

使用 OWL DL 形式化表达对象角色建模模型

潘文林^{1,2}, 刘大昕³

(1. 哈尔滨工程大学 机电工程学院, 哈尔滨 150001; 2. 黑龙江省现代制造工程技术研究中心, 哈尔滨 150001;

3. 哈尔滨工程大学 计算机科学与技术学院, 哈尔滨 150001)

(panwenlin@hrbeu.edu.cn)

摘要:对象角色建模(ORM)方法已应用于本体工程,因此需要将 ORM 模型转换为 OWL DL 公理,以便将 ORM 本体发布到语义 Web 上,同时还可使用支持 DL 的推理机来检查 ORM 本体的语义一致性和冗余问题。通过模型语义分析、模型等价转换、引入新的运算符和特性等方法,提出将 ORM 模型形式化表达为 OWL DL 公理的规则。除了外部唯一约束等四种约束外,其他形态的 ORM 模型都可以形式化表达为 OWL DL 公理。

关键词:对象角色建模; Web 本体语言; 描述逻辑; 模型映射

中图分类号: TP311 **文献标志码:** A

Representing object role modeling models with Web ontology language description logic axioms

PAN Wen-lin^{1,2}, LIU Da-xin³

(1. College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin Heilongjiang 150001, China;

2. Heilongjiang Modern Manufacturing Engineering Research Center, Harbin Heilongjiang 150001, China;

3. College of Computer Science and Technology, Harbin Engineering University, Harbin Heilongjiang 150001, China)

Abstract: Object Role Modeling (ORM) has been used in ontology engineering to model domain ontology, which needs to represent ORM models in OWL DL axioms to check semantic conflicts and redundancy with DL reasoners, and to publish ORM ontology on the semantic Web. By means of comparing the semantics of ORM model and OWL DL axioms, equivalently model-converting, and introducing new operators and properties, that mapping rules to represent ORM models in OWL DL axioms was proposed. Except a few constraints, most ORM model elements can be represented by OWL DL axioms.

Key words: Object Role Modeling (ORM); Web Ontology Language (OWL); Description Logic (DL); model mapping

0 引言

对象角色建模(Object Role Modeling, ORM)是一种面向事实的概念建模方法,是在自然语言信息分析方法(Natural language Information Analysis Method, NIAM)基础上发展起来的^[1]。与实体联系(Entity Relationship, ER)建模和统一建模语言(Unified Modeling Language, UML)等实体/对象建模方法不同,ORM的信息模型中没有属性的概念,因此能够构建更加稳定的模型,表达更复杂的业务规则。更重要的是,由于ORM源于NIAM,其概念模型可以完全转换为伪自然语言描述,这样就可以让领域专家通过阅读伪自然语言描述来确认内容的正确性,这是其他任何信息建模方法没法解决的问题,即领域专家无法真正参与模型内容的质量把控。

目前,ORM已经发展到2.0版^[2],并已开始应用到本体工程中,例如,DOGMA本体工程方法中使用ORM来表达领域本体模型^[3],对象管理组织在其提出的业务词汇和业务术语的语义标准规范(Semantics of Business Vocabulary and Business Rules, SBVR)中采用ORM的文本建模原语来表达业务规则^[4]。构建本体的目的是为了共享知识,尤其是在网络上交换和共享领域知识。Web本体语言(Web Ontology Language, OWL)是目前语义Web使用的标准本体描述语言。如果要通过语义Web共享ORM本体知识,需要将ORM的语

义模型转换为OWL的形式化公理。

使用ORM模型来表达本体知识,需要通过推理来检查模型的语义冲突和冗余等问题,目前还没有支持ORM模型推理的推理引擎。目前,多数推理引擎支持描述逻辑(Description Logic, DL),而OWL DL的逻辑基础是描述逻辑的一个子集SHOIN,有推理引擎支持。

基于上述两个原因,有必要深入研究如何用OWL DL形式化表达ORM 2.0的模型。现有的相关研究包括:Jarrar研究了如何将ORM映射为描述逻辑DL_{idf}(一种支持标识和函数的描述逻辑),提出了映射规则^[5],同时他还研究了如何将ORM映射为描述逻辑SHOIN,并提出了映射规则^[6]; Nguyen等人也研究了将ORM映射为描述逻辑SHOIN的一些基本规则^[7]。Jarrar完整地研究了对ORM各种模型形态的映射方法,但缺少对对象类型建模原语及取值约束等的映射分析; DL_{idf}的逻辑结构与SHOIN不完全相同,前者通过标识运算支持多元关系的描述,后者则只能支持二元关系,使用前者可以表达大部分ORM模型,但使用后者则不行;虽然SHOIN是OWL DL的逻辑基础,两者在语义上是等价的,但表达形式完全不同,有时不仅仅是对应转换的问题。

本文将完整地分析使用OWL DL形式化表达ORM 2.0模型的方法。为了简化,本文采用OWL DL的抽象语法表达式;文中用OWL代表OWL DL; OWL公理中省略了概念和特性

收稿日期:2010-08-10;修回日期:2010-11-26。

作者简介:潘文林(1972-),男,云南泸水人,副教授,博士研究生,CCF会员,主要研究方向:数据库、本体论、信息建模; 刘大昕(1941-),男,江苏海门人,教授,博士生导师,主要研究方向:数据库与知识工程。

等的名字空间;文中诸如“F01”等开始的段落表示 ORM 模型的 OWL 公理化,以“R01”等开始的段落为模型等价变换。

1 ORM 模型样例

为了便于解释,先给出一个 ORM 模型作为全文的样例。图 1 是一个图书相关的 ORM 模型,语义解释为:图书由国际标准书号(International Standard Book Number, ISBN)来标识;每一图书有且只有一个名称;每一图书最多可译自另一图书,且不可循环翻译;图书可完全分为纸质图书和电子图书两类,这两类没有重叠;图书具有是否售完的状态;每一图书至少有一个撰写的人,且图书和撰写者的组合是唯一的;图书和阅读者的组合是唯一的;每一图书或者没有评论或者至少 2 人评论,图书和评论者的组合是唯一的,评论某图书的人必须是阅读该图书的人,撰写者不能评价自己写的书;评论可能会给出为 1~5 的打分;每人具有唯一称呼(“先生”或“女士”)和唯一姓名。

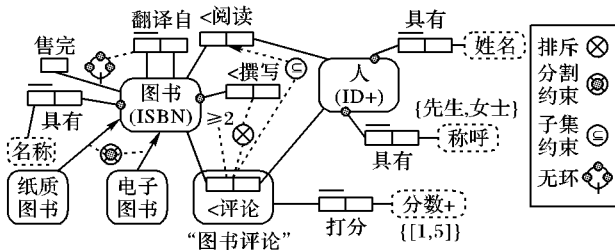


图1 图书相关的 ORM 模型

ORM 模型中:矩形代表对象类型,实线矩形是实体类型,虚线矩形是值类型;值类型名称后加“+”的代表数值(例如“得分+”),否则是字符串(例如“称呼”);对象类型之间的关系是事实类型,每个对象在事实类型中担当的角色用小条形矩形表示,事实用有阅读方向的谓词来描述,例如“<人>撰写<图书>”是一个事实类型;事实或角色组合的唯一性约束由标识在角色框上的直线表示,实线箭头代表实体类型之间的父子类型关系,分类关系可以有排斥、整体和分割约束;对象类型的所有实例是否强制担当角色由黑点对象类型及其角色连接线上的黑圆点表示;事实类型之间或角色值集合之间存在相等、排斥和子集约束;事实或角色值的频率约束直接用引线标识在事实或角色上,例如“≥2”;“{先生,女士}”和“{[1,5]}”是值类型的取值约束;事实类型“翻译自”是无环的。

2 对象类型和事实类型

ORM 模型的基本要素是对象类型及其之间的关系事实类型,对象类型担当事实类型中的各种角色。ORM 的对象类型分为实体类型和值类型两种。ORM 支持一到多元的事实类型(即关系),但 OWL 不能直接表达关系,OWL 将关系表达为特性,这样只能表达二元关系。因此,需要将 ORM 模型中的一元事实类型和三元以上的事实类型全部转换为二元事实类型,然后再用 OWL 形式化。Wintraecken 证明了可将多元关系等价地转换为二元关系^[8]。因此,本文只讨论一元和二元事实类型及其相关约束。

先讨论一元事实类型(又称为一元谓词)。一元谓词通常表达某实体对象是否处于某种状态,可以转换为二元事实类型来表达。我们引入表征状态的值类型“P 状态”来表达一

元谓词 P,取值为“true”或“false”。等价转换如图 2 所示。

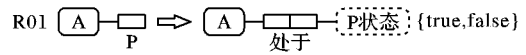


图2 一元事实类型等价转换为二元事实类型

经过图 2 所示转换,ORM 模型中只包含二元事实类型。

二元事实类型有两种形态:1)表达两实体对象类型之间的关系;2)表达实体对象类型具有某种特性。如图 3 所示。

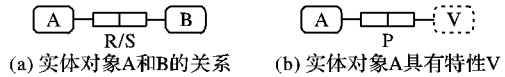


图3 二元事实类型的两种形态

先讨论实体对象类型之间的关系如何表达为 OWL 公理。实体对象类型与 OWL 中类的语义相同,因此,ORM 模型中的“纸质图书”和“电子图书”等实体对象类型可表达为 OWL 的类。OWL 的对象特性表达类之间的二元关系,因此可直接用来表达 ORM 模型中实体对象之间的关系。通常,实体对象之间的关系是单向的,如果是双向的则两个关系之间是可逆的,OWL 使用 inverseOf 来表达关系的这种特性。因此表达实体对象类型的二元事实类型其 OWL 公理化方法:实体对象类型分别表达为 OWL 类;谓词表达为一个 OWL 对象特性,其定义域和值域分别是谓词的起始和指向的实体对象类型;如果存在两个谓词,则两个谓词对应的 OWL 对象特性之间是互逆的。如果无强制约束,则出现在事实中的实体类型的实例可能只是其所有实例的部分,需要使用 OWL 的取值限定 restriction 类和类公理 SubClassOf 来表达。实体类型和二元事实类型的 OWL 公理表达如图 4。

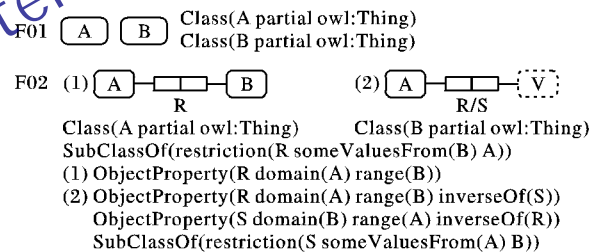


图4 实体对象和事实类型的转换规则

值类型和实体对象类型担当角色的二元事实类型通常表达实体对象类型具有某种特性,例如“姓名”和“称呼”等,因此,这样的二元事实类型及担当其角色的值类型可直接使用 OWL 的数据类型特性公理 DatatypeProperty 来表达,特性名称构成是“谓词名_值类型名”,其定义域是实体对象类型,值域是值类型的数据类型。值类型本身就蕴含了数据类型字符串(string)或数值(number)。ORM 字符串与 OWL 内置的字符串类型 xsd:string 一致,但 OWL 中表示数值类型的数据类型有多种,没有直接对应数值类型的数据类型。鉴于 ORM 的数值类型表达可做数值计算的数,我们定义一个易于实现的 OWL 数据类型 owl:number 来表达,该数据类型是 OWL 数值相关类型的并集。表达实体对象类型具有某特性值的二元事实类型的 OWL 公理表达如图 5 所示。

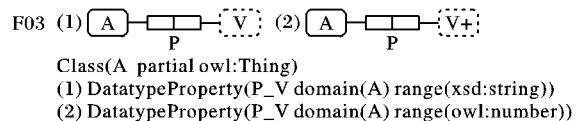


图5 表达特性关系的二元事实类型转换规则

由于实体类型和值类型都要扮演事实类型中的角色,为了方便形式化事实类型及相关约束,我们采用另一种公理

化方法,即把值类型也表达为 OWL 的类。由于 ORM 的数值类型蕴含了对象名称、值及其数据类型,因此,可以按照图 6 方式把值类型等价转换为实体类型。

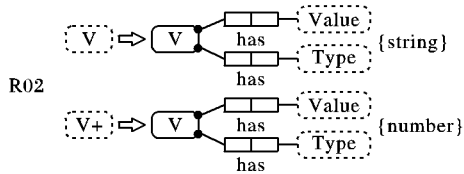


图 6 值类型的等价转换关系

转换后的模型中,Type 值类型的取值是唯一限定的,“string”或“number”。按照 F03 和上面等价转换后的模型,数值类型的 OWL 公理表达如图 7 所示。

F04 (1) $\{V\}$ (2) $\{V+\}$ Class(V partial owl:Thing)
DatatypeProperty(has_Value Domain(V) Range(xsd:string))
(1) DatatypeProperty(has_Type Domain(V) Range("String"))
(2) DatatypeProperty(has_Type Domain(V) Range("Number"))

图 7 值类型的转换规则

把数值类型等价转换为实体类型后,ORM 模型中的二元事实类型全部变换为表达实体对象之间的关系,按照 F02 来公理化即可。

ORM 的引用模式是实体对象与其引用的值对象之间存在的二元事实类型的简化表示,引用模式是实体对象的标识,即实体对象的标识码是其引用模式中的值类型。ORM 引用模式的等价模型如图 8 所示。



图 8 引用模式的等价变换

等价模型中涉及到的唯一约束和强制约束的公理化在本文后面讨论,与引用模式对应的值类型按照上面讨论的方式处理和公理化。

样例中的“图书评论”是对象化事实类型“评论”产生的虚拟实体类型。OWL 中没有将对象特性类化的概念,不能直接公理化对象化实体类型。对象化实体类型的实例是事实类型的标识角色值组合的集合,因此,可以把对象化实体类型做如图 9 的模型变换(虽然语义上不完全等价,但可以表达出对象化类型与标识角色的实体类型之间的关系)。

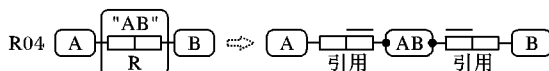


图 9 对象化类型变换为两个引用关系

经过这样的模型变换,可以使用 OWL 公理化对象化实体类型及其相关特性。

ORM 中的独立对象类型和外部对象类型是与具体实现相关的建模符号,在概念建模时不需要考虑,因此不予讨论。

至此,我们可以使用 OWL 公理化表达前面给出的 ORM 样例模型中的所有对象类型和事实类型:

-- 实体类型“纸质图书”和“电子图书”的公理化
Class(纸质图书 partial owl:Thing) Class(电子图书 partial owl:Thing)
-- 引用模式“图书”和“人”的公理化
Class(图书 partial owl:Thing) Class(ISBN partial owl:Thing)
DatatypeProperty(has_Value Domain(ISBN) Range(xsd:string))
DatatypeProperty(has_Type Domain(ISBN) Range("string"))
... -- 类“人”的公理结构与“图书”的完全相同,此处未列出
-- 值类型“名称”、“姓名”、“称呼”、“分数”的公理化

Class(名称 partial owl:Thing)
DatatypeProperty(has_Value Domain(名称) Range(xsd:string))
DatatypeProperty(has_Type Domain(名称) Range("string"))
... -- “姓名”和“称呼”的公理与“名称”的结构完全相同
DatatypeProperty(has_Value Domain(分数) Range(xsd:string))
DatatypeProperty(has_Type Domain(分数) Range("number"))
-- 一元事实类型“售完”的公理化
Class(售完状态 partial owl:Thing)
DatatypeProperty(has_Value Domain(售完状态) Range(xsd:string))
DatatypeProperty(has_Type Domain(售完状态) Range("string"))
-- 二元事实类型的公理化
ObjectProperty(图书_具有_名称 domain(图书) range(名称))
SubClassOf(restriction(图书_具有_名称 someValuesFrom(名称)) 图书)
ObjectProperty(图书_具有_售完状态 domain(图书) range(售完状态))
SubClassOf(restriction(图书_售完 someValuesFrom(售完状态)) 图书)
ObjectProperty(翻译自 domain(图书) range(图书))
SubClassOf(restriction(翻译自 someValuesFrom(图书)) 图书)
ObjectProperty(人_具有_姓名 domain(人) range(姓名))
SubClassOf(restriction(人_具有_姓名 someValuesFrom(姓名)) 人)
ObjectProperty(人_具有_称呼 domain(人) range(称呼))
SubClassOf(restriction(人_具有_称呼 someValuesFrom(称呼)) 人)
ObjectProperty(撰写 domain(人) range(图书))
SubClassOf(restriction(撰写 someValuesFrom(图书)) 人)
... -- “阅读”和“评论”的公理结构与“撰写”的完全相同
-- 对象化实体类型的近似公理化
Class(评论图书 partial owl:Thing)
ObjectProperty(引用 domain(评论图书) range(图书))
ObjectProperty(引用 domain(评论图书) range(人))
-- 对象化实体类型参与的二元事实类型“打分”的公理化
ObjectProperty(打分 domain(评论图书) range(分数))
SubClassOf(restriction(打分 someValuesFrom(分数)) 评论图书)

3 子类关系及相关约束

OWL 子类公理的语义定义类之间的子类关系,ORM 的子类关系定义实体类型之间的真子类关系。OWL 子类公理的语义包含了 ORM 子类关系的语义,因此,可直接公理化表达 ORM 的子类关系,如图 10 所示。

F05 $A \leftarrow B$ SubClassOf(B, A)

图 10 子类关系的转换规则

ORM 的子类关系之间存在三种约束,即排斥、全集和分割,分割约束是前两者的组合。OWL 有直接表达子类关系约束的类公理,如图 11 所示。

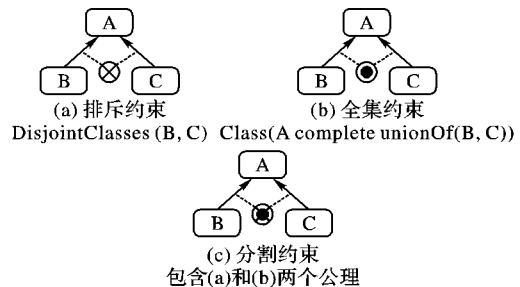


图 11 子类关系约束的转换规则

ORM 样例模型的“纸质图书”和“电子图书”是“图书”的子类,且有分割约束限制,其 OWL 公理化如下:

-- 子类关系及其约束
SubClassOf(纸质图书, 图书) SubClassOf(电子图书, 图书)

DisjointClasses(纸质图书, 电子图书)

Class(图书 complete unionOf 纸质图书, 电子图书))

4 约束

4.1 值约束

ORM 的值约束限定在值类型或其担当的角色上,分为枚举和取值范围两种:枚举的形态为“ $\{a_1, \dots, a_n\}$ ”其中的元素 a_i 可以是字符串,也可以是数值;取值范围的形态为“ $\{m..n\}$ ”、“ $\{a..b\}$ ”、“ $\{..a\}$ ”、“ $\{[a, b]\}$ ”、“ $\{[a, b)\}$ ”、“ $\{(a, b)\}$ ”和“ $\{(a, b]\}$ ”,前三种是离散数据(即整数),后四种是连续数据(即实数)。在数据类型相同的情况下,值约束可以是上述这些基本类型的混合,本文只讨论基本值约束。

对于枚举值约束的数值类型,其实例全部来自枚举约束限定的那些值,与 OWL 的枚举类公理语义相同。

F07 $\{A\} \{a_1, a_2, a_3\}$ EnumeratedClass(A a_1, \dots, a_n)

图 12 枚举值约束的转换规则

由于 OWL 未提供定义新数据类型的功能,不能表达取值范围约束。原子数据类型的取值范围约束都可以表示为取值区间,可表达为 $x < b$ 、 $x \leq b$ 、 $x > a$ 、 $x \geq a$ 这四种值比较及 $a < x < b$ 、 $a \leq x < b$ 、 $a \leq x \leq b$ 、 $a < x \leq b$ 这四种组合。引入四个易于实现的 OWL 内置数据类型特性来表达值比较运算符: MoreThan($>$)、MoreThanOrEqual(\geq)、LessThanOrEqual(\leq) 和 LessThan($<$),值类型是这些特性的定义域,区间边界值是值域。如此一来就可以使用这些特性来表达取值范围约束,其 OWL 公理表达如图 13 所示。

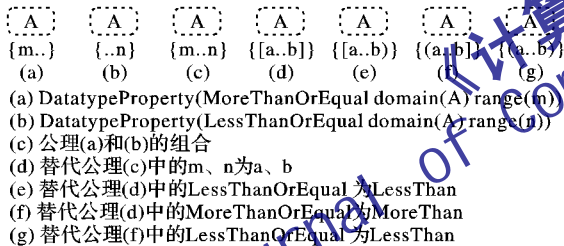


图 13 范围值约束的转换规则

对于限定在角色上的取值约束,无论是枚举值约束还是取值范围约束,我们可以将其等价转换为一个具有值约束的值类型,并让该值对象担当角色,新的值类型是原来担当角色的值类型的子类。等价转换模式如图 14 所示。

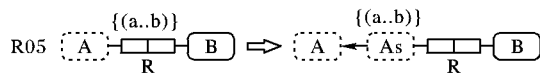


图 14 角色值约束的等价转换

等价转换后的模型可以使用 OWL 公理表达。

ORM 样例模型中值约束相应的 OWL 公理表达如下:

-- 删除前面给出的“称呼”的类公理,改为枚举类

EnumeratedClass(称呼“先生”,“女士”)

-- 增加“分数”类的取值范围约束公理

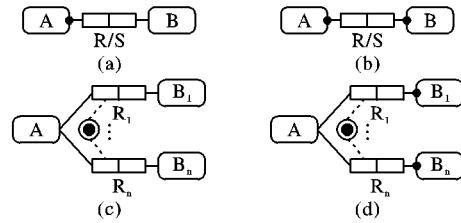
DatatypeProperty(MoreThanOrEqual domain(分数) range(1))

DatatypeProperty(LessThanOrEqual domain(分数) range(5))

4.2 强制约束

强制约束分为单角色强制约束和多角色析取强制约束。单角色强制约束的语义是对象类型的每个实例必须担当相应的角色,与 OWL 的类取值限定公理语义相同;多角色析取强制约束的语义则是对象类型的每个实例必须担当多个角色中的至少一个,即类的取值限定是这些角色取值范围的并集。

强制约束的 OWL 公理表达如图 15 所示。



(a) $\text{EquivalentClasses}(\text{restriction}(R \text{ someValuesFrom}(B)) A)$
(b) $\text{SubClassOf}(\text{restriction}(S \text{ allValuesFrom}(A)) B)$
(c) $\text{EquivalentClasses}(\text{restriction}(R \text{ allValuesFrom}(B)) A)$
(d) $\text{EquivalentClasses}(\text{restriction}(S \text{ allValuesFrom}(A)) B)$
(c) $\text{EquivalentClasses}(\text{unionOf}(\text{restriction}(R1 \text{ someValuesFrom}(B1), \dots, \text{restriction}(Rn \text{ someValuesFrom}(Bn)) A)$
(d) $\text{EquivalentClasses}(\text{unionOf}(\text{restriction}(R1 \text{ allValuesFrom}(B1), \dots, \text{restriction}(Rn \text{ allValuesFrom}(Bn)) A)$

图 15 强制约束的转换规则

ORM 模型样例的公理化中表达实体类型参与事实的限定约束因强制约束而改变了,需要更改相应的 OWL 公理:

-- 更改因强制约束而改变的限定约束

EquivalentClasses(restriction(图书_具有_名称 someValuesFrom(名称)) 书)

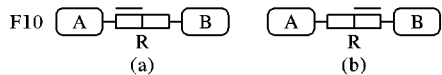
EquivalentClasses(restriction(人_具有_姓名 someValuesFrom(姓名)) 人)

EquivalentClasses(restriction(人_具有_称呼 someValuesFrom(称呼)) 人)

SubClassOf(restriction(撰写 allValuesFrom(图书)) 人)

4.3 唯一约束

ORM 的唯一约束分为内部的和外部的两种,内部唯一约束有可以分为角色上的唯一约束和谓词上的唯一约束。在二元事实类型中,谓词约束是一个平凡的约束,因此只需要考虑角色上的唯一约束。角色上的唯一约束的语义和 OWL 的函数特性语义相同,同时,OWL 的基数限定约束也可以用来表达限定在角色上的唯一约束。但 OWL 没有机制能够表达跨越多个特性的唯一约束,无法形式化表达外部唯一约束。角色上的唯一约束的 OWL 形式化公理如图 16 所示。



(a) $\text{ObjectProperty}(R \text{ Functional domain}(A) \text{ range}(B))$
或 $\text{SubClassOf}(\text{restriction}(R \text{ Cardinality}(1)) A)$
(b) $\text{ObjectProperty}(R \text{ InverseFunctional domain}(A) \text{ range}(B))$

图 16 唯一约束的转换规则

ORM 模型样例包含 4 个角色唯一约束,需要更改相应的 OWL 对象特性公理,更改的公理如下:

-- 更改因唯一约束需要增加函数特性的对象特性公理

ObjectProperty(图书_具有_名称 Functional domain(图书) range(名称))

ObjectProperty(翻译自 Functional domain(图书) range(图书))

ObjectProperty(人_具有_姓名 Functional domain(人) range(姓名))

ObjectProperty(人_具有_称呼 Functional domain(人) range(称呼))

ObjectProperty(打分 Functional domain(评论图书) range(分数))

4.4 频率约束

ORM 的频率约束的语义是对某一角色值在事实中出现的次数限制,要么按照次数出现,要么不出现;OWL 的基数限制类的语义与 ORM 的频率约束的语义相近。ORM 的频率约束有三种,其 OWL 公理表达如图 17 所示。

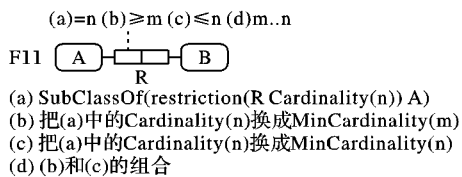


图 17 频率约束的转换规则

ORM 模型样例中参与“图书评论”的“图书”有基数约束限制,应增加相应的 OWL 公理:

-- 频率约束的公理

SubClassOf(restriction(评论 MinCardinality(2)) 图书)

4.5 基数约束

基数约束限定对象类型的一个实例允许出现在事实中的总次数。OWL 中没有直接限定类的实例出现次数的类或公理,我们引入一个适用于任何原子类的三个基数约束特性 cardinalityC、minCardinalityC、maxCardinalityC 来表示类的基数约束,这些特性的定义域是类,值域是相应的基数约束边界。基数约束的 OWL 公理表达如图 18 所示。

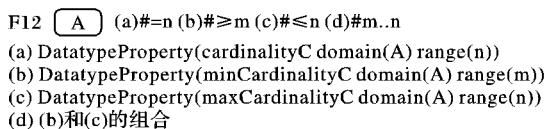


图 18 基数约束的转换规则

4.6 子集约束

子集约束限定处于不同事实类型中的两个角色/角色组的值/元组集合之间的包含关系。限定在两个角色上的子集约束的模型形态如下,其语义是事实集 R 中出现的 A 的实例集是事实集 S 中出现的 A 的实例集的子集,可以用 OWL 的子类公理形式化。

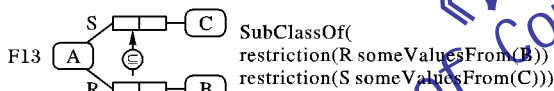


图 19 两角色上子集约束的转换规则

限定在两角色对值集合上的子集约束,在二元事实类型上就是限定在两事实类型上的子集约束,其模型形态如下,其语义是事实集 R 是事实集 S 的子集,即事实类型 R 是 S 的子类,与 OWL 的子特性公理的语义相同,公理表达如图 20 所示。

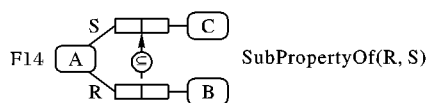


图 20 两谓词上子集约束的转换规则

ORM 模型样例中有子集约束,增加相应的 OWL 公理:

-- 子集约束的公理: 评论某图书的人是阅读该图书的人的子集
 SubPropertyOf(评论, 阅读)

4.7 等价约束

等价约束是子集约束的特例,即双向的子集约束。使用 OWL 中的等价类和等价特性即可表达单角色对和角色组对上的等价约束,形式公理如图 21 所示。

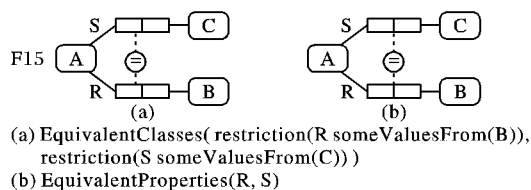


图 21 等价约束的转换规则

4.8 排斥约束

作用于两个二元事实类型中两角色的值集合之间的排斥

约束模型如下,其语义是出现在事实集 R 中的 A 的实例不允许出现在事实集 S 中,与 OWL 的排斥类语义相同,形式化公理如图 22 所示。

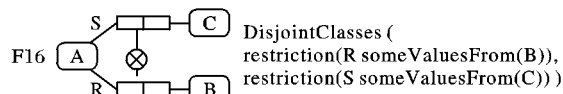


图 22 角色排斥约束的转换规则

作用于二元事实类型上的排斥约束,其模型形态如下,其语义是事实集合 R 和事实集 S 的交集为空,也就是说事实类型 R 和事实类型 S 相互排斥,OWL 类互斥公理具有相同语义,但只有将事实类型对象化才能使用 OWL 的类互斥公理。因为特性互斥的实现代价很小,引入特性互斥 DisjointProperties 公理到 OWL 中,形式化公理表达如图 23 所示。

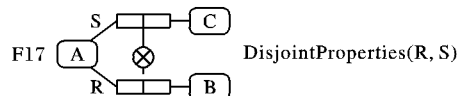


图 23 谓词排斥约束的转换规则

ORM 模型样例中有事实类型排斥约束,增加相应的 OWL 公理如下:

-- 集合排斥约束的公理: 评论某图书的人不能是撰写该图书的人
 DisjointProperties(评论, 撰写)

4.9 值比较约束

值比较约束作用于同一类对象参与的两个二元事实中计量单位相同的另一组角色上。OWL 中没有表达数值比较操作的公理,不能形式化表达值比较约束。

4.10 环约束

环约束的基本类型包括自反性、非自反性、对称性、非对称性、反对称性、传递性、非传递性、无环约束。环约束是对具有环性质的事实类型给予的取值约束,即对关系路径可以形成环路的关系的约束。OWL 中只提供了对称性和传递性两种关系约束,我们可以引入易于实现的环约束到 OWL 中,包括自反性 Reflexive、非自反性 Irreflexive、非对称性 Asymmetric、反对称性 Anti-symmetric 和反传递性 Intransitive 等,ORM 环约束的 OWL 公理化如图 24 所示。

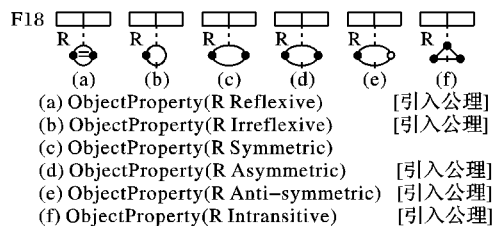


图 24 环约束的转换规则

ORM 模型样例中的无环约束不能表达为 OWL 公理。

5 结语

本文给出了将 ORM 概念模型转换为 OWL 形式化公理的通用规则。通过引入易于实现的运算符和专用特性,对某些模型进行等价变换,除了外部唯一约束、作用于多元事实类型上的约束、值比较约束、无环约束外,其他类型的 ORM 模型都可以用 OWL 的形式化公理描述。

使用 OWL 公理表达 ORM 模型可能会损失部分约束信息,但转换后可以使用描述推理引擎来推理 OWL 公理化的 ORM 模型,以便检查模型的潜在逻辑错误和冗余;同时,还可以将 ORM 模型发布为语义 Web 文档,以便通过语义 Web 共享 ORM 概念模型表达的本体知识。

(下转第 1069 页)

广泛采用的 Alignment API^[7]产生本体匹配文件并与标准匹配文件比较,做评估。测试数据采用 OAEI2008 的 benchmark 提供的测试数据。对比实验数据直接从 OAEI 网站中获取现有的参赛工具的匹配结果。评价指标采用查准率 P 、查全率 R 和 F -Measure 值。

P = 正确的匹配对 / 找到的匹配对

R = 正确的匹配对 / 应有的匹配对

F -Measure = $2PR / (P + R)$

为了充分体现树形结构在本体匹配中的作用,对比实验首先是与只使用了编辑距离的本体匹配工具 EDNA 进行对比。对比结果如表 1 所示。

表 1 EDNA 与本文方法的比较

匹配方法	精确率	查全率	F -Measure
EDNA	0.643 824	0.832 059	0.725 937
本文方法	0.959 941	0.940 294	0.950 016

由表 1 可以看到在利用了本体树形结构信息之后,本体匹配的效果得到了很大的提升。

同时,对比实验还包括与不完全利用了本体树形结构信息的 AgreementMaker 的匹配结果做了比较。比较结果如表 2 所示。

表 2 AgreeMaker 与本文方法的比较

匹配方法	精确率	查全率	F -Measure
AgreMaker	0.984 412	0.916 765	0.949 385
本文方法	0.959 941	0.940 294	0.950 016

由表 2, 可以看到, 与 AgreementMaker 相比, 本文的方法精确率偏低, 而查全率较高。在通过 F -Measure 指标的衡量下, 本文的方法只是略高于 AgreementMaker。然而, 与本文的方法不同的是, AgreementMaker 内部整合了多种本体匹配算法, 如 BSM、PSM^[8]、DSI、SSC 等。它的工作原理就是将各种算法运行的结果进行迭代匹配, 综合多种算法的匹配结果, 能够被多种算法同时发现的匹配成为正确的匹配的可能性自然而然加大, 由此增加了 AgreementMaker 的精确率。于此同时, 本文的方法将会耗时更少, 原因如下: 1) AgreementMaker 自下而上匹配顺序导致了冗余计算; 2) 多种算法的迭代匹配必然增加计算的时间开销; 3) SSC 方法的原理决定其计算量必然大于本文的基于子树范围的匹配原则。因为, SSC 在选取匹配对象时, 将会对源本体树的所以第 i 层子树节点与目标本体树的所有的第 i 层子树节点做匹配, 而忽略这些子孙

节点是否具有相同的父亲节点, 也就是 SSC 的节点选取方法不满足以子树为范围的选取原则。所以对这些节点做笛卡尔乘积式的相似度计算必然存在过多的冗余计算。经过以上分析, 可以看出本文的方法在保证匹配效果的前提下可以有效地减少计算量, 提高了本体匹配的效率。

4 结语

本文的方法在充分利用本体的树形结构特征的同时, 根据本体的 Name, Label, Comment 采用编辑距离计算相似度, 通过实验证明了方法的可行性。进一步的工作将会从本文中并未涉及到本体匹配中运用十分成功的基于 RDF 图的匹配^[9]以及基于结构信息的反馈机制, 如 SF 方法^[10], 等方面入手, 完善本文提到的方法。

参考文献:

- [1] CRUZ I F, ANTONELLI F P, STROE C, et al. Using the AgreementMaker to align ontologies for the OAEI Campaign 2009 [EB/OL]. [2010-09-08]. <http://www.cs.uic.edu/~ifc/webpapers/Cruz-OAEI-2009-new.pdf>.
- [2] CRUZ I F, SUNNA W. Structural alignment methods with applications to geospatial ontologies [EB/OL]. [2010-09-07]. <http://www.cs.uic.edu/~ifc/webpapers/gis-trans-publish.pdf>.
- [3] 简宁胜. 一个本体匹配工具的设计与实现[D]. 南京: 东南大学, 2006.
- [4] 邵志元, 刘大有. 本体匹配算法的研究[D]. 长春: 吉林大学, 2009.
- [5] HP Labs Semantic Web Programme, Jena [EB/OL]. [2010-09-08]. <http://jena.sourceforge.net/>.
- [6] OAEI [EB/OL]. [2010-09-08]. <http://oaei.ontologymatching.org/>.
- [7] EUZENAT J, LIG I. Alignment API [EB/OL]. [2010-09-08]. <http://alignapi.gforge.inria.fr/>.
- [8] CRUZ I F, ANTONELLI F P, STROE C. Efficient selection of mappings and automatic quality-driven combination of matching methods [DB/OL]. [2010-11-13]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.156.5563&rep=rep1&type=pdf>.
- [9] 王颖, 刘群, 王慧强, 等. 一种基于 RDF 图的本体匹配方法[J]. 计算机应用, 2008, 28(2): 460-462.
- [10] MELNIK S, GARCIA-MOLINA H, RAHM E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching [C]// ICDE 2002: 18th International Conference on Data Engineering. San Jose, CA: [s. n.], 2002: 117-128.
- [5] JARRAR M. Towards automated reasoning on ORM schemes mapping ORM into the DLRidf description logic [C]// ER 2007: Proceedings of the 26th International Conference on Conceptual Modeling, LNCS 4801. Berlin: Springer-Verlag, 2007: 181-197.
- [6] JARRAR M. Mapping ORM into the SHOIN/OWL description logic [C]// ORM'07: Proceedings of the International Workshop on Object-Role Modeling, LNCS 4805. Berlin: Springer-Verlag, 2007: 729-741.
- [7] NGUYEN T, THANH N. Modeling ORM schemas in description logics [M]// Complex Systems Concurrent Engineering: Collaboration, Technology Innovation and Sustainability. Berlin: Springer-Verlag, 2007: 547-555.
- [8] WINTRAECKEN J J V R. The NIAM information analysis method: theory and practice [M]. Berlin: Springer, 1990.

(上接第 1066 页)

参考文献:

- [1] HALPIN T. Object-Role Modeling (ORM/NIAM) [M]// Handbook on Architectures of Information Systems. Berlin: Springer-Verlag, 2006: 81-103.
- [2] HALPIN T. ORM 2 [C]// On the Move to Meaningful Internet Systems 2005: OTM Workshops, LNCS 3762. Berlin: Springer-Verlag, 2005: 676-687.
- [3] JARRAR M, MEERSMAN R. Ontology engineering - the DOGMA approach [C]// Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web, LNCS 4891. Berlin: Springer-Verlag, 2009: 7-34.
- [4] OMG. Semantics of Business Vocabulary and Business Rules (SBVR) [S]. OMG, 2008.