

RDFS 扩充性原理及应用

Principle and Application of RDFS Extensibility

李青山 陈 平

(西安电子科技大学软件工程研究所 西安710071)

Abstract RDF defines a frame model for describing and interchanging meta-data and supports semantic interoperability of Web data and knowledge exchange. As a type system for RDF data model, RDF Schema provides a mechanism extending new modeling primitives and their semantic constraints. The ontology and its related model and underlying technology, which can express domain model in knowledge presentation, can make the Web logical. From the view of semantic Web framework, RDFS extensibility mechanism plays an important role in domain-modeling and semantic interoperability in Web environment. Based on the analysis of the structure of hierarchical class model of RDFS, this paper puts forward and discusses deeply the meta-model control principle of RDFS extensibility mechanism. Finally, an instance of RDFS extensibility mechanism, that is, OIL ontology class definition and slot definition on meta-level, is established ground on this principle.

Keywords RDFS, Extensibility mechanism, Meta-model control, Ontology, OIL

1 引言

互联网从早期的静态页面信息发布到 Web 内容的机器动态生成,使得人与人之间不仅可以在互联网上实现信息互通,人与机器也可以互动交流。但前两代的 Web 主要以 TCP/IP 和 HTTP/HTML 为基础支持人与人、人与机器、机器与机器之间的语法上的互操作,很少也很难支持语义互操作。被称之为“语义化互联网(Semantic Web)”^[1]的第三代互联网以 Web 中的信息是机器可处理的信息为目标,以机器可以理解互联网中的信息和知识以及实现不同机器之间文档或数据语义互操作为特点^[2],最后,实现互联网中信息和知识不仅可以发布和生成,而且可以进行语义校验、机器推理、形式证明,真正让 Web 形式化和语义化。

语义化互联网的发展,本质上就是不同层次上表达机器可理解、可处理的信息的语言的发展。TCP/IP 协议给出了物理上字节流传输的统一标准,HTTP 和 HTML 提供了超文本文档传输和呈现标准方式。XML 可以通过定义特定领域的标记为不同应用系统之间定义可以实现内容交换和共享的文档结构,为 Web 信息互操作提供简单的、功能强大的、标准的语法支持^[3~4]。尽管 XML 为 Web 内容个性化和统一化提供了语法上的标准支持,通过 XML Schema,也可以支持一定程度的数据语义表达^[5~6]但在表达 Web 上的知识方面,XML 不具有提供信息语义互操作的能力。RDF 模型在 XML 层次之上为 Web 中信息表达和处理提供语义化支持。

资源描述框架 RDF 是一种描述和交换元数据的框架^[7],它提供了在互联网中交换机器可理解信息的应用系统之间的互操作性,强调了对互联网资源的自动处理。基于 RDF 的有向标记图数据模型,可以表达 Web 上各种资源之间内在的语义关联,同时,在推理规则的支持下,可以进行这些元数据之间的关系推理,产生新的知识,真正实现语义化互联网的目标。在表达语义方面,RDF 提供了一种统一的数据模型,这种

模型为 Web 上的知识表达、知识推理和知识校验提供了底层的标准。要实现语义化互联网的目标,即信息机器可理解、可处理及信息在语义层次上的互交换和互操作,仅仅有 XML 和 RDF 的语言和模型的支持是不够的。在这些语言之上,还应该表达领域模型以及推理和逻辑的语言或者标准。传统知识表示领域中的本体(ontology)概念^[8]及其相关模型和技术可以映射到 Web 中来,使得互联网更加逻辑化。向 Web 文档加入逻辑,可以支持使一类文档演绎出另一类文档的规则,可以支持基于一套约束规则的文档语义检查以及支持从语义级查询而不仅仅是字符匹配。表达领域模型的本体在 Web 程序之间知识的处理和共享中扮演了很重要的角色。Web 中知识本体交换语言的研究促进了互联网中形式化语义的引进。OIL (Ontology Interchange Language 或者 Ontology Inference Layer)实现了传统知识表示中本体及其相关推理技术在互联网的应用,很好地集成了语义化互联网中逻辑层与模型层,为互联网的真正形式化和语义化提供了更高层次的支持。

RDF Schema^[9]为 RDF 数据模型提供了类型系统支持,可以定义不同领域的核心词汇及它们之间的关系,基于其元模型及其扩充机制,可以定义领域 RDF 框架以及上层的本体描述语言。OIL 就是 RDFS 的一种扩充实例。在语义化互联网层次架构中,RDFS 扩充机制为互联网应用的领域建模及语义层次上信息互交换和互操作提供了关键支持。本文提出了 RDFS 扩充机制的层次化元模型控制原理,并应用该机制实例化生成本体交换语言 OIL。

2 RDFS 类型系统元模型控制原理

2.1 RDFS 类型系统

RDF 数据模型定义了非常有限的基本的建模原语,它既没有给出定义新的属性词汇的机制,也没有给出定义这些属性以及资源之间新的关系的机制。RDFS 不仅能定义资源的

李青山 博士研究生,讲师,主要研究领域为互联网应用模型、网络语言、面向对象软件工程和逆向工程。陈 平 教授,博导,研究领域为电子信息系统软件开发理论与技术、软件体系结构和面向对象软件工程。

属性词汇,还能定义这些属性词汇可以描述那些类型的资源以及其取值的范围约束。从职能范围上讲,RDFS本身并没有定义与应用领域有关的词汇,它仅仅是提供了一组核心概念(类型系统)和一套机制,这种机制为后续的进一步领域建模提供基础支持。

本质上讲,RDFS是一套模式规范语言(Schema Specification Language)。它不仅定义了可以扩充到不同领域的核心概念以及这些概念的层次和实例关系,充当元模型,而且提供了扩充机制,可以由核心模型中的层次化类型系统派生出特定领域的主要词汇以及词汇关联和附加在这些词汇本身以及词汇关联上的约束,进行领域建模,形成可以定义和描述特定应用领域的领域建模语言,充当模型。这种建模语言可以以实例化的形式描述具体的应用领域中本体、本体关联以及相关约束。

RDFS在支持Web语义化的语言层次结构中处于核心位置,如图1所示。RDF是XML的一个应用,RDFS用RDF来表达。基于RDFS,可以定义特定领域的核心词汇,形成领域建模语言,其中OIL是用于描述和交换本体的语言,是RDFS类型系统一个典型的扩充性应用实例。由于RDFS在RDF之上定义了基本的本体建模原语(Ontological Modeling Primitives)^[10],RDFS的扩充性加上本体本身的特点,形成了能充分表达语义的互联网环境中的建模语言。

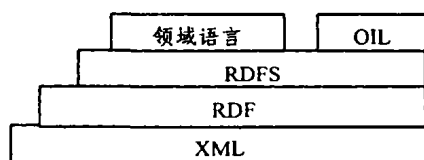


图1 语义化 Web 的语言层次结构图

2.2 RDFS 类型系统的类层次模型

RDFS由一系列核心词汇组成,这些词汇首先被分成若干类,类的层次关系反映了类中元素之间的依赖关系。RDF表达式所要描述的所有对象元素都被看作是 `rdfs:Resource` 类的实例。`rdf:Property` 是所有描述 `rdfs:Resource` 类中实例资源的所有属性的集合,它是 `rdfs:Resource` 类的一个子集。`rdfs:Class` 类用于描述类型或种类的概念,任何一个新的类都是 `rdfs:Class` 的一个实例元素,包括它自己。`rdfs:ConstraintResource` 类是 `rdfs:Resource` 的一个子集,它的实例是一些表达约束信息的资源个体,这些能描述约束的元素能被RDF处理器用来校验RDF数据模型的正确性。`rdfs:ConstraintProperty` 是 `rdfs:ConstraintResource` 和 `rdf:Property` 的公共子集,其实例是一些定义约束的属性。`rdfs:ContainerMembershipProperty` 类中的实例用于定义容器模型中成员元素的若干属性。`Bag`、`Sequence`、`Alternative` 是RDF定义三类容器对象。`Bag` 容器定义了一个资源或原子文字的无序列表,它用于表达一个属性有多个值的情况,并且值的次序没有明确语义。`Bag` 内值无序、可以重复、集合可以为空。`Sequence` 容器定义了一个资源或原子文字的有序列表,它用于表达一个属性有多个值的情况,并且值的次序有明确语义。`Sequence` 内值有序、可以重复、集合可以为空。`Alternative` 容器定义了描述特定属性的一组可选的值的集合,集合不可以为空。

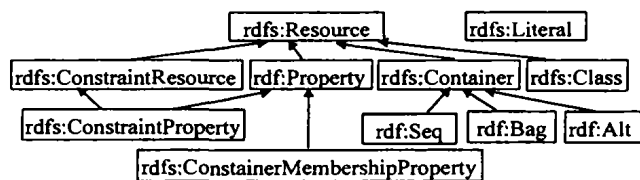


图2 RDF(S)类层次关系图

2.3 RDFS 类型系统的扩充机制

基于RDFS核心类型系统,一方面,可以直接通过类的继承或元素实例化扩充生成新的类和新的资源,这些新的元素可以与领域相关,形成一套领域模型。这种关系可以认为是从元模型到模型的派生过程。基于这种领域模型,可以实例化出应用相关的RDF数据模型。另一方面,通过直接对RDFS的扩充,生成某种本体描述语言(Ontological Descriptive Language)^[11]。这种语言的核心词汇可以由RDFS生成,比如后面要分析的OIL即是这种层次的扩充。然后在扩充后生成的语言的基础上,可以定义与领域相关的一套词汇。这两个层次的转换和映射可以认为是从元元模型到元模型,再从元模型到模型的过程。RDFS的扩充模型的层次架构如图3所示。

本质上看,RDFS是用RDF描述的,所以还有一套定义RDFS本身的更核心的类型系统,这种自描述和自生成的机制为RDF(S)模型的统一性和扩充性提供了基础保证。因此,RDFS也可以被看作是一种本体建模语言。

RDFS的扩充能力体现在两个方面:词汇的扩充、约束的扩充。词汇的扩充有两种形式,一是通过在核心类型系统基础上通过继承生成新的类;二是以实例化的方式生成新的元素。约束的扩充既包括在原有类型层次关系基础上扩充新的关于词汇以及词汇关系的约束,又包括对新扩充词汇引进词汇内涵语义约束及新词汇与其他概念之间关系的约束。这些约束的扩充可以利用RDFS核心类型系统的语义约束机制直接定义或根据规则推导出来。

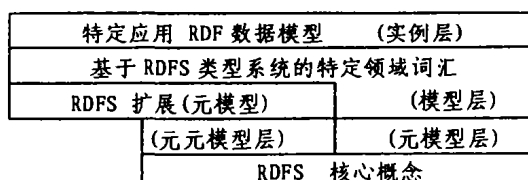


图3 RDFS 一般扩充模型层次架构图

3 RDFS 扩充机制的一个实例化——OIL

RDF模式提供了一套机制用于定义互联网元数据描述中的词汇、结构和约束规则,但它没有定义这些建模原语的形式化语义,因此这些原语的表达能力不足以用到语义化的本体建模和推理。在RDFS模式层之上,需要定义由更强表达语义能力的语言来描述和交换互联网中的语义信息。在语义化互联网的层次模型中,模式层之上应该是逻辑层,用于提供语义和推理支持。如何把形式化的语义表达用到互联网环境中呢?

OIL是一种本体定义和描述语言^[12,13],也是一种知识表示语言,是在描述逻辑(DL)、框架和互联网标准协议的基础上构造的,本身能提供形式化语义描述和高效的推理支持。基于本体本身的信息共享和公共性特点,本体建模语言应用到互联网中信息描述和交换是一个自然的思路。如果用OIL来

完成语义化互联网中逻辑层所要求的功能,就可以把 OIL 本身所具有的丰富的框架原语和强大的语义表达能力运用到互联网中,提高 RDF 模型在表达语义和支持推理方面的不足。

OIL 与语义化互联网的结合点存在在于可以把 OIL 作为 RDFS 类型系统的扩充,即把 OIL 作为 RDFS 的一个应用实例。原因在于 RDFS 本身就是一种本体建模语言。这种结合充分地把 OIL 的能力集成到基于 RDF 数据模型的语义化互联网环境中,为互联网的语义化提供了更高一个层次的支持。下面详细地描述了用 RDFS 如何定义 OIL,既要用到 RDFS 中已有的词汇和约束,也要扩充新的词汇和约束以形成 OIL 的建模原语。

3.1 OIL 本体定义的结构

OIL 中一个本体包括两部分:本体容器和本体定义。本体容器用于描述一个本体的特征,也就是一些元信息。本体容器用 DC 元数据元素集合词汇集标准来描述。本体定义用于定义实际的本体词汇及其描述。这部分定义的词汇可以实例化后产生与特定应用相关的信息。本体定义描述了带有明确语义的结构化的词汇。从层次上看,从本体容器到本体定义再到定义的实例化,可以看作从元元层到元层再到对象层。OIL 语言主要定位在元元层和元层,核心是元层的一套定义规范和方法。

OIL 的一个本体定义包括:并入(Import)、规则基(Rule-base)、定义(Definition)。并入属于可选部分,是一个指向其他 OIL 模块的引用列表,这些 OIL 模块可以被引用并加入到当前的本体中。规则集属于可选部分,是一个应用到该本体上的一个规则列表,这些规则定义了一些公理或者全局约束。一个规则元由一个类型名和一套无结构规则描述组成。定义由零个

或多个类定义和槽定义组成。一个类定义(class-def)关联了一个类名及对应类描述。类描述包括定义类型(primitive 或者 defined)、父类说明(subclass-of)和零个或多个槽约束(slot-constraints)。其中,如果类的类型是 Primitive,则说明类的约束条件对类中成员来讲是必要条件,而不是充分条件;如果子类型是 defined,则说明类的约束条件对类中成员来讲是充要条件。父类说明是一个类表达式(class-expression)的列表,类表达式可以是类名、槽约束或者表达式用与非或连接形成的逻辑组合。槽约束由一个或多个加到槽上的约束组成的列表,典型的约束包括 has-value、value-type、max-cardinality n。一个槽(槽也被称为角色或者属性)定义(slot-def)关联了一个槽名及对应槽描述。一个槽描述指明了应用到槽关系中的全局约束。一个槽描述包括父槽说明(subslot-of)、域(domain)和范围(range)限制以及其他关于槽性质的描述,包括反面槽(inverse slot)、传递性(transitive)和均衡性(symmetric)。下面主要分析本体定义中的类定义和槽定义是如何基于 RDFS 扩充机制定义的。

3.2 OIL 本体类定义的 RDFS 元层次扩充

用 RDFS 类型系统定义和描述 OIL 中的本体时,尽可能用 RDFS 已有的建模原语,但如果 RDFS 中的原语词汇不足以表达本体,则需要扩充 RDFS 以形成新的建模原语词汇。OIL 基于 RDFS 扩充后的核心的建模词汇、这些词汇之间关系以及这些词汇与 RDFS 中核心词汇的层次关系如图4所示。

类定义的类型 primitive 和 defined 描述可以通过在 RDFS 中扩充两个新的类来完成。oil:PrimitiveClass 与 oil:DefinedClass 都是 rdfs:Class 的子类,在 OIL 中定义的类必须是这两个类中某一个的实例。

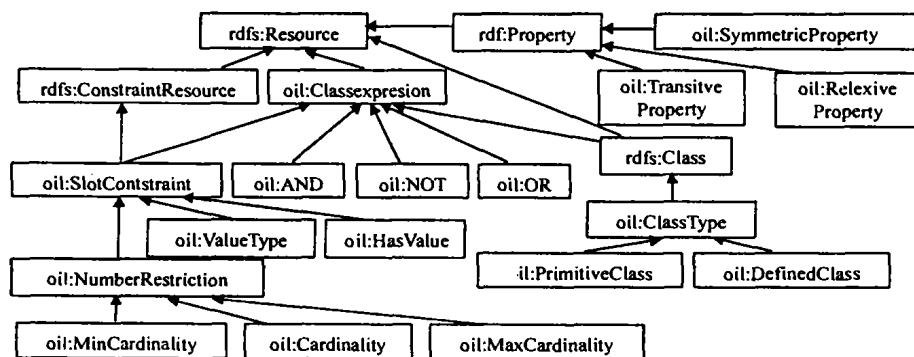


图4 RDFS 扩充后的 OIL 核心建模原语层次关系图

OIL 中类定义的父类说明中,一般父类的值是一个类表达式 class-expression,而类表达式是由逻辑运算符连接起来的布尔表达式,RDFS 中没有表达这种式子的建模原语,需要在 OIL 中对 RDFS 进行扩充定义。引进了一个新类 oil:Class-expression 后,rdfs:Class 可以作为新类 oil:Class-expression 的一个子类出现。这样一来,原来 RDFS 中的一些词汇的语义约束就会发生一些变化,比如 rdfs:subclassof 的取值范围就变为新类 oil:Class-expression。同时,引进一个新属性 oil:has-operand,它为 rdfs:Property 的一个实例,用于连接两个类表达式,指明操作符 AND、OR 与 NOT 的运算对象,这种约束描述如图6。

```
class-def defined herbivore
subclass-of animal,NOT carnivore
slot-constraint eats
value-type plant OR
```

slot-constraint is-part-of has-value plant

图5 一个 OIL 类定义范例

```
<rdf:Property rdf:ID="hasOperand">
  <rdfs:domain rdf:resource="# AND">
  <rdfs:domain rdf:resource="# OR"/>
  <rdfs:domain rdf:resource="# NOT"/>
  <rdfs:range rdf:resource="ClassExpression"/>
</rdf:Property>
```

图6 操作符 AND、OR 与 NOT 约束的 RDF 描述

槽约束描述了类的一些属性。引进新类 oil:SlotConstraint 以及其子类 oil:NumberRestriction、oil:Value Type、oil:HasValue 等,它们都作为类 oil:Class-expression 的子类。类似地,可以引进新属性 oil:hasClass 和 oil:hasSlotConstraint,其中属性 oil:hasSlotConstraint 用于指明一个类的

槽约束,这种约束描述如图7。根据语义, oil:hasSlotConstraint 和 oil:hasOperand 可以看作是 oil:hasClass 的子属性。

```
<rdf:Property rdf:ID="hasSlotConstraint">
  <rdf:type rdf:resource="http://www.w3.org/TR/1999/
    PR-rdf-schema-19990303#ConstraintProperty"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2000/01/
    rdf-schema#Class"/>
  <rdfs:range rdf:resource="SlotConstraint"/>
</rdf:Property>
```

图7 属性 oil:hasSlotConstraint 约束的 RDF 描述

基于以上扩充手段,图5这一小段本体的类定义对应的 RDF 模型描述如图8。

```
<rdfs:Class rdf:ID="herbivore">
  <rdf:type rdf:resource="http://www.ontoknowledge.org/oil/
    rdfschema/#DefinedClass"/>
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    <oil:NOT>
      <oil:hasOperand rdf:resource="#carnivore"/>
    </oil:NOT>
  </rdfs:subClassOf>
  <oil:hasSlotConstraint>
    <oil:ValueType>
      <oil:hasProperty rdf:resource="#eats"/>
    </oil:hasSlotConstraint>
  </oil:hasSlotConstraint>
</rdfs:Class>
```

```
<oil:hasClass>
  <oil:OR>
    <oil:hasOperand rdf:resource="#plant"/>
    <oil:hasOperand>
      <oil:Has Value>
        <oil:hasProperty rdf:resource="#is-part-of"/>
        <oil:hasProperty rdf:resource="#plant"/>
      </oil:Has Value>
    </oil:hasOperand>
  </oil:OR>
</oil:hasClass>
</oil:ValueType>
</oil:hasSlotConstraint>
```

图8 类定义实例对应的 RDF 模型

3.3 OIL 本体槽定义的 RDFS 元层次扩充

OIL 与 RDFS 都把属性放置在本体描述的第一位。OIL 的槽定义与 RDFS 中的属性定义有着直接的映射关系。槽定义中的“subslot-of”,“domain”和“range”也相应地对应着 RDFS 中属性描述中的“subproperty-of”,“domain”和“range”。要注意的是,在槽定义中,属性“range”取值的范围不像 RDFS 中限制只能是一个类中的元素,可以取自多个类。OIL 中槽定义与对应的 RDFS 建模原语对照关系如图9。

OIL 建模原语	slot-def	subslot-of	domain	range	inverse	transitive	symmetric
RDFS	rdf:Pro	rdfs:subPr	rdfs:	rdfs:	oil:inverse	oil:Transitive	oil:Symmetric
词汇	perty	opertyOf	domain	range	RelationOf	Relation	Relation

图9 OIL 中槽定义与对应的 RDFS 建模原语对照表

结束语 RDF 数据模型为互联网中信息的语义互操作提供了底层的数据层和语言层支持,在数据层之上,还需要模式层、逻辑层和推理层。RDFS 基于元模型的扩充机制为语义化互联网中模式层之上逻辑层的数据语义化提供了一致的、自扩充的和反射定义的模型支持。对 RDFS 类型系统反射性的进一步研究以及对 RDF(S)之上的逻辑层和推理层的研究是我们下一步的工作。

参考文献

- Berners-Lee T, Fischetti M. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. Harper, San Francisco, 1999
- Heflin J, Hendler J. Semantic Interoperability on the Web. Proc. Extreme Markup Languages 2000, Graphic Communications Assoc., Montreal, 2000. 111~120
- Kristensen A. Developing HTML based Web Applications. In: 1st Intl. Workshop on Web Engineering WWW7, 1998. <http://www-uk.hpl.hp.com/people/ak/doc/webe98.html>
- Khare R, Rifkin A. XML: A Door to Automated Web Applications. IEEE Internet Computing, 1997, 1(4): 78~87
- XML Schema Part 1: Structures, Specification, World Wide Web Consortium Recommendation. <http://www.w3.org/TR/1999/WD-xmlschema-1-19991105/>
- XML Schema Part 2: Datatypes, Specification, W3C Recommendation. <http://www.w3.org/TR/1999/WD-xmlschema-2-19991105/>

- 91217
- Weibel S. Metadata: The Foundations of Resource Description. D-Lib Magazine, July 1995. <http://www.dlib.org/dlib/July95/07weibel.html>
- Uschold M, Gruninger M. Ontologies: Principles, Methods and Applications. The Knowledge Engineering Review, 1996, 11(2): 93~136
- Brickley D, Guha R V. Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation, World Wide Web Consortium, 2000. <http://www.w3.org/TR/rdf-schema>
- Nejdl W, Wolpers M, Capella C. The RDF Schema Revisited. In: Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik, Modellierung 2000, St. Goar. Foelbach Verlag, Koblenz
- Staab S, et al. An extensible approach for Modeling Ontologies in RDF(S): [Technical Report 401]. AIFB, University of Karlsruhe, March 2000
- Horrocks I, et al. OIL: The Ontology Inference Layer: [Technical Report IR-479]. Vrije Universiteit Amsterdam, Faculty of Sciences, Sept. 2000. <http://www.ontoknowledge.org/oil/>
- Stuckenschmidt H. Using OIL for Intelligent Information Integration. In: V. Benjamins, A. Gomez-Perez, and N. Guarino, eds. Proc. of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conf. on Artificial Intelligence ECAI 2000, Berlin, Germany, 2000. 21~22