

DOI: 10.3969/j.issn.1672-6685.2009.04.008

# 基于 SWRL 和 Jess 构造语义 Web 规则及其对策分析<sup>\*</sup>

纪兆辉, 李存华

(淮海工学院 计算机工程学院, 江苏 连云港 222005)

**摘 要:**以 Family 本体为样例, 采用 OWL 开发平台 Protégé 及 SWRL 规则编辑器定义本体并构造语义 Web 规则, 基于 SWRLJessTab 和 JessTab 将 Jess 规则引擎与 OWL 和 SWRL 集成, 为语义 Web 提供基于规则的推理机制, 实现知识库共享和规则的互操作。对出现的问题及对策进行了较深入的分析。

**关键词:** 语义 Web; 本体; Protégé; OWL; SWRL; Jess; 规则

中图分类号: TP393

文献标识码: A

文章编号: 1672-6685(2009)04-0026-04

## Constructing and Strategy Analyzing of Rule System for the Semantic Web Based on SWRL and Jess

Ji Zhao-hui, Li Cun-hua

(School of Computer Engineering, Huaihai Institute of Technology, Lianyungang 222005, China)

**Abstract:** Taking the family ontology as an example, SWRL rule editor was used within the Protégé, to build ontology and construct rules for the semantic web. The integration of Jess rule engine with OWL and SWRL editor was implemented based on SWRLJessTab and JessTab, thus providing a rich rule-based reasoning facility for the semantic web. The sharing of OWL knowledge base and rules interoperability was realized. Some issues and strategies related to rules construction were also deeply discussed.

**Key words:** semantic web; ontology; Protégé; OWL; SWRL; Jess; rule

目前大多数基于规则的系统, 其互操作能力非常有限。语义 Web 的一个基本目标是实现互操作<sup>[1-3]</sup>, 良好的规则系统是实现这个目标的关键。为了实现规则库的共享及基于规则引擎处理规则库的目标, 产生了 RuleML, SWRL 等规则语言。其中 SWRL<sup>[4]</sup> 被设计为语义 Web 的规则语言, 致力于语义 Web 中规则的互操作。

由斯坦福大学开发的 SWRL 编辑器<sup>[5]</sup> 是一个具有高度交互性的开源 SWRL 编辑器, 它与 Protégé<sup>[6]</sup> 紧密集成, 提供了具有高度交互性的规则

编辑界面, 支持现有 SWRL 语言的所有特征, 允许在 SWRL 规则编辑和 OWL 本体编辑间进行无缝切换。用户可在 SWRL 规则中方便地引用 OWL 中的实体。SWRL 编辑器的一个主要目标是在 SWRL 和现有的规则引擎之间进行互操作。其重要部件是编辑器所提供的支持与规则引擎紧密集成的机制。这种机制由一个称为 SWRL Factory<sup>[2-3]</sup> 的子系统所支持。

本文基于 Jess 规则引擎<sup>[7]</sup> 与 SWRL 规则编辑器的集成来进行语义 Web 规则系统的开发。在

<sup>\*</sup> 收稿日期: 2009-09-23; 修订日期: 2009-11-09

基金项目: 江苏省高校自然科学基金项目(08KJD520013)

作者简介: 纪兆辉(1971—), 男, 江苏邳州人, 淮海工学院计算机工程学院讲师, 硕士, 主要从事数据挖掘、语义 Web 等方面的研究, (E-mail) jizhaohui@jsmail.com.cn.

Protégé 本体编辑环境中创建 Family 本体, 利用其中的 SWRLTab<sup>[5]</sup> 构造规则系统。使用 Jess 规则引擎交互地运行 SWRL 规则或通过创建 Jess 规则来操纵 OWL 知识库, 并将推理所产生的新知识更新到 OWL 知识库中。

1 样例本体的创建

1.1 OWL 中的三类实体

OWL 区分三类实体: ① 个体(实例)。代表感兴趣的某一领域中的实体或现象。② 类。具有相似特征的个体集合, 可以组织成父、子层次。如果一个个体是某个类的成员, 则该个体也是该类所有父类的成员。③ 属性。代表两个个体间或一个个体与某数据类型间的二元联系。

1.2 Family 样例本体的 OWL 描述

定义类的集合  $Class = \{ Person, Man, Woman \}$ , 其中 Person 是  $OWL:Thing$  的子类,  $Person \sqsubseteq Man \sqcup Woman$ , 且  $Man \sqsubseteq \neg Woman$ 。分别为

Person 类定义 Datatype 类型的属性 hasName, hasAge 和表达家族成员称谓的属性 hasParent, hasFather, hasMother 等 18 个对象属性, 并设置属性特性如传递、对称、逆反等。

2 基于 SWRL 的规则构造

2.1 SWRL 概述

在语义 Web 体系结构中, 本体属数据层, 用来描述语义 Web 资源。虽然本体支持推理, 但因没有提供用于推理的规则<sup>[8]</sup>, 其能力非常有限。

语义 Web 需要在本体数据的基础上加入规则以支持更为广泛的推理, 当前对 OWL 进行规则扩展主要基于 SWRL。

SWRL 规则呈现为前提和结论之间的蕴涵式, 主要作用于 OWL 的实例及其联系, 一般形式为:  $Antecedent \rightarrow Consequent$ 。

2.2 基于 Family 本体构造 SWRL 规则

Family 本体的 SWRL 规则描述如表 1 所示。

表 1 Family 本体的 SWRL 规则  
Table 1 SWRL rules for Family ontology

序号	规则名	规则表达式
1	Def-hasUncle	$hasParent(?x, ?y) \wedge hasBrother(?y, ?z) \rightarrow hasUncle(?x, ?z)$
2	Def-hasAunt	$hasParent(?x, ?y) \wedge hasSister(?y, ?z) \rightarrow hasAunt(?x, ?z)$
3	Def-hasNiece	$hasSibling(?x, ?y) \wedge hasDaughter(?y, ?z) \rightarrow hasNiece(?x, ?z)$
4	Def-hasNephew	$hasSibling(?x, ?y) \wedge hasSon(?y, ?z) \rightarrow hasNephew(?x, ?z)$
5	Def-hasGrandpa	$hasParent(?x, ?y) \wedge hasFather(?y, ?z) \rightarrow hasGrandpa(?x, ?z)$
6	Def-hasGrandma	$hasParent(?x, ?y) \wedge hasMother(?y, ?z) \rightarrow hasGrandma(?x, ?z)$

另外, 定义规则 7~12 来判断个体是否是 Woman 的实例, 如规则 7 为:  $hasMother(?x, ?y) \rightarrow Woman(?y)$ 。定义规则 13~18 来判断个体是否是 Man 的实例, 如规则 13 为:  $hasFather(?x, ?y) \rightarrow Man(?y)$ 。篇幅所限, 这些规则未在表中列出。

识表示为 Jess 的事实; ③ 基于规则和事实运行 Jess 推理引擎进行推理; ④ 用推理后得到的新事实更新 OWL 知识库。其工作过程如图 1 所示。

3 基于 Jess 的规则构造及推理

3.1 Jess 简介

Jess (java expert shell system) 是基于 Java 语言的 CLIPS 推理机。当用户使用 class/slot 的概念描述本体时, Protégé 仅能使用 Jess 进行推理。

JessTab<sup>[9]</sup> 是 Protégé 的插件, 允许用户集成使用 Jess 和 Protégé。

Jess 与 Protégé 集成, 可以完成 4 项任务: ① 将 SWRL 的规则转换为 Jess 的规则或直接在 Jess 中基于本体概念创建规则; ② 将有关 OWL 个体的知

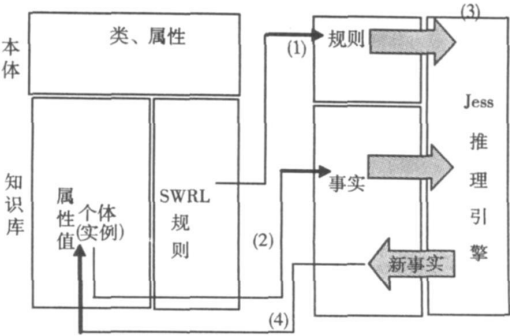


图 1 基于本体知识库的 Jess 推理过程  
Fig. 1 Inference process of Jess based on ontology knowledge base

### 3.2 将 Family 本体实例映射为 Jess 中的事实

在 JessTab 的工作台窗口中, 定义事实可以有两种方式<sup>[10]</sup>: ① 用 `deftemplate` 定义模板和 `slot`, 然后用 `assert` 声明一到多个事实; ② 用 Jess 的 `map-class` 或 `mapinstance` 函数将本体中的实例及其联系映射为 Jess 中的事实。映射后的事实为一到多个 object, 可以用 (facts) 函数直接查看。每个 object 具有如下形式:

```
(object (is-a Class-name) (is-a-name "Class-name"))
```

```
(OBJECT < External-Address; SimpleInstance >)
```

```
(Slot-name1 slot-value1) (Slot-name2 slot-value2) ...)
```

### 3.3 编写 Jess 规则并推理

篇幅所限, 本文结合 Family 本体, 仅用以下规则来说明 Jess 中规则的定义方法。

规则: 输出 Family 本体知识库中具有侄女的所有成员的姓名及其侄女姓名。

规则表达式:

```
(defrule Niece-role-of-Family
  (object (is-a Woman | Man) (OBJECT ? p)
    (hasName ? pn) (hasBrother ? b))
  (object (is-a Man) (OBJECT ? b) (hasName ? mn)
    (hasDaughter ? w))
  (object (is-a Woman) (OBJECT ? w) (hasName ? wn))
  => (printout t ? pn " has niece named: " ? wn
    crlf))
```

用 (run) 命令运行 Jess 推理引擎后, 所得到的新知识会自动更新到 OWL 知识库中。

## 4 规则构造中的常见问题及对策

### 4.1 规则的交叉定义问题

由于 OWL 可以借助于类及属性特征进行初步推理, 因此与之相应的规则应避免交叉定义。本文对 OWL 公理与 SWRL 规则的对应关系进行较全面的分析, 总结如下。

(1) 子类公理 ( $C \subseteq D$ )。对应的 SWRL 表达式为:  $C(?x) \rightarrow D(?x)$ 。

(2) 等价类公理 ( $C \equiv D$ )。对应的 SWRL 表达式为:  $C(?x) \rightarrow D(?x), D(?x) \rightarrow C(?x)$ 。

(3) 子属性公理 ( $Q \subseteq P$ )。对应的 SWRL 表达

式为:  $Q(?x, ?y) \rightarrow P(?x, ?y)$ 。

(4) 等价属性公理 ( $P \equiv Q$ )。对应的 SWRL 表达式为:  $P(?x, ?y) \rightarrow Q(?x, ?y), Q(?x, ?y) \rightarrow P(?x, ?y)$ 。

(5) 逆反属性 ( $P \equiv \bar{Q}$ )。对应的 SWRL 表达式为:  $P(?y, ?x) \rightarrow Q(?x, ?y), Q(?x, ?y) \rightarrow P(?y, ?x)$ 。

(6) 传递属性 ( $P^+ \subseteq P$ )。对应的 SWRL 表达式为:  $P(?x, ?y) \wedge P(?y, ?z) \rightarrow P(?x, ?z)$ 。

(7) 函数属性 ( $T \subseteq \leq 1P$ )。对应的 SWRL 表达式为:  $P(?x, ?y) \wedge P(?x, ?z) \rightarrow \text{same As} (?y, ?z)$ 。

### 4.2 SWRL 和 Jess 的规则调试问题

SWRL 规则编辑器可对处于编辑之中的 SWRL 规则进行拼写检查和知识库的匹配检查, 若出现拼写错误, 则在编辑窗口中会用红色边框加以提醒, 这类错误比较容易调试。但若出现对 OWL 知识库中并未存在的某个实体如个体、属性、公理或对不存在的 SWRL 内置函数的引用之类的错误, 则较难调试。另外, 在对 Jess 规则进行调试时, 要特别留意函数名的拼写及括号的匹配问题。错误的 SWRL 和 Jess 规则将不予保存。

### 4.3 规则的冲突问题

随着规则集中规则数目的增加, 可能出现规则的冲突, 导致推理引擎产生错误或自相矛盾的结果。该类问题在设计阶段很难发现, 一般是在推理机运行时出现了错误的推理结果时, 顺着产生推理结果的规则采用倒推的策略, 可逐步发现导致冲突的规则, 并将其删除。

### 4.4 规则集的约简问题

多数情况下, 要依据谓词演算知识, 对规则进行必要的约简。本文总结了如下 6 条经验准则, 简化起见, 此处将参数省略。

(1) 若有规则  $A \rightarrow B$ , 则有  $B \not\sqsubseteq A$ 。否则为平凡规则, 应从规则集中删去。

(2) 若有规则  $A \wedge B \wedge C \rightarrow D, B \rightarrow C$ , 则可以约简为  $A \wedge B \rightarrow D$ 。

(3) 若有规则  $A \rightarrow B, B \rightarrow C, A \rightarrow C$ , 则  $A \rightarrow C$  为冗余, 将其从规则集中删去。

(4) 若要表达规则  $A \rightarrow B_1 \wedge B_2$ , 应将其变换为  $A \rightarrow B_1$  和  $A \rightarrow B_2$ 。

(5) 若要表达规则  $A_1 \vee A_2 \rightarrow B$ , 应将其变换为  $A_1 \rightarrow B$  和  $A_2 \rightarrow B$ 。

(6) 若要表达规则  $A \rightarrow (B \rightarrow C)$ , 应将其变换为  $A \wedge B \rightarrow C$ 。

5 实验结果

在 Protégé 的集成环境中, 通过 SWRLTab 下的 SWRLJessTab<sup>[11]</sup> 可以将 SWRL 规则与 Jess 规则引擎进行集成。点击“OWL+SWRL→Jess”, 可将 OWL 及 SWRL 构成的知识库映射为 Jess 的事实和规则。在笔者的实验中, SWRL 的 18 条规则映射为 Jess 的 18 条规则; Man, Woman, Person 和 owl: Thing 分别映射为 Jess 的类; OWL 的公理如个体间的联系及 DifferentFrom, SameAs 关系导出到 Jess, 共有 52 条。点击“Run Jess”启动 Jess 规则引擎, 则基于已经导出的事实和规则进行推理, 推导出 30 条公理。其中, 对规则 1~6 推理产生实例间的 hasUncle 联系 7 个、hasAunt 联系 2 个、hasNiece 联系 6 个、hasNephew 联系 3 个、hasGrandma 联系 6 个、hasGrandpa 联系 6 个。推理后, 17 个 Person 个体被重新分类, 其中由规则 7~12 产生 9 个 Woman 实例, 由规则 13~18 产生 8 个 Man 实例。

在 SWRLJessTab 中点击“Jess→OWL”, 用推理后产生的新事实更新 OWL 知识库。现有 OWL 知识库中, 对原先的 17 个 Person 实例进行了重新分类, 并且出现 30 条新的公理。

6 结束语

本文以 Family 本体作为样例, 基于 Protégé3. 4 这一本体开发环境来创建本体中的类、属性、个体, 在 SWRLTab 环境下编辑 SWRL 规则, 并基于 Jess71p2 版本的推理引擎, 利用 Protégé 的 SWRLJessTab 和 JessTab 插件实现对 OWL 和 SWRL 构成的知识库进行推理, 为语义 Web 提供基于规则的推理机制, 实现了知识库的共享和规则的互操作。文中还对规则构造中的常见问题及对策进行了较深入的分析。

参考文献:

[ 1 ] O' CONNOR M J, SHANKAR R, NYULAS C, et al.

Developing a Web-Based Application using OWL and SWRL[ EB/ OL]. (2008-01-12)[ 2009-07-28]. <http://www.aaai.org/Papers/Symposia/Spring/2008/SS-08-01/SS08-01-012.pdf>.

[ 2 ] O' CONNOR M J, KNUBLAUCH H, TU S, et al. Writing Rules for the Semantic Web using SWRL and Jess[ EB/ OL]. (2005-07-18)[ 2009-05-27]. <http://www.med.univ-rennes1.fr/~cgolb/Protege2005/SWRLJessOConnor.pdf>.

[ 3 ] O' CONNOR M J, KNUBLAUCH H, TU S, et al. Supporting Rule System Interoperability on the Semantic Web with SWRL[ C]. Ireland Galway: Fourth International Semantic Web Conference (ISWC2005), 2005: 312-336.

[ 4 ] World Wide Web Consortium. SWRL: a Semantic Web Rule Language Combining OWL and RuleML[ EB/ OL]. (2004-05-21)[ 2009-03-27]. <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521>.

[ 5 ] CIM Engineering, Inc. ProtegeWiki: SWRLTab[ EB/ OL]. (2009-03-05)[ 2009-06-21]. <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab>.

[ 6 ] Stanford Center for Biomedical Informatics Research. The Protégé Ontology Editor and Knowledge Acquisition System[ EB/ OL]. (2009-06-18)[ 2009-06-29]. <http://protege.stanford.edu/>.

[ 7 ] Sandia National Laboratories. Jess the Rule Engine for the Java Platform[ EB/ OL]. (2008-11-11)[ 2009-07-12]. <http://Herzberg.ca.sandia.gov/jess>.

[ 8 ] 戴维民. 语义网信息组织技术与方法[ M]. 上海: 学林出版社, 2008: 238-254.

[ 9 ] ERICKSSON H. JessTab: Integrating Protégé and Jess[ EB/ OL]. (2006-11-20)[ 2009-07-15]. <http://www.ida.liu.se/~her/JessTab/>.

[ 10 ] ERICKSSON H. JessTab Tutorial at the 9th Intl. Protégé Conference[ EB/ OL]. (2006-07-23)[ 2009-07-15]. <http://www.ida.liu.se/~her/JessTab/tutorial06/>.

[ 11 ] CIM Engineering, Inc. ProtegeWiki: SWRLJessTab[ EB/ OL]. (2009-03-24)[ 2009-07-15]. <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLJessTab#nid6RJ>.

(责任编辑: 吉美丽)