

# Object-Oriented Representation for OWL Ontology Knowledge Base

Shaohua Zhang, Yingzhong Zhang

School of Mechanical Engineering, Dalian University of Technology, Dalian Liaoning  
Email: m18004094680@163.com

Received: Apr. 26<sup>th</sup>, 2018; accepted: May 8<sup>th</sup>, 2018; published: May 15<sup>th</sup>, 2018

---

## Abstract

Ontology technologies can better represent the knowledge implied in various terms and concepts in a structured, computable and shared form. The ontology based knowledge representation method has been widely used in many fields. In view of the lack of efficient OWL (Web Ontology Language, OWL) ontology knowledge processing and integration tools in the engineering information processing, based on the study of the knowledge representation framework, language syntax and document structure based on the OWL ontology, an object-oriented OWL ontology knowledge base representation model is proposed and designed. A method based on XML document parsing platform to parse the OWL documents in a RDF/XML format is implemented, which can convert the OWL ontology knowledge base into an object oriented ontology knowledge base. The presented work lays a foundation for the subsequent engineering semantic information reasoning and knowledge retrieval service.

## Keywords

Ontology, OWL, Knowledge Representation, Knowledge Base, Object-Oriented

---

# OWL本体知识库的面向对象表示

张少华, 张应中

大连理工大学, 机械工程学院, 辽宁 大连  
Email: m18004094680@163.com

收稿日期: 2018年4月26日; 录用日期: 2018年5月8日; 发布日期: 2018年5月15日

---

## 摘 要

本体技术能更好地以一种结构化的、可计算和可共享的形式表示各种术语、概念所隐含的知识, 基于本

体的知识表示方法已经在许多领域得到广泛的应用。针对目前工程信息处理中缺乏高效的OWL (Web Ontology Language, OWL)本体知识处理与集成工具的情况, 本文在研究OWL本体知识表示框架、语言句法和文档结构基础上, 基于面向对象技术, 提出和设计了一个面向对象的OWL本体知识库表示模型; 提出和实现一个基于XML文档解析平台, 对RDF/XML格式的OWL文档实现解析的方法, 将OWL本体知识库转换构建为面向对象的本地知识库; 为后续的工程语义信息推理和知识检索服务打下基础。

## 关键词

本体, OWL, 知识表示, 知识库, 面向对象

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

知识表示是一种机器可解释的真实世界知识的一种描述, 或者说是知识的一组约定, 一种计算机可以接受的用于描述知识的数据结构。基于知识表示可以对感兴趣的领域进行推理, 因此知识表示是实现人工智能应用的前提和基础。传统的知识表示方法如谓词逻辑表示法、产生式规则表示法、语义网络表示法、框架表示法都只能针对具体的知识类型, 无法兼顾知识表示能力和推理能力。因此在知识的共享、重用、维护、扩展方面存在一定的局限性。本体作为一种形式化的, 对于共享概念体系的明确而又详细的说明, 能更好地以一种结构化的、可计算和可共享的形式表示各种术语、概念所隐含的知识, 同时随着语义 Web 技术的发展, 基于本体的知识表示方法已经得到广泛的应用[1]。

OWL (Web Ontology Language, OWL) [2]是由国际万维网组织 W3C 推荐和使用的基于本体的语义 Web 建模语言, 集成语义 Web 规则语言 SWRL (Semantics Web rule language, SWRL) [3]构成目前强大的知识表示工具, 使得知识表示具有更强大表示能力和推理能力。OWL 本体知识表示的基础是建立在语义 Web 框架上的, 包括 XML、XML 模式、资源描述框架 RDF (Resource Description Framework, RDF) [4]、RDF 模式和一阶描述逻辑等。OWL 知识库通常包括本体概念定义库(TBox)、概念标注的知识实例库(ABox)和知识规则库, 主要以 XML/RDF 格式存储。OWL 语言是作为通用的本体知识描述语言标准且能够广泛的集成到推理操作中。在语义表达方面比 XML、RDF、RDFS 具备更强的可供机器理解内容的能力。很多建模工具支持 OWL 语言, 如 Protégé、Jena 的 OWLAPI、Fact++等。随着知识库和本体数量的增长及广泛应用, 高效率地表示和处理本体信息非常重要, 同时要求 OWL 本体知识表示与具体工程中智能信息处理融合。

## 2. 相关研究现状

目前 OWL 知识建模方式主要以开源库 Protégé [5]作为建模工具, OWL API 和 Apache Jena 完成查询和推理, 但采用 Java 语言编写, 由不同的模块组成, 知识推理和语义查询的效率较低, 从其它语言接近 Java API 比较困难, 最近, 采用 C 语言实现的 Redland RDF 框架[6]提供了对 RDF 三元组的解析、存储和查询功能, 但还缺乏对 OWL 的支持; 文献[7]采用 C++开发一个用于存储和查询 RDF 三元组、解析 RDF/XML 文档和公理的 OWLcpp 库; 孔静[8]研究 OWL 在.NET 环境下解析。但总体看, 还缺乏高效、与目前广泛使用的面向对象技术融合的本体语义查询和知识推理的工具, 影响 OWL 本体知识表示技术

的应用。

本文在对 OWL 本体知识表示架构、知识定义、文档格式和句法研究基础上, 采用 VisualC++2010 编程开发平台, 提出和实现了将 OWL 定义的知识概念、实例和规则映射到 C++定义的对象类、对象实例和产生式知识规则, 旨在集成本体技术和面向对象知识表示方法的优势, 弥补当前主流的本体知识库的表示方法在表示能力、知识共享、知识重用、知识维护等方面的缺点, 为后续高效、大规模知识推理打下基础。

### 3. OWL 本体知识库

#### 3.1. OWL 本体

OWL 是 W3C 组织推荐的国际通用的标准本体描述语言。它建立在 XML/RDF 等已有标准基础上, 通过添加大量的基于描述逻辑的语义原语来描述和构建各种本体。所以基于 OWL 建立的本体有丰富的语义表达能力并具有完善的推理机制, 能更清晰完整地表达领域内的概念和概念之间的联系。

OWL 提供了三种表达能力递增的子语言[4]: OWL Lite、OWL DL 和 OWL FULL。OWL Lite 表达能力最弱, OWL DL (DL 表示描述逻辑)能支持需要最强表达能力的推理, 且这个推理系统能够保证计算的一致性和可判定性, 它包括了 OWL 语言的所有成份, 但有一定的限制。OWL FULL 包括了 OWL 语言的所有成份, 并且取消了 OWL DL 中的限制。本文研究和讨论是 OWL DL。

OWL 是建立在 RDF 和 RDF Schema 之上, 采用基于 XML 句法。OWL 文档通常也称为 OWL 本体。一个 OWL 本体由概念类、类的属性和类实例组成的 XML 文档, 因此一个 OWL 本体可以表示为 XML 的层次树结构, 分解为根元素、不同层次的类元素和属性元素。

一个 OWL 本体的根元素是一个 rdf:RDF 元素, 根元素规定该本体定义的一组命名空间。

```
<rdf:RDF
xmlns:OWL="http://www.w3.org/2002/07/OWL#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
```

此外一个本体还可能从一组断言开始, 这些断言包括评述、版本控制和引入其它本体等, 成组地放在 OWL:Ontology 元素下。下面介绍 OWL 本体的类元素和属性元素[4]。

##### 3.1.1. 类元素

OWL 中预先定义了 OWL:Thing 和 OWL:Nothing 两个类, OWL:Thing 位于类层次结构的最顶端, 包含了所有客观存在的事物, OWL:Nothing 代表空类, 位于类层次结构的最底端。用户自定义的类分别对应于领域种最基本的概念。因此, 所有用户自定义的类都隐含地是 OWL:Thing 的子类。

###### 1) 类的定义

类定义可以通过使用 OWL:Class 标签来实现, 例如定义一个名为 associateProfessor 的类, 同时它属于教职人员 academicStaffMember 的子类:

```
<OWL:Class rdf:ID="associateProfessor">
<rdfs:subClassOf rdf:resource="#academicStaffMember"/>
</OWL:Class>
```

###### 2) 类与类之间关系公理

OWL 主要定义如下类公理关系:

- 不相交关系: 用 OWL:disjointWith 显式定义一个类与另一个类不相交, 或者无交集。
- 相等关系: 用 OWL:equivalentClass 显式定义一个类与另一个类是相等或者等价的。

### 3.1.2. 特性元素

类特性也称关系, 要联系一个类与另一个类或数据, 需要定义类的特性。

#### 1) 特性分类

OWL 包含如下两种类型的特性:

- 对象特性: 对象特性可以通过使用 OWL:ObjectProperty 标签来定义。对象特性通常表示两个类的实例之间的关系。
- 数据特性: 数据特性可以通过 OWL:DatatypeProperty 标签来定义。表示类的实例与 RDF 文字或 XML Schema 数据类型间的关系。

#### 2) 特性的定义

无论是对象特性还是数据特性, 在特性定义时, 需要用 rdfs:domain 定义特性关系的域(domain), rdfs:range 定义关系的范围。

OWL 定义了一组特性关系公理, 例如相反关系 inverseProperties、等价关系 equivalentProperty、函数关系 FunctionalProperty、对称关系 SymmetricProperty、和传递关系 TransitiveProperty 等关系。此外, 特性可以通过 OWL:Restriction 来进行限制, 属性限制的类型包含 allValuesFrom、someValuesFrom、OWL:cardinality、hasValue 等类型。

### 3.1.3. 实例元素

实例也称类的个体, OWL 通过类标签或者 rdf:type 来定义。由于实例是类的个体, 因此, 每一个实例具备所属类的全部特性。

实例与实例之间通过类特性建立关系, 大量相互交织的关系构成 RDF 形式的知识图。客观世界的知识不都是能抽象出知识规则, 但是可以表达为实例及其实例之间的关系。

## 3.2. OWL 本体知识库框架

本体以一种结构化的、可计算和可共享的、明确和显式的形式化方法, 通过描述领域内的各种术语、概念及关系来表示领域内的知识, 本体知识库可以用概念  $C$ 、特性  $P$ 、实例  $I$ 、关系  $R$  和公理  $A$  组成的如下 5 元组表示:  $KB = \{C, P, I, R, A\}$ 。

更进一步将上述 5 元组分为如下三个部分:

**TBox:** 是该领域的概念术语的集合。Tbox 收集领域内的知识概念、公理以及概念与概念之间的关系;

**ABox:** 是实例断言的集合。收集由概念标记的一组实例, 即操作知识对象的集合; 实例与实例之间通过特性关系采用 RDF 模式联系起来;

**规则库:** 规则库表示推理的知识产生式规则, 通常采用 SWRL 语言定义。

上述三部分构成 OWL 本体知识库框架, 如图 1 所示。

### 3.3. OWL 领域本体知识库构建

在对领域知识深入分析和大量知识获取基础上, 基于上述领域本体知识库框架和 OWL 语言就可以对领域知识建模。然而完全采用 OWL 语言去定义十分繁琐, 容易出错。一般采用本体建模工具完成本体的形式化描述, 目前应用较多的 OWL 本体建模工具是由斯坦福大学开发的 Protégé 本体编辑器。

在 Protégé 本体编辑器中本体结构以属性的层次目标结构显示, 用户可以通过点击相应的项目创建或

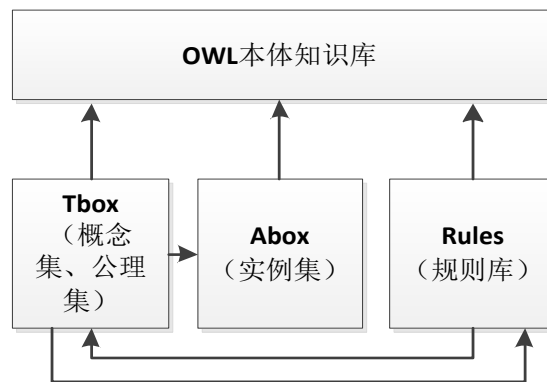


Figure 1. Ontology knowledge frame  
图 1. 本体知识库框架

编辑本体类、子类、属性、实例等, 使用户在概念层次上设计领域模型, Protégé 支持多重集成, 并对新数据进行一致性检查, 并且具有很强的可扩展性, 可以将 Protégé 的内部表示转化成多种形式的文本格式, 包括: XML, RDF(S), OIL, DAML, DAML, OWL 等系列语言。

本文要转换和处理的 OWL 本体就是采用 Protégé 本体编辑器构建的本体, 以 RDF/XML 格式的 OWL 文档文件。

## 4. OWL 本体知识的面向对象表示

### 4.1. 本体知识库面向对象设计

面向对象(Object-Oriented)方法是将客观世界抽象为具有共同属性的对象类及对象个体, 对象与对象之间通过关系联系起来。使用面向对象的表示方法, 具有封装性、模块性、继承性、易于维护和扩展性的优点, 已经广泛应用与各类信息系统的程序开发中。

基于上述 OWL 本体知识库的各元素分析, 可以抽象设计为如下 C++类:

#### 1) 概念类

将 OWL 本体知识库 TBox 中的定义的所有概念的基本属性抽象, 定义为一个 C++概念类: CClassObject, 即 TBox 中的定义的所有领域知识概念映射为 CClassObject 的一个实例对象。

根据上述 2.1 节介绍的 OWL 本体类元素特性, 主要定义如下类属性:

- 类名称。表示概念名称, 用一个 CString 变量表示。
- 命名空间。命名空间表示标识符的上下文或者标识符的作用范围, 这里是指定义的本体概念空间。一个概念标识符可在多个命名空间中定义, 它在不同命名空间中的含义是互不相干的, 一般采用可共享的统一资源描述符定义。
- 父类对象指针, 通过该指针实现 OWL 本体类的子类定义, 形成概念的层次结构。

#### 2) 对象特性类

将 OWL 本体知识库中 TBox 中的定义的属性类别中的对象属性进行抽象, 定义一个 C++对象属性类: CPropertyObject, 即 TBox 中的定义的对象属性描述映射为面型对象类 CPropertyObject 的属性成员。根据上述 2.1 节介绍的 OWL 本体属性元素特性, 主要定义如下对象属性:

- 定义域(domain):

定义域指对象特性关系中对象的范围, 根据上面定义, 只能在 CClassObject 定义的对象或者其子类定义的对象范围内, 因此可以通过定义一个 CClassObject 指针来表示。

- 值域(range):

值域是指定义域中所有对象在该特性法则下对应的所有对象所组成的集合。根据上面定义, 值域也只能在 CClassObject 定义的对象或者其子类定义的对象范围内, 可以定义一个 CClassObject 指针来表示。

### 3) 数据特性类

和对属性映射的机制类似, 不同的是对象属性表示的实例之间的关系, 而数据属性则是描述类实例和 RDF 文字或 XML Schema 数据类型间的关系。根据上述 2.1 节介绍的 OWL 本体属性元素特性, 将 OWL 本体中关于数据属性的描述映射为 CDataObject 中的数据成员。

### 4) 实例对象类

将 OWL 本体知识库 ABox 中的所有个体实例进行抽象, 定义一个 C++ 实例对象类: CInstanceObject, 即 ABox 中定义的所有具体的实例描述都映射为 CInstanceObject 对象。因此每个实例对象都需要一个所属类的指针变量, 以说明该实例的类标识。

部分面向对象表示的类及类关系如图 2 所示。

## 4.2. OWL 文档面向对象解析

前面已经介绍, OWL 文档采用 RDF/XML 格式, 本文采用 TinyXml2 [9] 作为 XML 解析引擎。TinyXml2 是一个基于 DOM (Document Object Model, 文档对象模型), 自由授权处理 XML 文档格式的 C 语言解析器, 定义了对保存为 XML 格式的本体文件数据进行读取、操作的方法。

本文从 XML 文档对象模型层次树的根节点开始, 遍历 OWL 类节点、特性节点、实例节点和规则节点, 逐层对本体文件中相关的数据进行解析, 解析过程如图 3 所示。

通过逐层解析, 将 OWL 中的类、特性、实例和规则转换为 C++ 中类对象、特性对象、实例对象和规则对象, 所生成的对象分别保存在各自的容器中, 实现 OWL 知识本体到面向对象的映射和转换, 为后续的语义推理打下了基础。

## 5. OWL 知识库面向对象表示实例

下面以制造特征自动识别知识库为例来说明 OWL 本体知识库的面向对象表示流程。

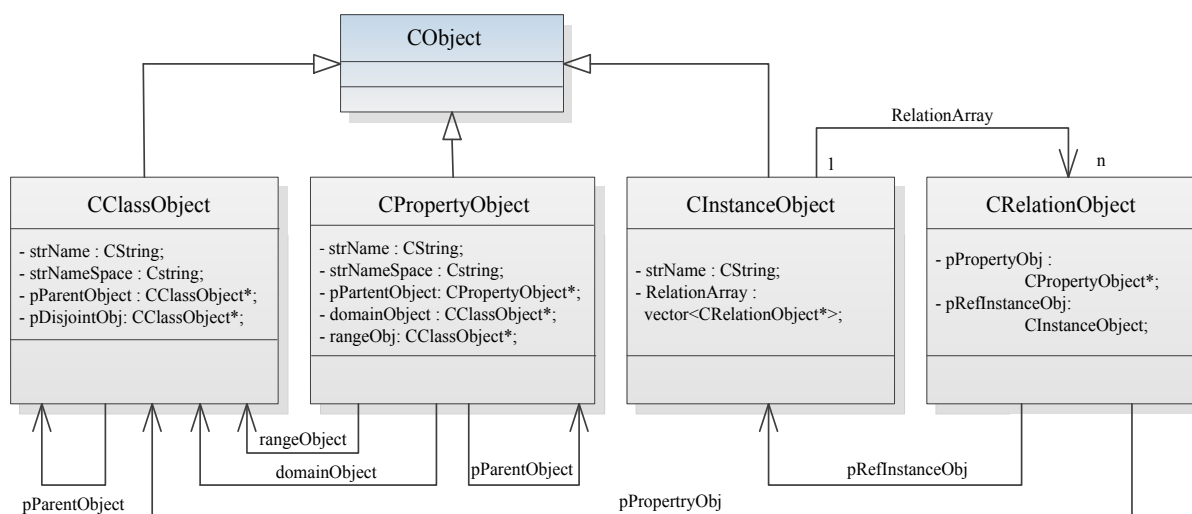


Figure 2. Part class diagram of the object-oriented knowledge base

图 2. 部分 OWL 知识库面向对象类图



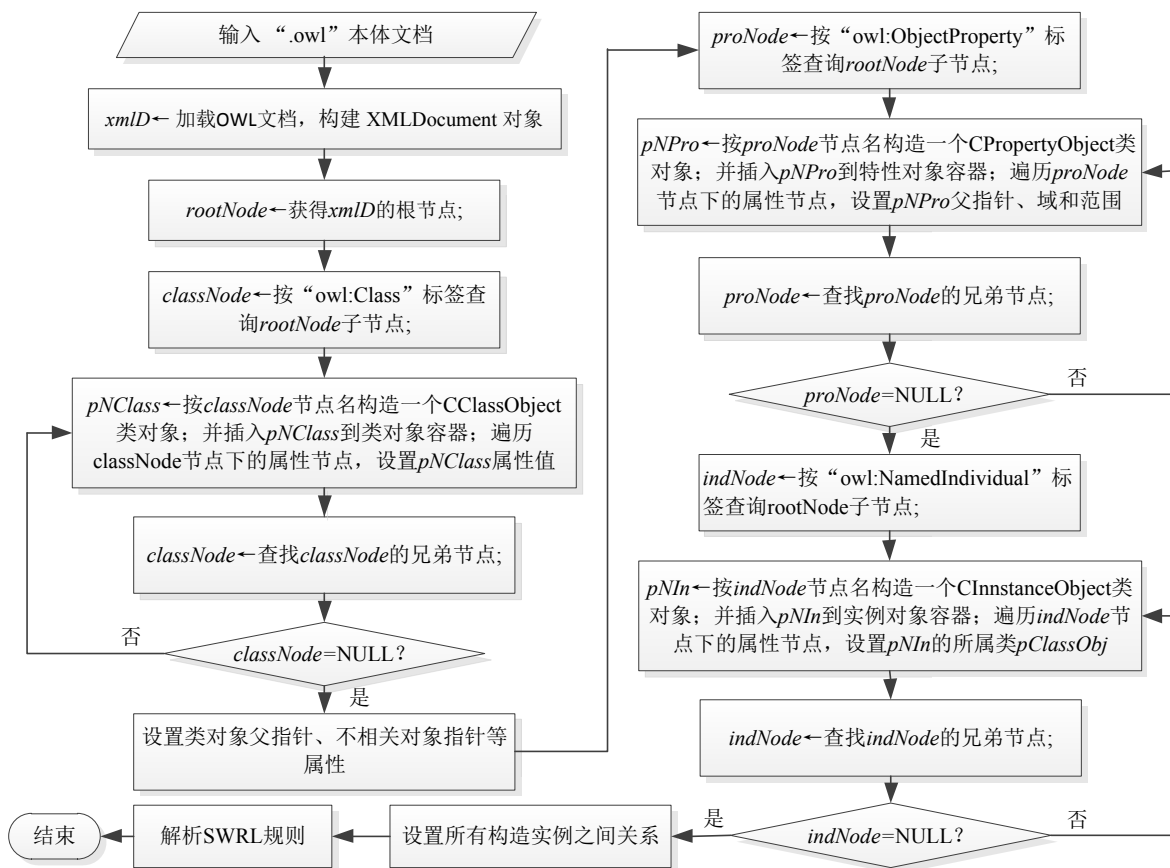


Figure 3. The parsing process of OWL files

图 3. OWL 文档解析流程

### 5.1. OWL 制造特征识别语义知识库

制造特征是机械零件上具有制造意义一组形状, 制造特征语义隐含丰富的关于形状和产品制造知识。本文作者之前提出一个加工特征自动识别的语义方法[10], 基于 OWL 本体技术, 采用 Protégé 构建一个 OWL 制造特征识别语义知识库, 主要由如下三大部分组成:

- TBox: 定义制造特征类(MFeature class)、加工面类(MFace class)、几何类(Geomface class)、对象特性和数据特性。图 4 给出部分内容和层次结构。
- ABox: 在制造特征识别语义知识库中, 实例是对实体边界模型几何面自动识别和语义标注动态生成的个体, 在后续识别推理过程中, 还不断有加工面实例、特征实例生成, ABox 是动态变化的, 因此该实例不讨论 ABox 的对象转换。
- 规则库: 使用 SWRL 语义网络规则语言定义的推理规则, 和实例一起送入推理引擎完成推理过程。通常 SWRL 定义的知识规则保存在单独的规则库文件中, 也可以保存在 OWL 文档中, 目前制造特征识别语义知识库定义了上百条识别知识规则。

上述三大部分内容保存在一个 Feature.OWL 文件中, 图 5 给出部分文档。

### 5.2. 面向对象转换

本文基于 Visual C++ 2010 开发平台和 TinyXml2, 编程实现 OWL 本体知识库的面向对象的转换。转换程序具备加载读取 OWL 文档文件、解析构造本体知识的面向对象表示、编辑修改、添加知识库、序

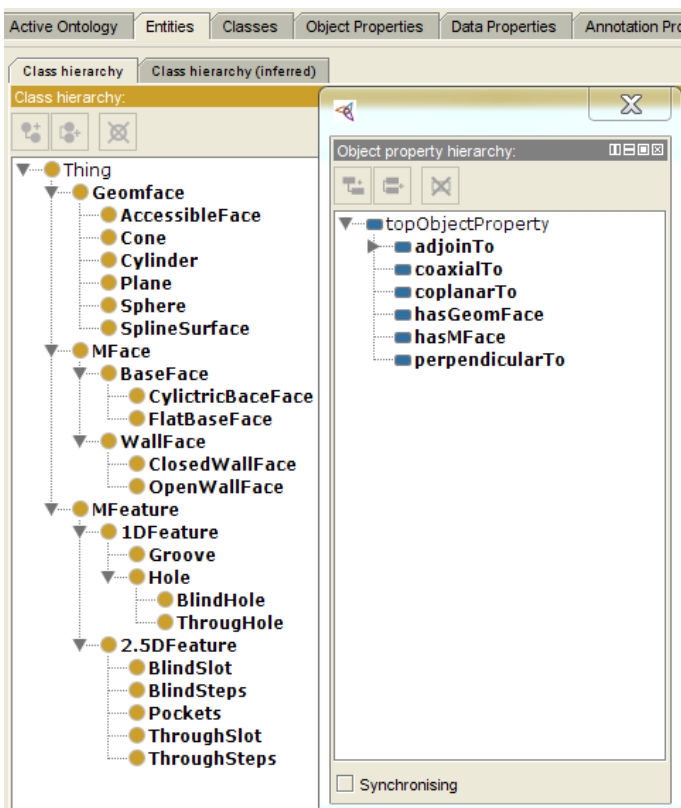


Figure 4. Part manufacture feature recognition ontology in semantic knowledge base class and object features

图 4. 部分制造特征识别语义知识库中本体类和对象特性

```

<owl:Classrdf:about="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#AccessibleFace">
<rdfs:subClassOfrdf:resource="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#GeomFace"/>
</owl:Class>
<!--
http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#AccessibleMFace -->
<owl:Classrdf:about="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#AccessibleMFace"/>
<!--
http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#BaseFace -->
<owl:Classrdf:about="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#BaseFace">
<rdfs:subClassOfrdf:resource="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#MFace"/>
</owl:Class>

<owl:ObjectPropertyrdf:about="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#adjoinTo">
<rdf:typeperdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
<rdfs:domainrdf:resource="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#GeomFace"/>
<rdfs:rangerdf:resource="http://www.semanticweb.org/administrator/ontologies/2016/1/untitled-ontology-31#GeomFace"/>
</owl:ObjectProperty>

```

Figure 5. Owl semantic knowledge base manufacturing identification document motioned

图 5. OWL 制造特征识别语义知识库文档示意



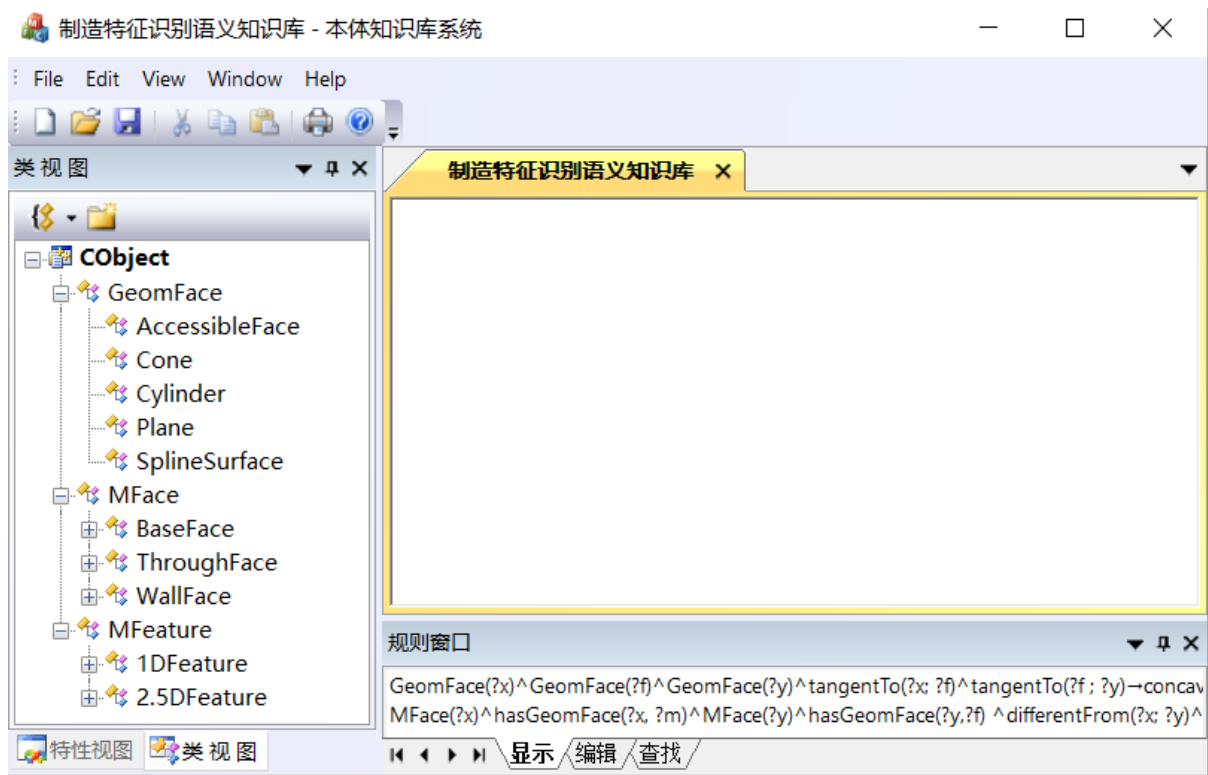


Figure 6. The part of results for the OWL ontology knowledge base to object-oriented conversion

图 6. OWL 本体知识库到面向对象转换的部分结果

序列化保存为 OWL 兼容的 RDF/XML 格式文档。

转换系统将本实例的 OWL 文档“Feature.OWL”打开读入，进行转换后在界面左侧以树结构形式列出解析的概念类对象、对象特性、数据特性；在中间主窗口以图形化方式显示知识库实例对象和关系的 RDF 图。图 6 给出笔者采用 Visual C++ 开发的本体知识库系统界面及部分转换的结果。

## 6. 结束语

本体技术能更好地以一种结构化的、可计算和可共享的形式表示各种术语、概念所隐含的知识，基于本体的知识表示方法已经得到广泛的应用。在工程信息处理中，高效率地表示和集成本体信息非常重要。本文针对目前缺乏高效的 OWL 本体知识处理与集成工具的现状，在研究 OWL 本体知识表示框架、语言句法和文档结构基础上，基于面向对象技术，设计了一个面向对象的 OWL 本体知识库表示模型；基于 XML 文档解析平台，对 RDF/XML 格式的 OWL 文档实现解析，将 OWL 本体知识库转换构建为面向对象的本体知识库，和当前针对 OWL 本体知识库的知识表示相比，采用了统一标准且使用广泛的知识表示方法，提高了解析的效率和可移植性，改善了当前解析技术中对于代码分散、知识层次结构划分不清晰的缺点，使得整个知识表示结构紧凑、逻辑性强，为高效的知识推理和知识检索服务打下了基础。

本文介绍的工作还只是面向对象的本体知识库工作的一部分，下一步还将在 SWRL 规则转换、本体知识库编辑、面向对象的语义推理和知识服务等方面开展更深入的研究。

## 参考文献

- [1] Zhang, Y., Luo, X., Zhang, H. and Sutherland, W.J. (2014) A Knowledge Representation for Unit Manufacturing Processes. *The International Journal of Advanced Manufacturing Technology*, 73, 1011-1031.

- <https://doi.org/10.1007/s00170-014-5864-x>
- [2] Lenzerini, M., Lepore, L. and Poggi, A. (2018) Metaquerying Made Practical for OWL 2 QL Ontologies. *Information Systems*, in Press. <https://doi.org/10.1016/j.is.2018.02.012>
- [3] Abadi, A., Ben-Azza, H. and Sekkat, S. (2018) Improving Integrated Product Design Using SWRL Rules Expression and Ontology-Based Reasoning. *Procedia Computer Science*, **1**, 7. <https://doi.org/10.1016/j.procs.2018.01.139>
- [4] Gryaznov, Y. and Rusakov, P. (2015) Analysis of RDF Syntaxes for Semantic Web Development. *Applied Computer Systems*, **18**, 33-42. <https://doi.org/10.1515/acss-2015-0017>
- [5] Jain, V. and Singh, M. (2013) Ontology Development and Query Retrieval Using Protégé Tool. *International Journal of Intelligent Systems and Applications*, **5**, 67-75. <https://doi.org/10.5815/ijisa.2013.09.08>
- [6] Gregor, D., Toral, S.L., Ariza, T. and Barrero, F. (2012) An Ontology-Based Semantic Service for Cooperative Urban Equipments. *Journal of Network and Computer Applications*, **35**, 2037-2050. <https://doi.org/10.1016/j.jnca.2012.08.002>
- [7] Levin, K.M. and Cowell, G.L. (2015) OWLcpp: A C++ Library for Working with OWL Ontologies. *Journal of Bio-medical Semantics*, **6**, 1-9. <https://doi.org/10.1186/s13326-015-0035-z>
- [8] 孔静. 一种面向.NET平台的OWL解析器的设计与实现[D]: [硕士学位论文]. 青岛: 中国海洋大学, 2010.
- [9] Thomason, L. (2018) TinyXML-2. <https://github.com/leethomason/tinyxml2>
- [10] Zhang, Y., Luo, X., Zhang, B. and Zhang, S. (2017) Semantic Approach to the Automatic Recognition of Machining Features. *The International Journal of Advanced Manufacturing Technology*, **89**, 417-437. <https://doi.org/10.1007/s00170-016-9056-8>

#### 知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>  
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-2286, 即可查询
2. 打开知网首页 <http://cnki.net/>  
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [sea@hanspub.org](mailto:sea@hanspub.org)