

语义 Web 中工程设计类知识表示研究

镇 璐, 蒋祖华, 刘 超, 梁 军

(上海交通大学机械与动力工程学院, 上海 200030)

摘要: 研究了如何在语义 Web 中, 实现工程设计类知识的表示与应用。对工程设计类各种异构知识进行统一描述, 在此基础上, 运用 RDF(S)技术, 建立了语义 Web 环境下, 工程设计类知识的 Ontology。提出了基于语义 Web 的知识存储模型以及面向上层应用程序的接口设计, 为语义 Web 环境下工程设计类知识管理系统的开发提供了底层平台。

关键词: 语义 Web; 工程设计知识; 知识表示; 资源描述框架

Research of Engineering Design Knowledge Representation on Semantic Web

ZHEN Lu, JIANG Zuhua, LIU Chao, LIANG Jun

(School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai 200030)

【Abstract】 This paper focuses on the engineering design knowledge representation on semantic Web, a global knowledge database. It models those categories of knowledge in the unique way. Using the RDF(S) technology, the ontology for engineering design knowledge on semantic Web is built. It brings out a store model for knowledge on semantic Web and its API, which paves the way for developing engineering design knowledge management system on semantic Web.

【Key words】 Semantic Web; Engineering design knowledge; Knowledge representation; Resource description framework(RDF)

工程设计类知识是在工程领域长期发展形成的领域设计知识, 其内容形式多样, 为了实现对混合类型的知识的表示, 本文采用面向对象的知识表示方法; 同时, 为了实现语义 Web 环境下知识的表示, 采用资源描述框架(RDF), 并结合面向对象的思想, 对工程设计类知识构建 Ontology 模型; 最后, 针对该知识表示方法, 提出了具体的存储模式以及面向上层应用程序的接口, 为语义 Web 环境下工程设计类知识管理系统的开发提供了一个完整的平台。

1 工程设计类知识的表示

本文在文献[1]的基础上, 对工程设计类知识, 进行了扩充, 并用 BNF 范式描述如下:

代数式公式类知识的描述

<代数表达式> ::= <symbol> <等号> <expression>

设计约束类知识的描述

<设计约束> ::= <expression> <RelationOperator>

<expression>

<RelationOperator> ::= <|> | <=> | <=> | <=>

模糊规则类知识的描述

<模糊规则> ::= <前提> — <结论>

<前提> ::= 空 ^ <命题> ^ ^ <命题>

<结论> ::= 空 ^ <命题> ^ ^ <命题>

过程类知识的描述

<过程> ::= <过程名> <过程说明>

<过程名> ::= <字符串>

<过程说明> ::= <步骤> [<步骤>,]

实例类知识的描述

<实例库> ::= <实例 List>

<实例 List> ::= <实例>,, <实例>

<实例> ::= <实例名> <参数 List> <参数值 List>

本文采用面向对象的知识表示方法对各式各样的工程设计类知识进行描述, 图 1 是运用面向对象的知识表示方法对工程设计类知识进行描述。

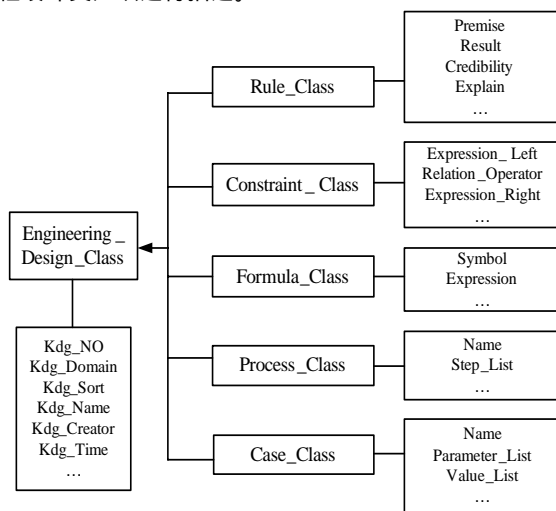


图 1 工程设计知识的类图

2 面向语义 Web 的知识描述语言

W3C 制定的 RDF 模型和语法规范中提出: RDF 是处理元数据的基础, 它提供用于交互 Web 信息的应用程序之间的互操作能力^[2]。RDFS (RDF Schema) 为 RDF 提供了一个类型系

基金项目: 国家“973”计划基金资助项目(2003CB317005); 上海市教委曙光计划基金资助项目(05SG15)

作者简介: 镇璐(1981-), 男, 博士生, 主研方向: 知识管理; 蒋祖华, 教授; 刘超、梁军, 博士生

收稿日期: 2006-06-29 **E-mail:** lzhen@sjtu.edu.cn

统。该类型系统在某些方面类似于C++、Java这样的面向对象编程语言的类型系统。在第1节的基础上,运用RDFS来对工程设计类知识进行描述^[3],以文本文件或者数据库形式存储在RDF存储层;这样就能通过应用程序接口(RDF API)被语义Web应用程序访问、调用。具体框架见图2。本文主要通过研究语义Web环境下工程设计类知识的表示,构建出图2中的下面2层。

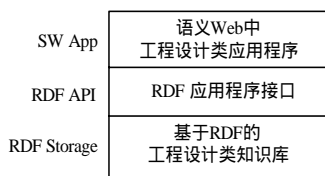


图2 语义 Web 应用程序系统的总体框架

3 基于 RDFS 的工程设计知识表示

3.1 各种知识类以及相互继承关系的定义

在对工程设计类知识,运用 RDFS 进行描述之前,首先要定义并构建一个专门针对工程设计类知识的命名空间,这里用前缀 edk:来表示:

```
Xmlns:edk="http://eng-design-kdg.org/schemas"
```

在 RDFS 中,一个类是任何具有 rdf:type 特性、并且该特性的值为 rdfs:Class 的资源。因此,可以这样来描述一个工程设计知识类:为该类指定一个 URL,比如 edk:Eng_Design,然后将这个工程设计知识类描述为一个具有 rdf:type 特性、并且特性值为 rdfs:Class 的资源,即 eng-design-kdg.org 应该编写如下的 RDF 声明:

```
edk:Eng_Design rdf:type rdfs:Class
```

同理,对规则类、约束类、公式类、过程类、实例类知识进行同样的定义:

```
edk:Rule_kdg rdf:type rdfs:Class
```

...

两个类之间的继承关系可以用预定义的特性 rdfs:subClassOf 来描述:

```
edk:Rule_kdg rdfs:subClassOf edk:Eng_Design
```

...

3.2 知识实例的描述

为了在别处的 RDF 实例数据中引用这些类,eng-design-kdg.org 可采取下列方式:声明一个恰当的 xml:base,该处专门存放工程设计类知识的具体实例,包含具体的设计规则、设计公式等。并书写相对 URL,然后根据这两者所确定的绝对 URL 来标识类;一个规则知识类(edk:Rule_kdg 类)的实例(Oilpump_design_rule1)如下表示:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://eng-design-kdg.org/schemas#"
  xml:base="http://eng-design-kdg.org/instances">
  <ex:Rule_kdg rdf:ID="Oilpump_design_rule1"/>
</rdf:RDF>
```

3.3 知识类中属性的定义的描述

除了描述工程设计知识的类,通常还需要定义刻画这些类的特性(properties)。在 RDFS 中,特性是用 RDF 类 rdf:Property 以及 RDFS 特性 rdfs:domain(定义域)、rdfs:range(值域)以及 rdfs:subPropertyOf 来描述的。下面还是以规则类知识(edk:Rule_kdg 类)为例:

```
<rdf:Property rdf:ID="premise">
```

```
<rdfs:domain rdf:resource="#Rule_kdg"/>
<rdfs:range rdf:resource="#proposition"/>
</rdf:Property>
<rdf:Property rdf:ID="result">
<rdfs:domain rdf:resource="#Rule_kdg"/>
<rdfs:range rdf:resource="#proposition"/>
</rdf:Property>
<rdf:Property rdf:ID="credibility">
<rdfs:domain rdf:resource="#Rule_kdg"/>
<rdfs:range rdf:resource="&xsd:float"/>
</rdf:Property>
<rdf:Property rdf:ID="explain">
<rdfs:domain rdf:resource="#Rule_kdg"/>
<rdfs:range rdf:resource="&xsd:string"/>
</rdf:Property>
```

3.4 一个基于 RDF/XML 的具体知识实例的描述

以一个具体的规则为例,说明其知识的描述方法。

某油泵设计规则:“如果,供油速率前急后缓,且,喷油持续期过长;那么,调压弹簧刚度增大”规则强度:0.95。该规则对应的 URI 为

```
"http://eng-design-kdg.org/instances/Oilpump_design_rule1"
```

2个前提以及1个结果,它们都属于命题类(proposition),其 URI 分别对应于:

```
"http://eng-design-kdg.org/proposition/0001"
```

```
"http://eng-design-kdg.org/proposition/0002"
```

```
"http://eng-design-kdg.org/proposition/0003"
```

这样上述设计规则可用如下 RDF/XML 来描述:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://eng-design-kdg.org/schemas#"
  xml:base="http://eng-design-kdg.org/proposition">
<rdf:Description
  rdf:about="http://eng-design-kdg.org/instances/Oilpump_design_rule1"
  <s:premise>
  <rdf:Bag>
    <rdf:li rdf:resource="#0001"/>
    <rdf:li rdf:resource="#0002"/>
  </rdf:Bag>
  </s:premise>
  <s:result>
  <rdf:Bag>
    <rdf:li rdf:resource="#0003"/>
  </rdf:Bag>
  </s:result>
  <s:credibility>
    rdf:datatype="&xsd:float">0.95
  </s:credibility>
</rdf:Description>
</rdf:RDF>
```

4 基于 RDFS 知识存储模式及应用

4.1 数据库存储

RDF的属性对应于关系数据库的字段名,RDF的属性值对应于关系数据库的字段单元。但是如果直接这样存储,会造成数据库中存储表格的存储稀疏。整个数据表许多字段单元为空,极大地浪费了存储资源^[4]。

因此,直接对 RDF 的三段式(主体-谓词-客体)进行存储,这样数据库就只有3列:subject列(主体),predicate列(谓词),object列(客体)。事实上,每个主体、谓词、客体对应的都是

一个 URI 标识, 考虑到, URI 有重复, 而且通常很长, 字节数较多; 因此, 用另外一张表, 专门存储 URI, 然后给每个不同的 URI 给一个编号, 这样, 前面三段式存储的数据库中, 每个字段单元只需存入一个编号即可。既节省存储空间, 又方便字段的匹配、检索, 如表 1、表 2 所示。

比如: edk:premise rdf:type rdf:Property
edk:result rdf:type rdf:Property

表 1 URI 编号

id	URI
...	
00007	http://www.w3.org/1999/02/22-rdf-syntax-ns/type
...	
00011	http://www.w3.org/1999/02/22-rdf-syntax-ns/Property
...	
00101	http://eng-design-kdg.org/schemas/premise
00102	http://eng-design-kdg.org/schemas/result
...	

表 2 RDF 三元组存储

subject	predicate	object
...		
00101	00007	00011
00102	00007	00011
...		

4.2 RDF 知识存储与上层应用程序接口

以 RDF 形式存储的知识数据, 从存储介质到上层的应用程序之间, 由于不同的存储介质有着不同的存储格式, 因此为了对上层应用程序隐藏具体的存储格式, 以 RDF 三元组 (RDF_Triples) 作为数据格式转换的中介格式, 因为三元组是 RDF 最本质、最简单的表达方式, 如图 3 所示。

文本文件通过 RDF 解析器转换成三元组, 而本文中设计的数据库, 就可以直接以三元组的形式返回 RDF 数据。针对上层的语义 Web 应用程序, 以什么样的数据结构来表示 RDF 数据模型是关键问题所在。图 3 中的 RDF_CoreData 就是连接 RDF 三元组和上层语义 Web 应用程序的桥梁。

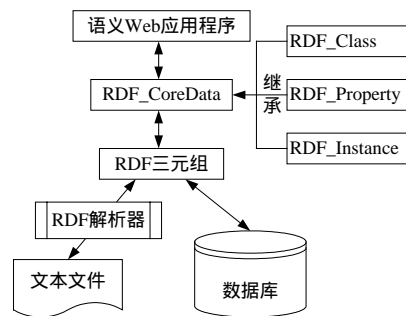


图 3 数据格式的转换

对上层的语义 Web 应用程序而言, RDF_CoreData 是一个类(Class), 它有 3 个子类: RDF_Class 类, RDF_Property 类, RDF_Instance 类。

5 结束语

本文针对工程设计类知识, 运用面向对象方法对其建模, 从而对工程设计类各种异构知识进行统一描述; 在此基础上, 运用 RDF(S) 技术, 结合面向对象的思想, 建立了语义 Web 环境下工程设计类知识的 Ontology; 并且提出了基于语义 Web 的知识存储模型, 以及面向上层应用程序的接口设计。这些研究为后续基于语义 Web 的应用系统开发奠定了基础。

参考文献

- 1 蒋祖华, 苏海. 工程设计类知识管理技术研究[J]. 计算机集成制造系统—CIMS, 2004, 10(10).
- 2 Lassila O, Swick R R. Resource Description Framework(RDF) Model and Syntax Specification[EB/OL]. 1999-02-22. <http://www.w3.org/TR/1999/REC-rdf-syntax-1999-02-22>.
- 3 Brickley D, Guha R V. Resource Description Framework(RDF) Schema Specification 1.0[EB/OL]. 2000-03-27. <http://www.w3.org/TR/2000/CR-rdf-schema-2000-03-27>.
- 4 Agrawal R, Somani A, Xu Y. Storage and Querying of E-commerce Data[C]//Proc. of VLDB, Roma. 2001.

(上接第190页)

在这种链表无序的条件下每个链表的平均长度为 N/M 。因此每一次的平均传输量为

$$40 \left(\frac{N}{M} \right) + l_{sig}$$

对于一个大规模应用的 AA 中心, 假设有 100 万份提前撤销的属性证书, 分布到 10 万个链表中, 则每次查询返回内容为 $400 + l_{sig}$, l_{sig} 为 100B 左右, 整个返回内容约为 500B。

另外一个需要考虑的通信负荷是 AA 对发布机构的更新, 仍以上面所述数据为例。假设证书的平均有效期为 1 年, 证书撤销目录每天更新一次, 则每次新撤销和删除的证书平均为 $1000000/365=2740$, 每次更新需要传输 2740 个证书序列号以及最终的签名。每个序列号 20B, 签名 100B, 总的通信量最多为 328800B, 负担较轻。可以看出, 这种方案适用于高频度的证书撤销更新。因为更新时的负担较轻, 同时如果增加更新次数, 每次相应的更新量会随之减少, 所以完全可以进行每小时一次更新甚至更高频率, 可以满足对证书撤销时效性比较敏感的需求。

4 协议的进一步改进

目前每个链表中的结点是无序排列的, 即假定查找的每个结点是等概率出现的。而在实际应用过程中, 不同的属性

证书的使用次数会有很大的差异, 可根据属性证书的使用频率对链表中的结点进行排序, 将查询频率最高的属性证书作为链表头, 之后按频率由大到小逐一排列。用户可根据应用系统的实际情况建立相应的优先等级, 这样会节约查找时间。同时在考虑撤销列表的分布式存储时可根据链表的访问频率高低进行地址的划分, 尽量做到负载均衡。

参考文献

- 1 IETF. X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile[S]. RFC3281. 2002.
- 2 Micali S. Efficient Certificate Revocation[R]. MIT Laboratory for Computer Science, Technical Report: TM-542b, 1996.
- 3 Kocher P. On Certificate Revocation and Validation[J]. Financial Cryptography, 1998, 14(65): 172-177.
- 4 范磊, 许崇祥. 基于二叉树的证书撤销管理[J]. 计算机工程, 2002, 28(6): 33-34.
- 5 王永静, 谢冬青, 陈华勇. 证书撤销机制的分析与设计[J]. 小型微型计算机系统, 2004, 21(9): 147-149.
- 6 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 1997: 251-262.