库以作为推理的知识库,同时应用本体强大的语义表述和推理能力,设计推理规则,结合 Jena 推理机制,实

现高效、高速的构件功能检索过程。同时,研究过程中也发现,基于本体的构件检索,后续工作还有很多,例如本体表示汉化能力,以及对复杂数据结构如 Functor 的推理等。

| 构件功能检索演示系统 Powered by yw | | |
|-----------------------------|-----------|---|
| 检索 | | |
| 检索结果 | | |
| 构件信息 | | |
| | 检索项: 构件功能 | 检索值: register |
| 构件名称 | 构件版本 | 主要功能 |
| RegisterInstance42 | v2.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register |
| RegisterInstance23 | v1.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register |
| RegisterInstance32 | v2.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register |
| RegisterInstance31 | v2.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register |
| RegisterInstance21 | v1.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register |
| RegisterAndLoginInstance4 | v3.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register http://www.owl-ontologies.com/ComponentOntology.owl#login |
| RegisterInstance22 | v1.0 | http://www.owl-ontologies.com/ComponentOntology.owl#register |

图 4 具有 register 功能的构件列表

参考文献:

- [1] Gruber T R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing[J]. Information Journal of Human Computer Studies, 1995, 43: 907-928.
- [2] Gibb F, McCartan C, O'Donnell R, et al. The Integration of Information Retrieval Techniques within a Software Reuse Environment[J]. Journal of Information Science, 2000, 26(4): 520-539.
- [3] 徐正权. 软件复用方法与技术[M]. 武汉: 华中理工大学出版社, 1998.
- [4] Musen M A. Dimensions of Knowledge Sharing and Reuse[J]. Computers and Biomedical Research, 1992, 25: 435-467.
- [5] Stuckenschmidt H. Ontology-Based Information Sharing in Weakly Structured Environments [D]. Amsterdam: AI Department, Vrije Universiteit Amsterdam, 2002.
- [6] Doan A, Madhavan J, Domingos P, et al. Learning to map between ontologies on the Semantic Web [C]//In 11th International World Wide Web Conference. Honolulu, USA: [s. n.], 2002.
- [7] Baader F, Sattler U. Description logics with aggregates and concrete domains[J]. Information Systems, 2003, 28(8): 979-1004.
- [8] Zoe L. Web data retrieval and extraction[J]. Data and Knowledge Engineering, 2003, 44(3): 347-367.
- [9] 李选如, 何洁月. 语义集成: 本体映射方法研究[J]. 计算机技术与发展, 2007, 17(2): 127-130.
- [10] 黄烟波, 张红宇, 李建华, 等. 本体映射方法研究[J]. 计算机工程与应用, 2005(18): 33-35.
- [11] 李雄飞, 李军. 数据挖掘与知识发现[M]. 北京: 高等教育出版社, 2003: 11-19.
- [12] 毛国君. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2005: 171-175.
- [13] 朱明. 数据挖掘[M]. 合肥: 中国科学技术大学出版社, 2002: 22-32.
- [14] 李秀芳, 李志成. 基于数据挖掘的聚类算法研究[J]. 计算机技术与自动化, 2006(3): 1-2.
- [15] 段明秀, 杨路明. 对层次聚类算法的改进[J]. 湖南理工大学学报, 2008(2): 1-2.

(上接第 110 页)

- [1] Proceedings of the SIGCOMM Workshop on Delay Tolerant Networking. [s. l.]: [s. n.], 2005.
- [2] Leguay J, Friedman T, Conan V. DTN Routing in a Mobility Pattern Space[C]//In proc. ACM SIGCOMM 05 Workshop on Delay Tolerant Networking and Related Topics (WDTN-05). [s. l.]: [s. n.], 2005.
- [3] Mukarram M, Tariq B, Ammar M H, et al. Message ferry route design for sparse ad hoc networks with mobile nodes[C]//Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing. [s. l.]: [s. n.], 2006: 37-48.
- [4] 王行甫. 一种 DTN 节点自定位方案及其性能分析[J]. 中国科学院研究生院学报, 2008, 25(3): 367-371.
- [5] Cook W, Applegate D, Bixby R, et al. Concorde: A code for solution of Travelling Salesman problem [J/OL]. 2005. http://www.tsp.gatech.edu/.

(上接第 113 页)

- [1] trieval. [s. l.]: ACM Press, 1996: 76-80.
- [2] 贺玲, 吴玲达, 蔡益朝. 数据挖掘中的聚类算法综述[J]. 计算机应用研究, 2007(1): 10-13.
- [3] 谢立宏. 面向高维数据的聚类算法研究[D]. 武汉: 武汉大学, 2002: 10-12, 29-30.
- [4] Makhoul J, Roucos S, Gish H. Vector quantization in speech coding[J]. Proc. of the IEEE, 1985, 7: 1551-1556.
- [5] Karyapis G, Han E H, Kumar V. Chameleon: A hierarchical clustering algorithm using dynamic modeling[J]. IEEE Computer, 1999, 32(8): 68-71.
- [6] 李雄飞, 李军. 数据挖掘与知识发现[M]. 北京: 高等教育出版社, 2003: 11-19.
- [7] 毛国君. 数据挖掘原理与算法[M]. 北京: 清华大学出版社, 2005: 171-175.
- [8] 朱明. 数据挖掘[M]. 合肥: 中国科学技术大学出版社, 2002: 22-32.
- [9] 李秀芳, 李志成. 基于数据挖掘的聚类算法研究[J]. 计算机技术与自动化, 2006(3): 1-2.
- [10] 段明秀, 杨路明. 对层次聚类算法的改进[J]. 湖南理工大学学报, 2008(2): 1-2.