# 语义 Web 中的本体推理研究

马硕

# 语义 Web 中的本体推理研究

马硕

（大连海事大学 信息科学技术学院， 大连 中国 116026）

**摘 要** 从语义 Web 的基本概念开始，介绍了语义 Web 的层次结构;介绍了本体的基本概念以及用于本体描述的几种语言。用 W3C 推荐的本体描述语言 OWL 描述了一个本体实例，通过此实例对本体推理在本体建立中的冲突消解、描述优化、本体的合并和实例归类中的应用进行了研究，说明了本体推理在本体建立及本体应用中的作用。本体技术是语义 Web 的核心技术，所以建立和维护本体是语义 Web 中的主要工作之一，而基于本体的推理可以帮助建立和维护本体。

关键词 语义 Web；本体；OWL；推理；W3C

**中图分类号** TP301

## Research on Reasoning on Ontology in Semantic Web

Ma Shuo

（Department of Information Science and Technoloy DaLian Maritime University , Dalian 116026 , China）

**Abstract** After the concepts of ontology and semantic Web are introduced , the layers of structure of semantic Web and several Web ontol-ogy languages are also in troduced .Reasoning on ontology is analyzed through an ontology instance rep resented in OW L .It is the Web on-tology language recommended by W3C .Reasoning on ontology can be used in the building, maintaining and using of ontologies .The on-tology is one of the most important tech nology in semantic Web , so building , main taining and using of ontologies are also the main w ork . Reasoning on ontology can help people to build , maintain ontologies .

**Key words** semantic W eb ;ontology ;OWL ;reasoning;W3C

## 1 语义 Web

万维网(WWW)已经改变了人们获取和发布信息的方式、人们的通信方式以及商业交易方式。但其构成的庞大的信息网也给使用者带来了很多问题和苦恼。面对日益增多且只在展示时才采用简单链接方式的信息，人们越来越抱怨难以找到有用的信息及有效地维护这些信息的方法，于是语义 Web 应运而生。

语义 Web 被称为第二代 Web ，下一代 Web ，由 Tim Berners -Lee 在 2000 年第一次提出，目标是使得 Web 上的信息是计算机可理解的，从而的信息是计算机可理解的 ，从而实现机器自动处理信息,同时他提出了语义 Web 的层次结构 ,如图 1 所示。

该结构一共有七层 ：

(1)Unicode 和 URI ， 是整个语义 Web 的基础 ， Uni-code 处理资源的编码，URI 负责标志资源 ；

(2)XM L +NS +xmlschema ， 用于表示数据的内容和结构;

(3)RDF +rdfschema ,用于描述 Web 上的资源及其类型;

(4)Ontology Vocabulary ， 用于描述各种资源之间的联系 ；

(5)～ (7)是在下面四层的基础上进行的逻辑推理操作。

核心层为 XM L ， RDF ， Ontology ,这三层用于表示 Web信息的语义。而这三层中又以Ontology

层为中心。语义 Web 在信息的组织和描述上注重对信息语义的刻画 。

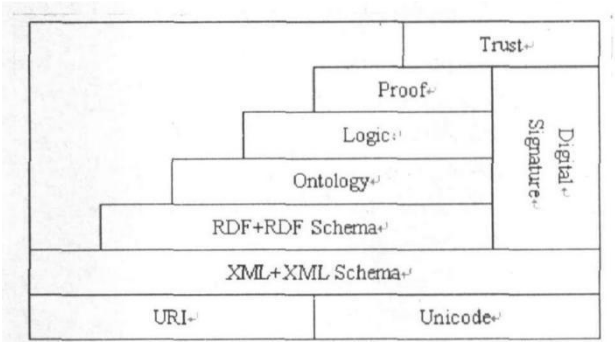


图 1 语义 Web 的层次结构图[1]

## 2 本体 Ontology

本体原是一个哲学上的概念，用于描述事物的本质,在近几年作为信息抽象和知识描述的工具被计算机领域所采用。近年来，人们将本体的概念引入人工智能、知识工程和图书情报领域，用以解决信息提取、知识概念表示和知识组织体系方面的有关问题。利用本体思想从不同角度对信息集合进行标引，表示信息内容与知识组织体系之间的链接关系，可以将本体与信息系统进行链接，从而使用户在使用信息的过程中更加便捷地浏览和理解相关概念和资源，还可以利用本体中的语义关系及推理规则集合进行推理，从而实现基于本体的智能分析和知识组织，并通过智能分析来预测知识增长点。同时，由于本体描述信息的语义，并采用一定的编码语言让计算机可以读懂，更加有利于实现智能检索和查询。可以说，本体是机器自动推理和智能化高级信息服务的基础。

本体构建是一项复杂的系统工程，需要领域专家和知识工作人员以及系统分析人员等各方人员按照一定的方法、采用适当的工具协作完成。由于目前本体构建大都针对某一特定问题域，因此，领域本体的构建也代表了本体构建的思想和方法。

关于 Ontology 很多人给出了不同的理解，其中 Gruber 的定义得到众多认可 ，Ontology 是概念模型的明确规范的描述。Ontology 是面向特定领域 ， 描述特定领域的概念模型即关于该领域的一个公认的概念集，其中的概念有公认的语义 ,通过概念之间的关联来体现[2] 。

关于本体的描述语言有很多种 ,OWL 是 W3C 推荐的本体描述语言。它是一种较新的语义网本体描述语言 ， 由 W3C 的 Web 本体工作小组开发[3] 。OWL 语言按表达力从低到高包括三个子语言:OWL Lite ， OWL DL 和 OWL Full 。OWL Lite 用于提供给那些只需要一个分类层次和简单约束的用户;OWL DL 用于支持那些需要最强表达能力而需要保持计算完备性(computational completeness ，

即所有的结论都能够确保被计算出来)和可判定性(decid-ability ,即所有的计算都能在有限的时间内完成);OWL Full 支持那些需要尽管没有可计算性保证 ， 但有最强的表达能力和完全自由的 RDF 语法的用户[4] 。

## 3 基于 OWL 的本体推理

推理的一个基本内容就是由给定的知识获得隐含的知识，在本体中的推理从根本上说就是把隐含在显式定义和声明中的知识通过一种处理机制提取出来。但是，盲目地获得隐含的知识对建立和使用本体帮助不大 ,所以要先分析清楚 Web 本体推理的应用需求，才能有效地组织推理机制。本体的推理有多方面的应用:对于本体的建立者，推理的主要作用是检测冲突 ,优化表达和本体融合 ；对于本体的使用者，推理的作用主要是获得本体中的知识和运用本体中的知识解决问题[5] 。下面结合一个本体的实例具体讨论 ：

```
<owl:Class rdf:ID=“ Plant”
  > <owl:disjointWith>
    <owl:Class rdf:ID =“Animals”/
  > </ owl:disjointWith >
  <rdfs:subClassOf >
    <owl:Class rdf:ID =“ Biology”/
  > </ rdfs:subClassOf >
</owl:Class>
<owl:Class rdf:ID=“ Mother-Animals”
  > <rdfs:subClassOf >
    <owl:Restriction>
      <owl:minCardinality rdf:dataty pe =“ http://www .w3.
org/ 2001/XM LSchema #int” >1</owl:minCardinality >
      <owl:onProperty >
        <owl:ObjectProperty rdf :ID
      =“hasChild”/ > </ owl:onProperty >
```

```
    </owl:Restriction
  > </ rdfs:subClassOf
  > <rdfs:subClassOf
  >
    <owl:Class rdf:ID =“ Male -animals”/
  > </ rdfs:subClassOf >
</owl:Class>
    <owl:Class rdf :about=“ #Animals”/
  > </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf :about=“ #Male-animals”/
  > </owl:disjointWith>
</ owl:Class>
<owl:Class rdf :about=“ #Animals” >
  <owl:disjointWith rdf:resource =“ #Plant”/ >
  <rdfs:subClassOf>
    <owl:Class rdf :about=“ #Biology”/
  > </rdfs:subClassOf>
</ owl:Class>
<owl:Class rdf :about=“ #Biology” >
</ owl:Class>
<owl:Class rdf :about=“ #M ale-animals” >
  <rdfs:subClassOf rdf :resource
  =“ #Animals”/ >
  <owl:disjoint With rdf:resource=“ #Female-animals”/
> </ owl:Class>
<owl:ObjectProperty rdf:about
  =“ #hasChild” > <rdfs:domain
  rdf:resource =“ #Animals”/ > <rdfs:range
  rdf:resource =“ #Animals”/ >
</ owl:ObjectProperty >
<M ale -animals rdf :ID =“ Tiger1”/ >
```

以上是用 OWL 语言定义了一个简单本体(不是全部代码 ,列出了中心定义)。其中包括六个类:biology , ani-mals ,plant , male_aniamls , female_animals , Mother_animals ; 一个属性 :hasChild ;其中 animals , plant 是 biology 的子类, male-animals , female-animals 是 animals 的子类 ;animals 和 plant 是没有交集的两个类 ,也就是说 aniamls 里的元素都是非 plant 类的, plant 里的元素也都是非 animals 类的;同样 male-animals 和 Female-animals 同样是没有交集的两个类;Mother-animals 是 Female-animals 的子类 , 并且 Mother_animals 有一个属性 hasChild ;还声明了一个 Male_animals 的实例:Tiger1 。

### 3 .1 本体中的冲突消解

本体建立者要想建立正确、一致的本体就需要借助推理 ,这一般由推理机完成。所以好的推理机应该能检查出本体中的冲突 ,一般包括实例体系的冲突和定义体系的冲突。例如 Tiger1 是上面本体的一个实例, 可是后来又有其它本体建立者在这个本体中加入以下的实例:

<Female-animals rdf:ID =“ Tiger1”/ >这句代码声明了 Tiger1 是 Female_animals 的实例。因为 Tiger1 前面已经做为 Male-animals 的实例了, 而 Male-animals 和 Female-animals 是无交集的类 , 所以此时就会产生不一致的情况, 形成了一个实例体系的冲突。同样在定义体系中也会存在冲突, 而且这种冲突往往会导致更加致命的错误。所以要用推理机检测这样的冲突, 目前已经有了这样推理机工具 , 例如 RACER[6] 。它提供的主要服务就是检测一个类是否是另一个类的子类。本体中所有的类中进行这样的检测 ,可以达到消除冲突的目的。

### 3 .2 本体中的描述优化

在建立上面的本体时, 有时可能会出现这样的描述 :

```
<owl:Class rdf:ID=“ Mother-animals”
  > <owl:disjointWith>
    <owl:Class rdf:ID =“ Male -animals”/
  > </ owl:disjointWith >
  <rdfs:subClassOf >
    <owl:Class rdf:ID =“ Biology”/
  > </ rdfs:subClassOf >
</owl:Class>
```

这段代码定义 Mother-animals 类和 Male-animals 类是无交集的, 还定义 Mother-animals 类是 Biology 类的子类。

定义 Mother-animals 类是 Biology 类的子类是多余的, 因为 Mother-animals 类是 Female-animals 类的子类, 而 Female_animals 类是 Biology 类的子类 , 所以 Mother_animals 类肯定是 Biology 类的子类 。同样定义 Mother-animals 类和 M ale-animals 类无交集是多余的 ,因为 Moth-er_animals 类的父类 Female_animals 和

Male_animals 类是无交集的，所以 Mother_animals 类和 Male_animals 类肯定无交集。在一个本体中不可能声明类似的关系，这样会使本体过于庞大，难以实现自动处理。

在实际应用中，类、属性和实例之间的关系是非常复杂的，推理机要能够把这些错综复杂的关系整理清楚，用符合应用需求的格式组织本体中的信息，例如:用树结构来描述类和属性的层次关系，用图来表达实例之间的联系等等。这样在获取本体信息的时候就可以使用成熟的、低复杂度的算法，从而提高效率。本体工具主要有本体的编辑管理工具、本体解析工具和推理机。[3] 最常见的两个本体编辑工具是 Protégé和 KAON，本体解析工具是 Jena，推理机 Racer。尽管手工构造也有一些规范的方法和可用的工具，[4] 然而，构建本体的工作还基本停留在人工或半人工阶段，国内外为数不少的本体构建工具和方法体系还没有一个完全成熟的，因此仍难以进行大范围的本体构建。而且，已有的领域本体研究主要是对本体的编制与构建进行尝试与实验，较少研究相应的编制规范与标准。目前，需要领域专家参与是本体构建方法中的瓶颈，如何通过知识挖掘手段自动获取本体是当前，也是今后一段时期的重要研究领域。

领域本体的应用研究也有广泛的内容。如：中国科学院致力于研究形式化本体在领域知识的复用和共享中的基础和作用、基于专业领域知识复用的虚拟领域本体的构建。[5] 此外，由于本体明确地表达概念及其之间的关系，并且具有推理能力，因此，利用本体可以实现自动推理和智能化高级信息服务。在情报学领域，梁战平提出在本体信息环境下进行信息获取和智能分析的情报学研究模式。[6] 基于本体的信息研究与分析将成为一个新方向。

从整体来看，本体研究目前仍处于理论研究日趋成熟、应用研究相对滞后的阶段。单纯从技术角度描述本体的较多，理论联系实践并在实际系统中应用的领域本体则比较少。

### 3．3 本体的合并

本体合并是指将已经存在两个或两个以的本体合成一个新的本体，在开放的信息环境下，本体的合并显得特别重要，可是手工合成本体显得力不从心。推理机在这方面可以辅助人们完成工作。操作者可以先指定不同本体中的一些同义概念，由推理机针对概念的联系把本体合并。本体的构建是对概念本身以及概念与概念之间的关系进行形式化描述，多是面向特定领域。出于对不同学科领域和具体工程的不同考虑，领域本体构建的过程各不相同。现行的本体构建方法都不是经权威标准化机构认证的方法。比较成功的本体构建项目大多借鉴软件工程方法，产生了一系列诸如面向对象思想（基于 UML）、原型化思想等的本体构建方法，并从系统需求分析出发，明确需求、规范文档、实时评价等方面规范领域本体的构建，[8] 再根据项目自身的特点和专家经验进行。例如现在有另一个表示动物的本体：

```
<owl:Class rdf:ID=“ Mother-creature”
  > <rdfs:subClassOf >
    <owl:Class rdf:ID =“ Female-creature”/
  > </ rdfs:subClassOf >
</owl:Class>
<owl:Class rdf:ID=“ Male-creature”/ >
<owl:Class rdf:ID=“ Father -creature” >
  <rdfs:subClassOf rdf:resource=“ #M ale -creature”/ >
</owl:Class>
```

互等价的类 Female_creature 和 Female_animals 是相互等价的类就行了。下面推理机会自动得出 Father-creature 是 Male-animals 的子类，Mother-creature 类是 Female-animals 类的子类，Male_creatureFemale_creature 无交集，Father-creature 类和 Mother-creature 类无交集等等的信息。当然这些信息中可能会有冗余，需要优化。

### 3．4 实例的归类

实例归类是指把一个已经描述好的实例，要归结到一个最能反应它特征的类中。通常本体的定义也可能不是很复杂、庞大，但是本体的实例会特别众多，所以能通过推理来实现自动归类有着重要的现实意义。例如有如下描述：

```
<Animal rdf :ID =“ Tiger3”/ >
<owl:Individual rdf:about
  =“ Tiger2” > <hasChild
  rdf :resource =“ Tiger3”/ >
</ owl:Individual>
```

加到刚才的本体中。这时交给推理机分类的话，因为 Tiger2 具有 hasChild 属性，在本体定义中只有 Mother-an-imal 有 hasChild 属性，所

以在本体中描述 Tiger2 的具体类应该是:Mother-animal 类。

这种具体的实现方法是构造包括这个实例的最小类(它可能不在本体中，然后在类定义系统中查找这个类的最小父类 ,所得到的就是描述该实例的最具体类[5] )。

在应用本体时，除了这些基本的推理，还可以定义很多附加规则来实现高层次的推理 ，但这些上层的推理主要是针对具体领域、具体应用的。目前还没有一致认可的本体确认和评价的标准，对本体的评估涉及是否满足需求分析阶段所设定的目标，涉及如何正确构建本体，涉及本体及其定义内容的清晰性、一致性、完整性、可扩展性以及灵活性。本体初步构建起来以后，可以据此进行评价和改进。领域本体也是具有生命周期的，在本体初步构建好之后将是长期的本体操作阶段、维护阶段，其间还要持续地进行本体的完善工作，这是因为本体的构建不是一蹴而就的，而是需要不断的改进。

## 4 结 论

本体技术是语义 Web 的核心技术 ，所以建立和维护本体是语义 Web 中的主要工作之一，而基于本体的推理可以帮助建立和维护本体。OWL 是 W3C 推荐的本体描述语言，并且可能成为本体描述的标准语言，所以随着 W3C 的 OWL 标准发布，会有很多基于这种语言的推理技术出现，而这将会进一步引发 Web 上本体应用的快速发展。

## 参 考 文 献


[1] Antoniou G , van Harmelen F .A Semantic Web Primer[ M] . Cambridge , M A , USA :The MI T Press, 2014.

[2] 邓 芳.Ontology 在语义 Web 中的应用研究[ J] .计算机应用研究, 2014(6):97-99.

[3] Horrocks I , Pate l -Schneider P F , van Harmelen F .From SHIQ and RDF to OWL :T he making of a web ontology language [ EB/OL] .2013.http://w ww .cs.man.ac.uk/-horrocks/ Teaching/cs646/ .

[4] W3C .OWL Web Ontology Language Overview[ EB/ OL] . 2014 .http://www .w3.org/T R/ 2004/ REC -owl -features -20140210/

[5] Y. Ding, S. Foo. Ontology research and development: part1-a review of ontology generation［J］. Information Sci－ ence, 2012, 28（2）: 234-260.

[6] M.F. López, etal. Building a Chemical Ontology Using Methonotology and the Ontology Design Environment［J］. IEEE Intelligent System. Jan./ Feb.2015，（1）: 37-46.

[7] IBM Research Center for Software Engineering. Orthog － onal Defect Classification ［ EB/OL ］ ［ .201404-12 ］ http:// www.research.ibm.com/softeng/ODC/ODC.HTM.

[8] S. Staab, etc. Knowledge processes and ontologies［J］.IEEE Intelligent Systems, Special Issue on Knowledge Management, 2015, 16（1）: 26-34.

［9］张玉峰等. 基于 Semantic Web 的个性化网络导航机制［J］. 情报学报，2013，（24）：438-444.