

# 基于 SQWRL 的本体知识库语义查询

王 飞,张应中,罗晓芳

(大连理工大学 机械工程学院,辽宁 大连 116024)

**摘 要:**本体技术是知识工程领域的研究热点,本体对于领域知识具有强大的语义描述能力和推理能力,能够很好地被计算机识别和理解,从而可在语义层面实现知识共享、重用和扩展。本体技术在形式化、开放性和共享性等方面比其他知识建模技术能提供更强大的能力。介绍了基于本体的知识库的概念以及基本构架;描述了本体知识库查询的语言特点,重点阐述了 SWRL 规则以及 SQWRL 语义查询;采用语义网规则语言(SWRL)将材料选择本体中的因果知识表达为一组知识规则,增强本体知识库的推理能力;采用 Java 编程技术调用 SWRL API 和 OWL API,将知识库层、用户界面层及推理层三大模块建成一个完整的语义查询系统,并使用基于描述逻辑的 Drools 规则推理机对本体进行规则推理。给出了一个基于 SQWRL 语义查询的工程材料领域本体知识查询实例进行说明。

**关键词:**本体;知识库;SWRL 规则;SQWRL;语义网;语义查询

**中图分类号:**TP182

**文献标识码:**A

**文章编号:**1673-629X(2017)02-0015-05

**doi:**10.3969/j.issn.1673-629X.2017.02.004

## Semantic Query of Ontology Knowledge Base Based on SQWRL

WANG Fei,ZHANG Ying-zhong,LUO Xiao-fang

(School of Mechanical Engineering,Dalian University of Technology,Dalian 116024,China)

**Abstract:** Ontology technology is a hot research topic in the field of knowledge engineering. Ontology has a powerful semantic description and reasoning ability for domain knowledge, which can be well recognized and understood by computers so as to realize the knowledge sharing, reusing and extension in semantic level. The ontology technology can provide more powerful abilities than other knowledge modeling techniques in the formalism, openness and sharing. The basic concept and framework of the ontology based knowledge base is introduced, and the characteristics of language in ontology knowledge base is described, especially for SWRL and SQWRL. The causal knowledge for material selection is represented as a set of knowledge rules coded with Semantic Web Rule Language (SWRL), which enhance the reasoning ability of the ontology knowledge base. A Java programming technology which uses SWRL API and OWL API is employed, which integrates the knowledge base layer, the inference layer, and the user interface layer, and builds a overall material selection knowledge system. The Drools rule reasoning machine is employed to implement the ontology reasoning, and a semantic query example based on SQWRL for engineering material domain knowledge is presented.

**Key words:** ontology; knowledge base; SWRL rule; SQWRL; semantic web; semantic query

## 0 引 言

随着语义网的发展,语义 Web 使互联网成为全球化知识共享平台。海量的信息与知识资源为人们提供了认识世界的前提,知识是与具体领域相关的,知识的表达与组织是领域知识面临的挑战与机遇,也是知识发现和挖掘的基础,如何从显式表达的知识中有效地获得人们想要的隐含信息成为知识表示研究的重点。

本体是对领域知识的一种概念化的显示表达,是一种描述信息的概念模型建模工具,具有十分强大的表达能力和对逻辑推理的支持。语义检索是获得本体领域蕴含知识的方式,只要将需要的隐含信息查询出来才能便于人们更加深入地识别和理解知识。

目前,本体查询方式有:基于描述逻辑(Description Logic, DL)查询、基于 RDF(Resource Description

收稿日期:2016-03-21

修回日期:2016-06-13

网络出版时间:2017-01-04

基金项目:国家自然科学基金资助项目(51375069)

作者简介:王 飞(1989-),男,硕士研究生,研究方向为基于本体的知识工程、语义 Web;张应中,硕士生导师,研究方向为基于本体的语义信息技术及工程应用、产品三维数字建模;罗晓芳,硕士生导师,研究方向为计算机图形图像处理、概念设计、基于网络的协同设计、智能 CAD。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20170104.1102.096.html>

Framework)三元组查询,即 SPARQL (Simple Protocol and RDF Query Language)、基于 OWL 语义查询,即 SQWRL<sup>[1]</sup> (Semantic Query-enhanced Web Rule Language)。然而基于描述逻辑的查询语言,包括 OWL-QL 和 DIG's ASK protocol,由于它们表达意义不够而不能很好地作为 OWL 查询语言;SPARQL 是标准的 RDF 查询语言,但它不能够与知识规则结合。OWL 不能表示“IF-THEN”产生式知识规则,仅靠 OWL,其知识表示的能力有限。因此,需要采用语义规则语言 (Semantic Web Rule Language, SWRL) 定义因果关系知识。SWRL 能够很好地将 OWL 概念和关系与产生式规则相结合,形成强大的知识表达能力。SQWRL 是基于 SWRL 规则的查询语言,能够很好地兼容 SWRL 规则并能够存储在 OWL 本体中,并能进行语义查询。

近年来,对本体及其相关应用方面的研究一直是研究热点,特别是本体的语义查询、知识检索技术。张艳涛等<sup>[2]</sup>阐述了利用图像本体和 SWRL 规则来实现图像情感识别,在 Protégé 上进行规则推理,引入 Jess 推理引擎,获得图像隐性属性信息;张宗仁等<sup>[3]</sup>使用 SPARQL 进行三元组查询,使用 Jena 进行读取和推理,但是查询过于局限;王海等<sup>[4]</sup>使用 SQWRL 语言进行 Web 语义查询,对 OWL-S 形式的服务进行发现,初次探索了 SQWRL 语言等,但是都不够系统。文中在进行大量研究之后,建立了一种通用的基于 SQWRL 的本体语义查询框架,可以针对本体知识库进行语义查询,充分展现了 SQWRL 语言的优势,通过程序进行外部控制本体,这就给本体应用提供了开放式、共享服务,而且便于扩展<sup>[5]</sup>。

## 1 基于本体的知识库

### 1.1 知识的概念表示

本体最早来源于哲学概念,是共享概念模型的确的形式化的规范说明<sup>[6]</sup>,在知识工程领域,知识可以通过概念及其关系表示。本体中知识概念是指某领域的相关概念,以及概念与概念之间的关系。概念又称为类,概念与概念之间通过属性产生关系,通过 OWL<sup>[7-8]</sup>描述语言对本体进行形式化描述,使得计算机能够理解本体描述的知识。

本体知识就是关于某一领域概念的一套规则的清晰的描述,包括基本的概念,概念间的关系,概念的相关属性,属性的限制条件。因此本体清楚地表达领域知识中概念的精确含义以及概念之间的关系。通过对领域知识中的一般概念以及概念间的关系形成概念集和关系集,然后不断完善应用领域知识,最后形成 TBox (Terminology Box),即概念和关系的集合,如下:

$$TBox = \{a_1^p, a_2^p, \dots, a_m^p, a_1', a_2', \dots, a_n'\}$$

其中, $a$  表示概念或术语; $m$  表示属性; $p$  表示关系。

### 1.2 基于 SWRL 的知识规则

SWRL 是万维网联盟 (W3C) 于 2004 年提出,用于描述推理规则。它是以 OWL 子语言 OWL DL、OWL Lite 和 RuleML<sup>[9]</sup> 为基础的规则描述语言。最终是为了能让规则与 OWL 结合起来,以提高本体的推理能力和逻辑表达能力。

SWRL 规则的基本成分是 Atom,包括七种 Atom 类型(类、实例属性、数据属性、不同实例原子、相同实例原子、内置式原子、数据值域原子),并且支持开放式假设和 OWL 类表达式。内置式原子 (Built-in) 包含了规则中可以自由使用的逻辑比较关系,还可以通过开放式假设在本体中增加新的概念定义,作为本体的中间概念,提升本体的推理能力并且扩展了本体知识库。SWRL 规则表示形式如下:

$$Atom \wedge Atom \wedge \dots \rightarrow Atom \wedge Atom$$

其中,Atom 表示  $p(\arg_1, \arg_2, \dots, \arg_n)$ ,  $p$  代表类、属性或者数据断言符,  $\arg_1, \arg_2, \dots, \arg_n$  代表实例个体、数据或者变量参数。

SWRL 中的内部模块 Built-in,包含了规则中可以使用的逻辑关系,例如:材料 (materials) 本体碳钢的碳含量低于 0.25%,称为低碳钢,那么定义规则如下:

$$\text{CarbonSteel} (?c) \wedge \text{hasProperty} (?c, ?p) \wedge \text{CarbonComposition} (?p, ?comp) \wedge \text{hasValue} (?comp, ?v) \wedge \text{swrlb:lessThan} (?v, 0.25) \wedge \text{hasUnit} (?v, ?u) \wedge \text{UnitPercent} (?u) \rightarrow \text{LowCarbonSteel} (?c)$$

其中, $?$  表示 SWRL 语法规则,与实例; $c$  表示碳钢实例,即  $c$  是 CarbonSteel 的一个实例;hasValue 代表数据属性;swrlb:lessThan 表示逻辑小于;hasUnit 表示对象属性,用来与 UnitPercent 类产生联系;UnitPercent 代表百分数单位类。

SWRL 规则是由推理前提 (Antecedent) 和推理结果 (Consequent) 组成,一条规则,若其前提被满足,则其结论也必为真<sup>[9]</sup>。文中工程材料本体中部分规则如表 1 所示。

### 1.3 知识实例

知识实例是指概念与概念之间关系的具体体现,既包含概念实例,又包含各种关系以及属性。OWL 中领域知识的概念又称为类,知识概念实例叫做个体,是概念的特例或者说是底层的对象,即某个类的具体实现,类似计算机编程面向对象思想中类的对象。属性是概念间非层次关系的集合,用来描述和定义本体概念和实例,其包括对象属性和数据属性。对象属性的取值范围是定义的类,而数据属性的取值范围是



输入自然语言,经过转换后生成 SQWRL 查询语句,同时界面可以显示查询结果。

### 2.2.3 推理层

推理层主要完成推理任务,通过 Drools Rule Engine 规则引擎来执行推理任务,Drools 集成在 SWRLAPI 中,可以加载 SWRL 规则,通过 SQWRL 查询引擎进行语义查询,然后将执行推理后的查询结果返回到用户查询界面。

## 3 SQWRL 语义查询的实现

### 3.1 开发环境以及 SWRLAPI 的构建

本材料查询系统中采用美国斯坦福大学研发的本体编辑工具 Protégé<sup>3</sup>。4.1 构建本体,集成开发环境是 Eclipse4.4,win10 x86,Java8,使用 Drools<sup>[8-9]</sup> 查询推理引擎进行规则推理查询。Drools 集成了基于 SWRLAPI<sup>[11]</sup> 的 OWL2 RL<sup>[8]</sup> 推理机和 SWRL 规则引擎,使用基于 Java 语言的 API<sup>[9]</sup>,通过自定义规则语句,实现对本体的推理。经过推理后,SWRL 规则中隐含的关系通过属性与各实例直接关联,得到新的本体关系,产生新的本体知识。

虽然 SWRLTab 在早期的 Protégé 版本中支持规则处理,但是不能很好地使用外部控制,而且 SQWRL 查询也没有集成在里面,Protégé 暂时只能支持 SPARQL 三元组查询。因此,需要根据美国斯坦福大学提供的源码构建 SWRLAPI,这样才能在 Eclipse 中使用 Java 操控 SQWRL;总共分为两步进行,需要两个构建工具,一个是 Git(分布式版本控制系统)、一个是 Maven(项目管理工具)。

#### (1) Git。

Git 是一种免费、开源的版本控制系统,可以高效地管理各种大小项目,可以对项目进行分布式管理。服务器上设置 Git 仓库,本机上设置分支仓库,个人可以自己独立开发分支,然后远程提交给服务器,以便于对开发人员进行分布式管理,相互之间互不影响。

使用 git clone 命令克隆到本地,SWRLAPI<sup>[12]</sup> 需要克隆的地址如下所示:

```
git clone https://github.com/protegeproject/swrlapi.git
git clone https://github.com/protegeproject/swrlapi-drools-engine.git
git clone https://github.com/protegeproject/swrltab.git
git clone https://github.com/protegeproject/swrltab-plugin.git
git clone https://github.com/protegeproject/swrltab-project.git
git clone https://github.com/protegeproject/swrlapi-example.git
git clone https://github.com/protegeproject/swrlapi-integration-tests.git
```

tion-tests.git

将克隆地址复制到 Git Bash 界面,即可执行克隆操作,就可以将源文件保存到本地文件夹。

#### (2) Maven。

Maven 是 Java 工程中最常用的项目管理工具,同样可以通过命令的方式对项目进行管理和控制,可以对源码进行编译、测试、打包、发布等,对依赖包进行组织和管理非常方便。打开 windows 系统的命令提示符工具(cd D:\workspace\swrl)进入文件夹,其中 pom.xml 是依赖包进行管理,在 SWRLAPI 文件夹中的 pom.xml 添加如下依赖包,就可以编译打包操作。

```
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-core</artifactId>
    <version>1.1.3</version>
</dependency>
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.1.3</version>
</dependency>
```

然后使用 Maven 命令 mvn -DskipTests=true clean install,这样 SWRLAPI 就创建成功了,如表 3 所示。因此,可以使用 SWRL 和 SQWRL 的可编程接口进行开发。

表 3 SWRLAPI 程序包表

名称	Jar 包
swrlapi-1.0.0 版	swrlapi-1.0.0-SNAPSHOT.jar
swrlapi-drools-engine-1.0.0 版	swrlapi-drools-engine-1.0.0-SNAPSHOT.jar
swrltab-1.0.0 版	swrltab-1.0.0.jar
swrltab-1.0.0 依赖包	swrltab-1.0.0-jar-with-dependencies.jar
swrltab-plugin-1.0.0 版	swrltab-plugin-1.0.0-SNAPSHOT.jar

### 3.2 语义查询引擎

文中主要利用 Protégé 中的开源 OWL API 以及 SWRL API,并使用 Drools 规则推理机,是基于描述逻辑(DL)的推理,对规则进行加载并推理,以便于用来推理查询。主要执行过程如下:

(1) 利用 OWLOntologyManager 创建一个本体模型管理器,用来加载 OWL 本体到内存中;

(2) 利用 SWRLRuleEngine 接口将 SWRL 规则载入 Drools 推理机,建立连接,并进行推理;

(3) 利用 SQWRLQueryEngine 接口对推理后的本体文件进行语义查询,通过 SQWRLResult 来获取查询后的结果,结果是以二维表的形式返回给用户。

推理过程是对规则先进行推理,所有规则同时加

载到 OWL 本体中,但是查询仅仅获得了跟查询相关的规则的结果,其主要代码如下所示:

```
//定义 OWL 本体管理器
public static OWLOntologyManager manager;
//定义 OWL 本体对象
public static OWLOntology ontology;
//定义规则变量
public static SWRLAPIRule swrlRule1;
//定义规则文件路径变量
public static String fileDirectory;
//定义查询语句变量
public static String Query1;
//定义存储规则容器
List list = new ArrayList();
File file = new File( fileDirectory );
try {
    BufferedReader br = new BufferedReader( new FileReader
(file) );
    //构造一个 BufferedReader 类来读取规则文件
    String s = null;
    while( (s = br.readLine()) != null ) { //使用 readLine 方法,
        一次读一行
        s = s.substring( s.indexOf( " = " ) + 1 );
        list.add( s ); //将获取规则存入 list 容器中
        br.close();
        manager = //加载本体管理器 OWLManager.createOWLOntologyManager();
        //使用本体管理器来加载本体文件,创建本体模型
        ontology = manager.loadOntology( IRI.create( "file:/d:/materials.owl" ) );
        //使用 SWRLRuleEngine 规则引擎添加规则
        SWRLRuleEngine ruleEngine = SWRLAPIFactory.createSWRLRuleEngine( ontology ); //加载规则
        for( i = 1; i <= list.size(); i++ ) {
            //ruleEngine 的 createSWRLRule 方法直接将规则加载到内存中
            swrlRule1 = ruleEngine.createSWRLRule( "rule" + i, list[i] );
            ruleEngine.infer(); //对本体进行推理
            //使用 SQWRLQueryEngine 创建语义查询
            SQWRLQueryEngine queryEngine1 = SWRLAPIFactory.createSQWRLQueryEngine( ontology );
            //运行语句查询,并获得结果集,Query1 如表 2 所示
            SQWRLResult result = queryEngine.runSQWRLQuery( "Q1", Query1 );
            //遍历输出查询后的结果集
            while( result.next() ) { //获取查询后的实例
                SQWRLNamedIndividualResultValue materialValue = result.getNamedIndividual( "mat" );
                String material = materialValue.getShortName();
            }
        }
    }
}
```

3.3 语义查询实例

以工程材料本体为例,执行语义查询引擎,即可以查询得到结果,如图 3 所示。



图 3 用户查询界面

其中,SWRLAPI 程序包里面已经集成了 Drools<sup>[12]</sup>引擎,因此可以建立桥机制将 OWL 知识和 SWRL 规则联系起来,使用规则和描述逻辑进行推理,然后再通过 SQWRLQueryEngine 查询引擎,来执行查询语句,获得查询结果。由图 3 可知,根据 Rule1 规则,通过查询语句可以得到查询的材料为 45,40,40Cr 三种钢材材料,而 40Cr 是属于调质钢系列,但是碳含量在 0.4%,通过执行规则可以推理得到 40Cr 钢也属于中碳钢实例,说明满足规则 Rule1 的调质钢也是中碳钢。因此,通过 SQWRL 不仅能够对继承类进行查询,而且通过 Drools 引擎执行 SWRL 规则后,还完成了对推理后本体的查询任务。

4 结束语

使用基于规则的语义查询语言(SQWRL)进行推理查询,并使用 SWRLAPI 内置的 Drools 规则引擎,构建工程材料本体进行实验验证,并用第三方用户界面进行工程材料选择查询,对本体资源进行深入挖掘,对隐含的知识进行查询,进一步完善本体中的关系。使用 SQWRL 不仅可以完成简单的查询,也可以完成复杂的查询,并能推理得到新的知识。工程材料选择是一个十分复杂的过程,文中仅通过构建基于本体知识库的查询框架,展示 SQWRL 强大的规则推理能力。基于 SQWRL 的知识框架是开放式、可扩展、可重用的,该框架可以应用于其他本体的查询,可以对领域本体进行扩展,通过编写外部程序来对本体知识库进行开发应用并获取知识,并且可以不断完善,为今后语义

(下转第 24 页)



对两组实验进行综合分析,第一组测试针对静态相机拍摄视频,通过图 2 和表 1 可看出,文中算法不仅提高了匹配正确率,而且缩短了匹配时间,证明了在该情况下的有效性。第二组实验视频是在运动相机下拍摄且存在严重噪声,比较图 2(c)和图 3(c)可明显看出,匹配对间的连线在图 2(c)中是水平的,在图 3(c)中是斜的,这是由于存在相机运动,导致两帧间有一定程度的旋转,进而得到图 3(c)匹配对间的连线是斜的。该组实验验证了文中算法在该情况下的有效性。通过两组实验证明了文中算法对静态相机、相机抖动及噪声存在情况下拍摄的视频进行连续帧匹配的有效性。

#### 4 结束语

研究了视频帧间的基于 SURF 的匹配方法。采用 Fast-Hessian 快速检测特征点,并对特征点进行描述,然后通过改进的最近邻与次近邻比的方法进行双向匹配,最后用 RANSAC 算法剔除错误匹配点。通过在静态相机、动态相机、噪声这几种不同情况下拍摄的视频进行实验,证明文中算法不仅提高了匹配正确率,而且缩短了匹配时间,对于视频中基于特征匹配的跟踪、识别等都有很大的实际作用。下一步可以在特征的提取和匹配上使用 GPU 加速进一步提高匹配效率。

#### 参考文献:

- [1] 余道明. 图像配准技术研究及应用[D]. 成都:西南交通大学,2014.
- [2] Lowe D G. Object recognition from local scale-invariant fea-

ture[C]//Proceedings of the seventh international conference on computer vision. [s. l.]:IEEE,1999:1150-1157.

- [3] Lowe D G. Distinctive image features from scale-invariant features points[J]. International Journal of Computer Vision, 2004,60(2):91-110.
- [4] Bay H. SURF:speeded up robust features[J]. Computer Vision and Image Understanding,2008,110(3):346-359.
- [5] 葛盼盼,陈 强. 基于 SURF 特征提取的遥感图像自动配准[J]. 计算机系统应用,2014,23(3):16-24.
- [6] 张开玉,梁凤梅. 基于改进 SURF 的图像配准关键算法研究[J]. 科学技术与工程,2013,13(10):2875-2879.
- [7] 崔振兴,曾 威,杨明强,等. 一种改进的 SURF 快速匹配方法[J]. 江苏师范大学学报:自然科学版,2014,32(3):41-46.
- [8] Yang Gui. Point-pattern matching method using SURF and shape context[J]. OPTIK,2013,124:1869-1873.
- [9] 佟爱华. 一种改进的高精度视频帧间匹配算法[C]//第二届仪表、自动化与先进集成技术大会论文集. 出版地不详:出版者不详,2008:146-150.
- [10] 卓武汉,严京旗. 基于 SURF 的连续帧图像配准及高光去除[J]. 微型电脑应用,2011,27(1):37-39.
- [11] Yu J, Jeon M, Pedrycz W. Weighted feature trajectories and concatenated bag-of-features for action recognition[J]. Neurocomputing,2014,131(7):200-207.
- [12] 赵璐璐. 基于 SURF 和快速近似最近邻搜索的图像匹配算法[J]. 计算机应用研究,2013,30(3):921-923.
- [13] Feng Qi, Xu W, Zhang W, et al. Research of image matching based on improved SURF[J]. Telkomnika Indonesian Journal of Electrical Engineering,2014,12(2):1395-1402.
- [14] 宋卫艳. RANSAC 算法及其在遥感图像处理中的应用[D]. 北京:华北电力大学,2011.

(上接第 19 页)

查询提供另一种方式。但是语义规则不具有通用性和易读性,规则可能造成本体关系冲突,维护成本较高。在实际应用中,如何提升查询的准确率和效率,简化 SQWRL 语言,解决冲突等问题是今后的研究方向。

#### 参考文献:

- [1] O'Connor M J, Das A. SQWRL: a query language for OWL [C]//Fifth international workshop on OWL: experiences and directions. Chantilly, VA:[s. n.],2009.
- [2] 张艳涛,陈俊杰,相 洁. 基于 SWRL 本体推理研究[J]. 微计算机信息,2010,26(3-3):182-183.
- [3] 张宗仁,杨天奇. 基于自然语言理解的 SPARQL 本体查询[J]. 计算机应用,2010,30(12):3397-3400.
- [4] 王 海,范 琳,李增智. 基于 SQWRL 的语义 Web 服务发现[J]. 微电子学与计算机,2010,29(9):76-79.
- [5] Zhang Yingzhong, Luo Xiaofang, Zhao Yong, et al. An ontology-based knowledge framework for engineering material selec-

tion[J]. Advanced Engineering Informatics,2015,29(4):985-1000.

- [6] 周 亮,黄志球,倪 川. 基于 SWRL 规则的本体推理研究[J]. 计算机技术与发展,2015,25(10):67-70.
- [7] O'Connor M J, Das A. A pair of OWL 2 RL reasoners[C]//Proceedings of OWL experiences on directions workshop. [s. l.]:[s. n.],2012:1-10.
- [8] Horridge M, Bechhofer S. The OWL API: a java API for OWL ontologies[J]. Semantic Web,2011,2(1):11-21.
- [9] Horrocks I, Patel-Schneider P F, Boley H, et al. SWRL: a semantic web rule language combining OWL and RuleML[C]//World wide web consortium. [s. l.]:[s. n.],2004.
- [10] 张新华,张 飞. “知识”概念及其涵义研究[J]. 图书情报工作,2013,57(6):49-58.
- [11] Protégéteam. SWRLTab build project[EB/OL]. 2014. <https://github.com/protegeproject/swrltab-project>.
- [12] 刘 伟. Java 规则引擎-Drools 的介绍及应用[J]. 微计算机应用,2005,26(6):717-721.