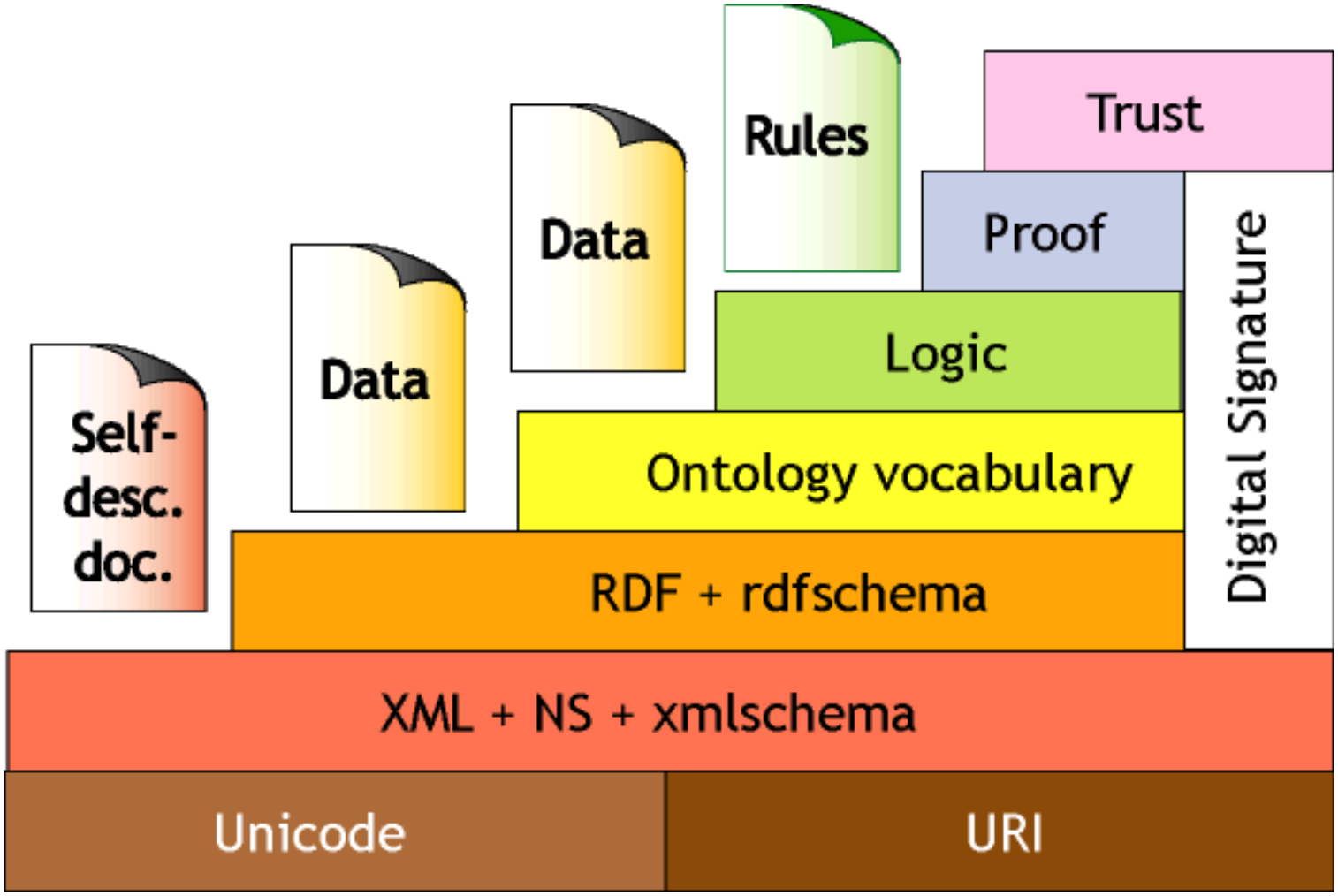
第二讲:语义网核心技术介绍

杨现民 博士
江苏师范大学

教育技术前沿课程之语义网研究专题
江苏师范大学，2012/10/25

Semantic Web

技术框架



Standar

Standar

Standa

七层模型

- 第1层：URI和Unicode，语义网的基础设施
- 第2层： XML、 Namespaces、 XML Schema和XML Query，语义网的语法层
- 第3层： RDF Model、RDF Schema和RDF Syntax，语义网的数据互操作层
- 第4层：Ontology，语义网的知识集合
- 第5层： Logic，规则及其描述方法是自动推理的基础
- 第6层：Proof，推理结果应该是可以验证的
- 第7层：Trust，语义网应该是一个可以信任的网络

URI

- 通用资源标识符（Uniform Resource Identifier)
- http://baike.baidu.com/cms/s/helloworld/img/hema-160x81.jpg

URI


姓 名 韦小宝
性 别 男 民 族 汉
出 生 1654 年 12 月 20 日
住 址 北京市东城区景山前街 4 号
紫禁城敬事房
公民身份证号码 11204416541220243X
UPLOADED TO WWW.U148.NET

XML

- Extensible Markup Language
- 可扩展标记语言：用于标记电子文件使其具有结构性的标记语言，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。

XML与HTML

```
<UL>
        <LI>
                <H1>e-Learning新解：网络教学范式的转换</H1>
                <H2>余胜泉、 程罡、董京峰</H2>
                <H3>北京师范大学现代教育技术研究所</H3>
        </LI>
        <LI>
                <H1>泛在学习环境下的学习资源设计</H1>
                <H2>余胜泉、 杨现民</H2>
                <H3>北京师范大学现代教育技术研究所</H3>
        </LI>
        ……
</UL>
```

XML与HTML

```
<Papers>
        <Paper>
          <title>e-Learning新解：网络教学范式的转换</title>
          <author>余胜泉、 程罡、董京峰</author>
          <affiliation >
                北京师范大学现代教育技术研究所
           </affiliation >
        </Paper>
        < Paper >
                <title>泛在学习环境下的学习资源设计</title>
                <author >余胜泉、 杨现民</ author >
                <affiliation >
                        北京师范大学现代教育技术研究所
                </affiliation>
        </ Paper >
        ……
</Papers>
```

XML与HTML

- XML与HTML的设计区别
  - XML的核心是数据，其重点是数据的内容。而HTML 被设计用来显示数据，其重点是数据的显示。
- XML和HTML语法区别
  - HTML的标记不是所有的都需要成对出现，XML则要求所有的标记必须成对出现；HTML标记不区分大小写，XML则大小敏感，即区分大小写。

XML的优势

- 轻量级的数据储存文件，使用的是非专有的格式，不受版权、专利、商业秘密或是其他种类的知识产权的限制。
- XML的功能是非常强大的，同时对于人类或是计算机程序来说，都容易阅读和编写。因而成为交换语言的首选。

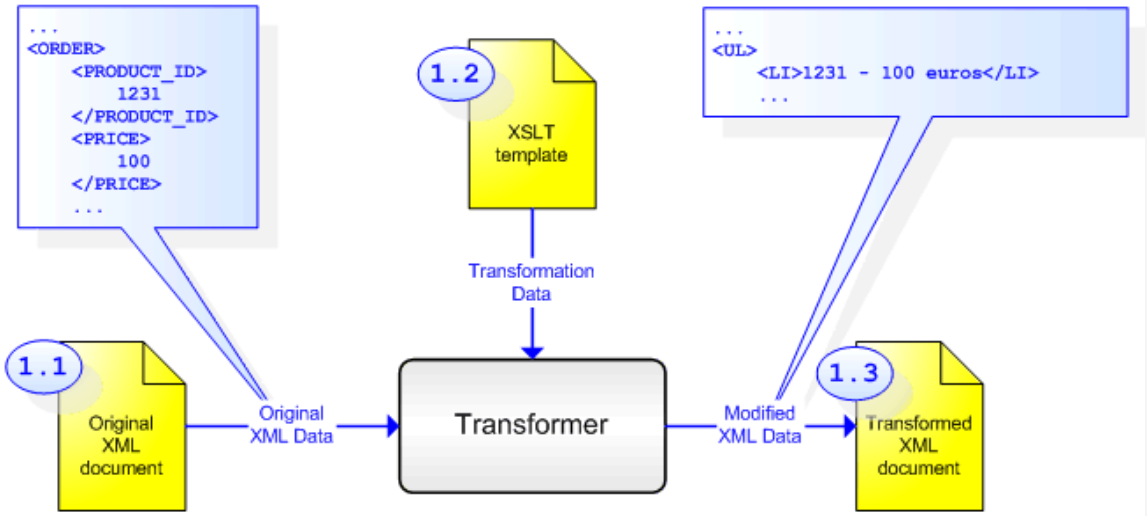
XML的优势

- 良好的格式
- 具有验证机制
- XML文档的内容和结构完全分离
- 互操作性强、规范统一
- 可扩展
  - MathML（数学标记语言）、CML（化学标记语言）和TecML（技术数据标记语言）

XML相关知识

- XSLT
  - 是一种用于将 XML 文档转换为 XHTML 文档或其他 XML 文档的语言
- XPath
  - XPath 是一种用于在 XML 文档中进行导航的语言
- DTD、XML Schema
  - Document Type Definition
  - 对xml文档结构进行规定，element、attribute、datatype
- 更多信息请阅读
  - http://www.w3school.com.cn/x.asp

XSLT




```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
 <html>
 <body>
   <h2>My CD Collection</h2>
   <table border="1">
     <tr bgcolor="#9acd32">
       <th>Title</th>
       <th>Artist</th>
     </tr>
     <tr>
       <td>.</td>
       <td>.</td>
     </tr>
   </table>
 </body>
 </html>
</xsl:template>

</xsl:stylesheet>
```

XPath

- XPath 是一门在 XML 文档中查找信息的语言。XPath 可用来在 XML 文档中对元素和属性进行遍历。

路径表达式	结果
bookstore	选取 bookstore 元素的所有子节点。
/bookstore	选取根元素 bookstore。 注释：假如路径起始于正斜杠(/)，则此路径始终代表到某元素的绝对路径！
bookstore/book	选取属于 bookstore 的子元素的所有 book 元素。
//book	选取所有 book 子元素，而不管它们在文档中的位置。
bookstore//book	选择属于 bookstore 元素的后代的所有 book 元素，而不管它们位于 bookstore 之下的什么位置。
//@lang	选取名为 lang 的所有属性。

DTD

- 文档类型定义（DTD）可定义合法的XML文档构建模块，它使用一系列合法的元素来定义文档的结构。
- 通过 DTD，您的每一个 XML 文件均可携带一个有关其自身格式的描述，一致地使用某个标准的 DTD 来交换数据。

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to      (#PCDATA)>
  <!ELEMENT from    (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body    (#PCDATA)>
]>
<note>
  <to>George</to>
  <from>John</from>
  <heading>Reminder</heading>
  <body>Don't forget the meeting!</body>
</note>
```

以上 DTD 解释如下：

!DOCTYPE note (第二行)定义此文档是 note 类型的文档。

!ELEMENT note (第三行)定义 note 元素有四个元素："to、from、heading,、body"

!ELEMENT to (第四行)定义 to 元素为 "#PCDATA" 类型

!ELEMENT from (第五行)定义 frome 元素为 "#PCDATA" 类型

!ELEMENT heading (第六行)定义 heading 元素为 "#PCDATA" 类型

!ELEMENT body (第七行)定义 body 元素为 "#PCDATA" 类型

XML Schema

- DTD的替代者，描述 XML 文档的结构
- 可针对未来的需求进行扩展
- 更完善，功能更强大
- 基于 XML 编写
- 支持数据类型
- 支持命名空间

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3school.com.cn"
xmlns="http://www.w3school.com.cn"
elementFormDefault="qualified">

<xs:element name="note">
    <xs:complexType>
      <xs:sequence>
         <xs:element name="to" type="xs:string"/>
         <xs:element name="from" type="xs:string"/>
         <xs:element name="heading" type="xs:string"/>
         <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:schema>
```

XML的不足

- It’s flexible, extensible, but just for syntactic, not for semantic.

语义描述能力不足

RDF

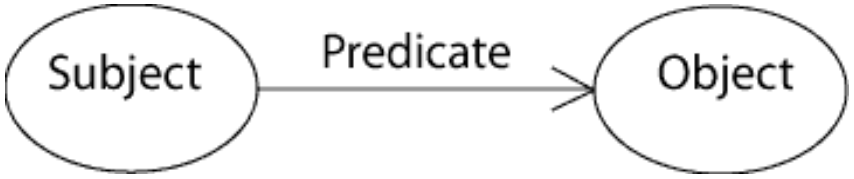
- Resource Description Framework
- RDF 指资源描述框架（Resource Description Framework）
- RDF 是一个用于描述 Web 上的资源的框架
- RDF 被设计为提供一种描述信息的通用方法，这样就可以被计算机应用程序读取并理解
- RDF 被设计为可被计算机阅读和理解
- RDF 描述不是被设计用来在网络上显示的
- RDF 使用 XML 编写
- RDF 是 W3C 语义网络活动的组成部分
- RDF 是一个 W3C 推荐标准

Standard

RDF

- RDF 使用 Web 标识符来标识事物，并通过属性和属性值来描述资源
  - 资源是可拥有 URI 的任何事物，比如 "http://www.w3school.com.cn/rdf"
  - 属性是拥有名称的资源，比如 "author" 或 "homepage"
  - 属性值是某个属性的值，比如 "David" 或 "http://www.w3school.com.cn"（请注意一个属性值可以是另外一个资源）

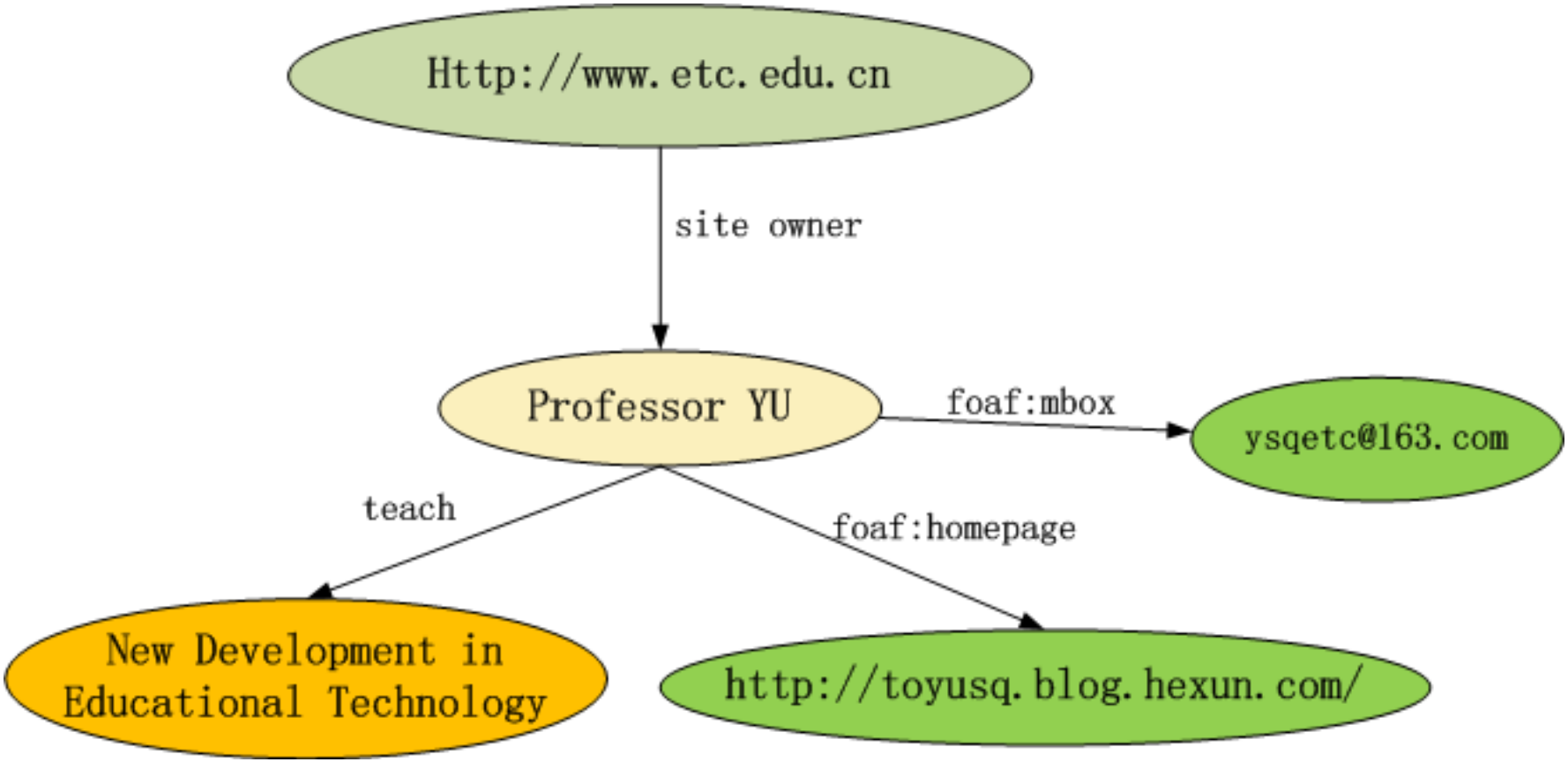
- 资源、属性和属性值的组合可形成一个陈述（被称为陈述的主体、谓语和客体）



RDF

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
 rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>

</rdf:RDF>
```

RDF


Http://www.etc.edu.cn
site owner
Professor YU
foaf:mbox
ysqetc@163.com
teach
foaf:homepage
New Development in Educational Technology
http://toyusq.blog.hexun.com/

RDF&RDFs

```
<rdf:Description rdf:about="http://www.etc.edu.cn">
    <site owner rdf:resource= “#Professor YU ” />
</rdf:Description>

<rdf:Description rdf:id=“Professor YU”>
    <rdf:type rdf:resource= “http://www.etc.edu.cn/university/#Professor ” />
    <foaf:mbox>ysqetc@163.com</foaf:mbox>
    <foaf:homepage>http://toyusq.blog.hexun.com</foaf:homepage>
    <UC:teach rdf:resource=“#NDET”></UC:teach>
</rdf:Description>

<rdf:Description rdf:i=d“NDET”>
   <rdf:type rdf:resource= “http://www.etc.edu.cn/university/#Course ” />
   <UC:coursename>New Development in Educational
Technology</UC:coursename>
   <UC:coursemark>3</UC:coursemark>
</rdf:Description>
```

RDFs

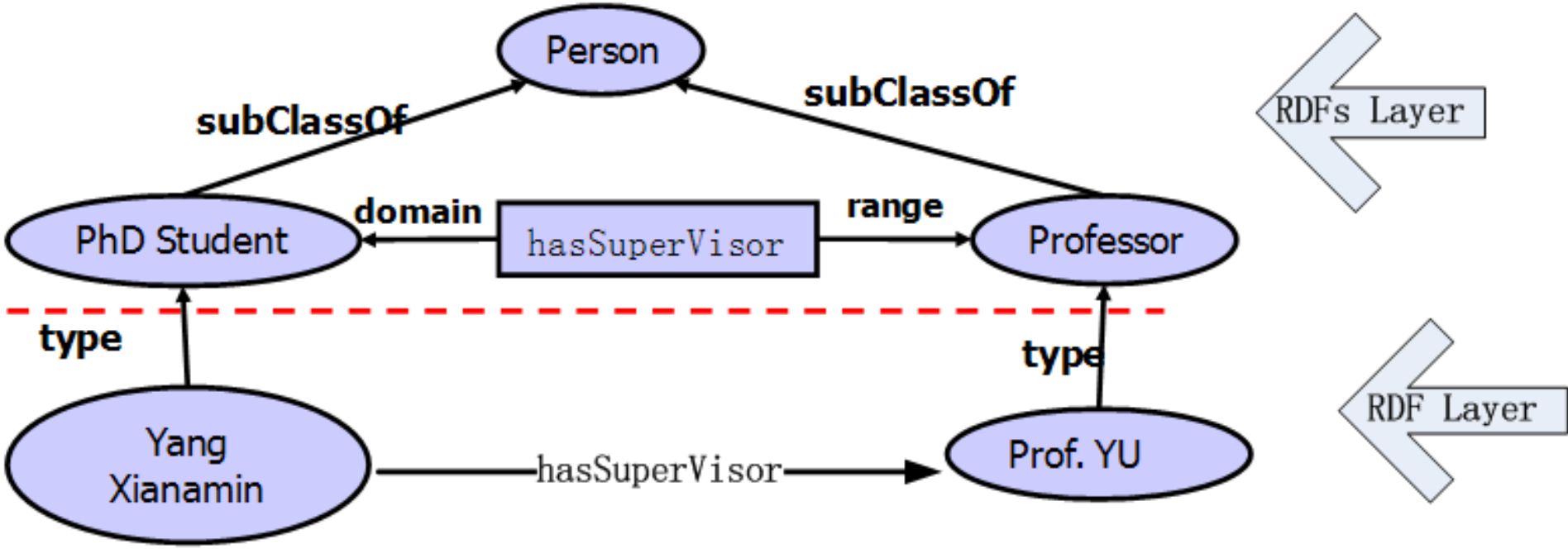
- RDFS即RDF Schema，用于定义元数据属性元素（例如“创建者”），以描述资源的一种定义语言。

- 其中RDFS可认为是一种本体语言，它讨论了类和属性(二元关系)、值域和定义域在属性上的约束以及子类和子属性的包孕关系。

RDFs

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base=  "http://www.animals.fake/animals#">

<rdfs:Class rdf:ID="animal" />

<rdfs:Class rdf:ID="horse">
  <rdfs:subClassOf rdf:resource="#animal"/>
</rdfs:Class>

</rdf:RDF>
```

RDFs


Person
subClassOf
subClassOf
RDFs Layer
PhD Student
domain
hasSuperVisor
range
Professor
type
type
RDF Layer
Yang
Xianamin
hasSuperVisor
Prof. YU

RDFs

- SPARQL 、 RQL 、 RDQL、TRIPLE、N3、Versa、SeRQL、……

- SPARQL  Standard
  - RDF查询语言 sparql 2007成为w3c的推荐标准

- 更多信息
  - http://www.w3school.com.cn/x.asp
  - http://www.w3.org/TR/rdf-sparql-query/

RDF&RDFS的不足

- 对于语义网来说，RDFS并不是一个合适的基础，它在充分详细地描述资源方面的表达能力太弱，更进一步来说，要将这些描述有效地应用于自动处理过程，还需要具有自动推理能力(如确定语法上不同的术语之间的语义关系)，而RDFS显然缺乏这样的能力。


本体语言
PROTO-INDO-EUROPEAN
INDIC
IRANIAN
NORTHEAST
CENTRAL
SOUTHWEST
TOCHARIAN
BALTO-SLAVIC
HELLENIC
ALBANIAN
ARMENIAN
GERMANIC
CELTIC
ITALIC
NORTH GERMANIC
WEST GERMANIC
EAST GERMANIC
INSULAR
CONTINENTAL
SABELLIC
LATINO-FALISCAN
Bihari
Sinhalese
Marathi
Guajarati
Hindi
Sindhi
Punjabi
Urdu
Romany
Bengali
Assamese
Classical Sanskrit
Middle Indic (Pali, Pankrits)
Vedic Sanskrit
Khotanese
Parthian
Bactrian
Sogdian
Avestan
Ossetic
Pashto
Baluchi
Kurdish
Old Persian
Middle Persian
Modern Persian
Tajik
Hittite
Palaic
Tocharian
Armenian
Medieval Greek
Greek
Serbian
Polish
Czech
Slovak
Slovene
Serbo-Croatian
Macedonian
Bulgarian
Old Church Slavonik
Ukranian
Belarussian
Russian
Old Prussian
Lithuanian
Latvian
Thracian
Illyrian
Messapic
Venetic
Gheg
Tosk
Icelandic
Faroese
Old Icelandic
Old Norse
Old Norwegian
Old Swedish
Old Danish
Middle Norwegian
Middle Swedish
Middle Danish
Norwegian
Swedish
Danish
Old English
Old Frisian
Middle English
English
Old Low Franconian
Old Saxon
Old High German
Middle Dutch
Middle Low German
Middle High German
Gothic
Celtiberian
Gaulish
Brittonic
Goidelic
Old Irish
Middle Irish
Medieval Breton
Breton
Old Welsh
Medieval Cornish
Cornish
South Picene
Umbrian
Oscan
Faliscan
Latin
Catalan
Rhaeto-Romance (Rumantsch)
Romanian
Old Italian
Old French
Old Provençal
Old Portuguese
Old Spanish
Italian
French
Provençal
Portuguese
Spanish

本体概念

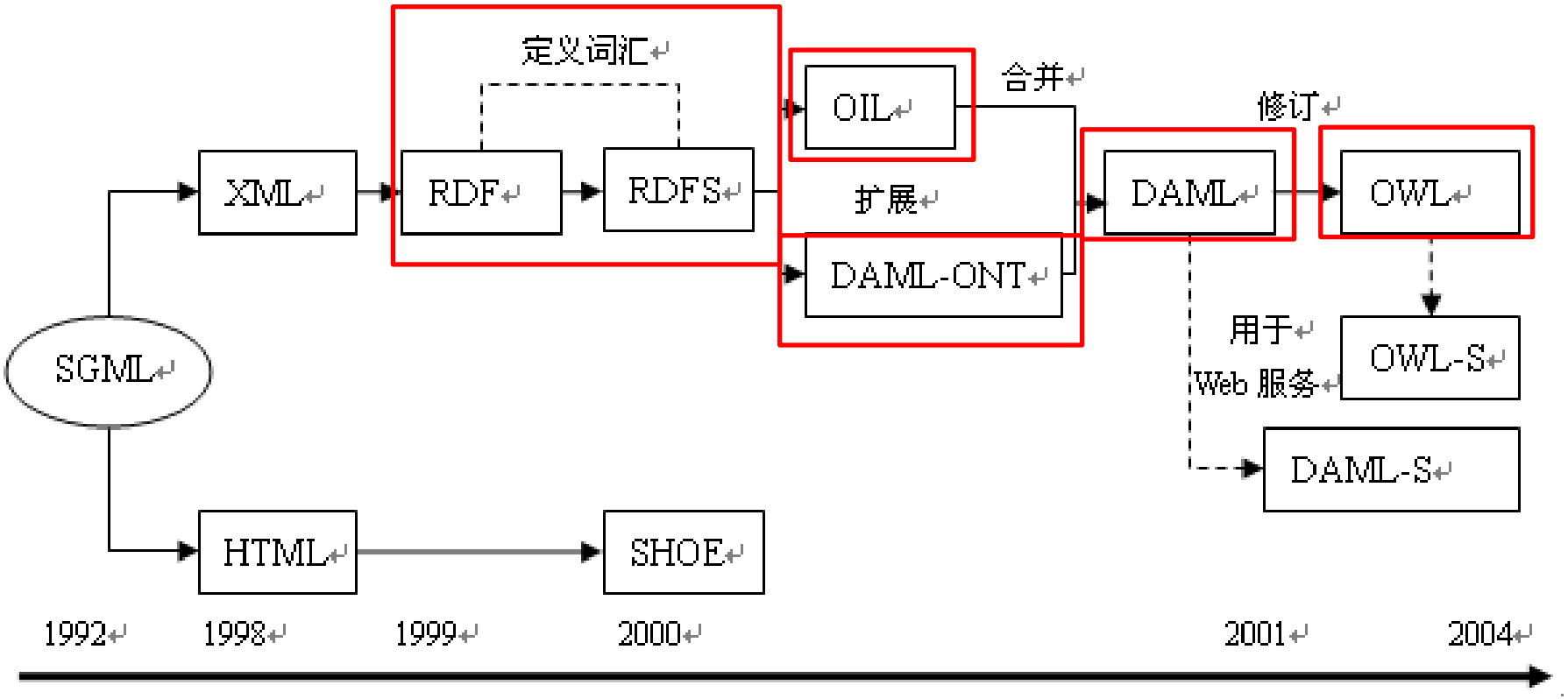
- 起源于哲学：存在的本质
- An ontology is an explicit specification of a conceptualization. (Gruber, 1993)
- 本体是对某领域包含的基本概念及其概念之间关系的一致的、规范化的说明
- 简单的说本体就是某领域常用概念和关系的词汇表

本体概念

- An ontology describes the concepts and relationships that are important in a particular domain.

- Composition
  - Key Concepts & Their Relationships
- Commonsense in one domain
- Semantic interoperability

本体描述语言


定义词汇
OIL
合并
修订
XML
RDF
RDFS
扩展
DAML
OWL
DAML-ONT
用于
Web 服务
OWL-S
SGML
DAML-S
HTML
SHOE
1992
1998
1999
2000
2001
2004

OWL语言

- OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

--From W3C OWL官网

OWL发展

- **OWL 1.0：published at 2004 -2-10**
  - **http://www.w3.org/TR/2004/REC-owl-features-20040210**
- **OWL2.0：published at 2009-11-12**
  - **http://www.w3.org/TR/owl2-overview**
  - OWL 2 has a very similar overall structure to OWL 1.

OWL2.0

- **OWL2.0 新特性**
  - keys;
  - property chains;
  - richer datatypes, data ranges;
  - qualified cardinality restrictions;
  - asymmetric, reflexive, and disjoint properties; and
  - enhanced annotation capabilities

--From W3C OWL官网

OWL子语言

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraints.
- *OWL DL* supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time).
- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees.

OWL 文档结构

- Part1： 命名空间

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <rdf:RDF xml:base="http://sw.opencyc.org/concept/"
3       xmlns="http://sw.opencyc.org/concept/"
4       xmlns:owl="http://www.w3.org/2002/07/owl#"
5       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
7       xmlns:skos="http://www.w3.org/2004/02/skos/core#"
8       xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
9       xmlns:cyc="http://sw.cyc.com/"
10      xmlns:opencyc="http://sw.opencyc.org/"
11      xmlns:cycAnnot="http://sw.cyc.com/CycAnnotations_v1#">
12
```

Xmlns=表明是缺省命名空间；
Xml：base=为当前文档指定了基准URI（base URI）

OWL 文档结构

- Part2： 本体元数据描述

```
13    <owl:Ontology rdf:about="">
14        <owl:versionInfo>Version 2.0.0</owl:versionInfo>
15        <rdfs:comment xml:lang="en">
16    OpenCyc Knowledge Base
17
18    Copyright© 2001-2009 Cycorp, Inc., http://www.cyc.com/, Austin, TX, USA
19
20    This file contains an OWL representation of information contained
21    in the OpenCyc Knowledge Base. The content of this OWL file is
22    licensed under the Creative Commons Attribution 3.0 license whose
23    text can be found at http://creativecommons.org/licenses/by/3.0/legalcode.
24    The content of this OWL file, including the OpenCyc content it represents,
25    constitutes the "Work" referred to in the Creative Commons license. The terms of
26    this license equally apply to, without limitation, renamings and other
27    logically equivalent reformulations of the content of this OWL file
28    (or portions thereof) in any natural or formal language, as well
29    as to derivations of this content or inclusion of it in other ontologies.
30
31    Mappings between OpenCyc terms and Wikipedia article names provided by
32    Olena Medelyan and Catherine Legg, University of Waikato, NZ under a Creative
33    Commons Attribution 3.0 license.
34
35        </rdfs:comment>
36    </owl:Ontology>
```

OWL 文档结构

- Part3：本体的主体

```
<owl:Class rdf:about="Mx8Ngh4rvYthVpwpEbGdrcN5Y29ycB8Ngh4rvVjqHpwpEbGdrcN5Y29ycB4rvpUnVpwpEbGdrcN5Y29ycA">
    <rdfs:label xml:lang="en">raquetball accessory</rdfs:label>
    <Mx4rwLSVCpwpEbGdrcN5Y29ycA xml:lang="en">accessory for playing racquetball</Mx4rwLSVCpwpEbGdrcN5Y29ycA>
    <Mx4rwLSVCpwpEbGdrcN5Y29ycA xml:lang="en">raquetball accessories</Mx4rwLSVCpwpEbGdrcN5Y29ycA>
    <cycAnnot:label xml:lang="en">(AccessoryFn-Activity (PlayingFn Racquetball-TheGame))</cycAnnot:label>
    <rdfs:comment xml:lang="en"><![CDATA[The collection of all raquetball accessories. A type of <a href="http:/
    <rdf:type rdf:resource="Mx4rejmDbrF_QdeIENT1qpbYzQ"/>
    <rdf:type rdf:resource="Mx4rpPHhAOB1EdqAAAACs6hRXg"/>
    <rdfs:subClassOf rdf:resource="Mx4rvVkH_ZwpEbGdrcN5Y29ycA"/>
    <owl:sameAs rdf:resource="http://umbel.org/umbel/sc/Raquetball_accessory"/>
    <owl:sameAs rdf:resource="http://sw.cyc.com/concept/Mx8Ngh4rvYthVpwpEbGdrcN5Y29ycB8Ngh4rvVjqHpwpEbGdrcN5Y29y
</owl:Class>

<owl:Class rdf:about="Mx8Ngh4rvYthVpwpEbGdrcN5Y29ycB8Ngh4rvVjqHpwpEbGdrcN5Y29ycB4rvrzM_pwpEbGdrcN5Y29ycA">
    <rdfs:label xml:lang="en">rugby accessory</rdfs:label>
    <Mx4rwLSVCpwpEbGdrcN5Y29ycA xml:lang="en">accessory for playing rugby</Mx4rwLSVCpwpEbGdrcN5Y29ycA>
    <Mx4rwLSVCpwpEbGdrcN5Y29ycA xml:lang="en">rugby accessories</Mx4rwLSVCpwpEbGdrcN5Y29ycA>
    <cycAnnot:label xml:lang="en">(AccessoryFn-Activity (PlayingFn Rugby))</cycAnnot:label>
    <rdfs:comment xml:lang="en"><![CDATA[The collection of all rugby accessories. A type of <a href="http://sw.c
    <rdf:type rdf:resource="Mx4rejmDbrF_QdeIENT1qpbYzQ"/>
    <rdf:type rdf:resource="Mx4rpPHhAOB1EdqAAAACs6hRXg"/>
    <rdfs:subClassOf rdf:resource="Mx4rvVkH_ZwpEbGdrcN5Y29ycA"/>
    <owl:sameAs rdf:resource="http://umbel.org/umbel/sc/Rugby_accessory"/>
    <owl:sameAs rdf:resource="http://sw.cyc.com/concept/Mx8Ngh4rvYthVpwpEbGdrcN5Y29ycB8Ngh4rvVjqHpwpEbGdrcN5Y29y
</owl:Class>
```

OWL 本体基本要素

- 要素1：类
  - <owl:Class rdf:ID=“VegetableDish">
  - <rdfs:subClassOf rdf:resource=“#Dish"/>
  - <rdfs:label xml:lang="en">vegetable dish</rdfs:label>
  - <rdfs:label xml:lang=“ch”>素菜</rdfs:label>
  - ...
  - </owl:Class>

OWL 本体基本要素

- 类的集合操作
  - owl:unionOf
  - owl:intersectionOf
  - owl:complementOf

```
<owl:Class rdf:ID="Student">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#BoyStudent" />
    <owl:Class rdf:about="#GirlStudent" />
  </owl:unionOf>
</owl:Class>
```

OWL 本体基本要素

- 几种特殊类
  - 等价类 **owl: equivalentClass**
  - 不相交类**owl: disjointWith**
  - 枚举类**owl: oneOf**

OWL 本体基本要素

- 等价类
  - <owl:Class rdf:ID=“Author">
  - <owl:equivalentClass rdf:resource="&lc;Creator"/>
  - </owl:Class>

OWL 本体基本要素

- 不相交类
  - <owl:Class rdf:ID=“Book">
  - <rdfs:subClassOf rdf:resource="#ReadingMaterial"/>
  - <**owl:disjointWith** rdf:resource="#Magazine"/>
  - <owl:disjointWith rdf:resource="#PPT"/>
  - <owl:disjointWith rdf:resource="#Newspaper"/>
  - </owl:Class>

OWL 本体基本要素

- 枚举类
  - <owl:Class rdf:ID=“SkinColor">
  - <rdfs:subClassOf rdf:resource="#Color"/>
  - <owl:oneOf rdf:parseType="Collection">
  - <owl:Thing rdf:about="#White"/>
  - <owl:Thing rdf:about="#Yellow"/>
  - <owl:Thing rdf:about="#Black"/>
  - </owl:oneOf>
  - </owl:Class>

OWL 本体基本要素

- 要素2：属性
  - datatype properties
    - birthDay (x)=2010-1-1，是xsd中具体的数据类型
  - object properties
    - selectCourse(x,y) x选修了y课程，y是对象

OWL 本体基本要素

- 数据类型属性
  - <owl:DatatypeProperty rdf:ID=“birthDay">
  - <rdfs:domain rdf:resource="#Person" />
  - <rdfs:range rdf:resource="&xsd;positiveInteger"/>
  - </owl:DatatypeProperty>

OWL 本体基本要素

- 对象属性
  - <owl:ObjectProperty rdf:ID=“selectCourse">
  - <rdfs:domain rdf:resource=“#Student" />
  - <rdfs:range rdf:resource="#Course" />
  - </owl:ObjectProperty>

OWL 本体基本要素

- 属性的特性
  - TransitiveProperty 【larger】
    - P(x,y) 与 P(y,z) 蕴含 P(x,z)
  - SymmetricProperty【friend】
    - P(x,y)当且仅当P(y,x)
  - FunctionalProperty【birthDate】
    - P(x,y) 与P(x,z) 蕴含 y = z
  - inverseOf【larger & smaller】
    - P1(x,y) 当且仅当P2(y,x)
  - InverseFunctionalProperty【hasID】
    - P(y,x) 与 P(z,x) 蕴含 y = z

OWL 本体基本要素

- 属性的限制
  - allValuesFrom ∀
  - someValuesFrom ∃
  - Cardinality(owl:minCardinality\owl:maxCardinality)
  - hasValue ∋

(2) 设 P(x)、W(x, y)、F(x, y)和 M(x, y)分别表示“x 是人”、“x 是 y 的外公”、“x 是 y 的父亲”和“x 是 y 的母亲”，则所给命题可符号化为：$(\forall x)(\forall y)(P(x) \wedge W(y,x) \rightarrow (\exists z)(M(z,x) \wedge F(y,z)))$.

OWL 本体基本要素

- 属性的限制
  - <owl:Class rdf:ID=“Student">
  - <rdfs:subClassOf rdf:resource=“#Person" />
  - <rdfs:subClassOf>
  - <owl:Restriction>
  - <owl:onProperty rdf:resource="#selectCourse" />
  - <owl:allValuesFrom rdf:resource="#Course" />
  - </owl:Restriction>
  - </rdfs:subClassOf>
  - ...
  - </owl:Class>

OWL 本体基本要素

- 要素3：个体(实例)
  - 类的成员
  - 引入一个体（individual），只需将它们声明为某个类的成员。
  - <Student rdf:ID=“Yangxianmin" />
  - Or
  - <owl:Thing rdf:ID=“Yangxianmin" />
  - <owl:Thing rdf:about="#Yangxianmin">
  - <rdf:type rdf:resource="#Student"/>
  - </owl:Thing>

OWL 本体基本要素

- 等价个体
  - 两个个体声明成一致
  - <Student rdf:ID=“Yangxianmin>
  - <owl:sameAs rdf:resource="#Max Yang" />
  - </Student>
- 不同个体
  - <Student rdf:ID=“Yangxianmin">
  - <owl:differentFrom rdf:resource=“#YXM"/>
  - <owl:differentFrom rdf:resource="#Yangxm"/>
  - </Student>

本体工具

Protégé

- Protégé is a free, open source ontology editor and knowledge-base framework.
- http://protege.stanford.edu/
- The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema.
- Currently being used in clinical medicine and the biomedical sciences

Protégé

Protégé is a free, open source ontology editor and knowledge-base framework.

The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema. (more)

Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. (more)

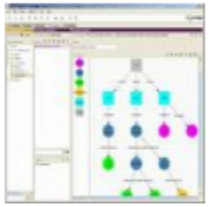
Protégé is supported by a strong community of developers and academic, government and corporate users, who are using Protégé for knowledge solutions in areas as diverse as biomedicine, intelligence gathering, and corporate modeling.

community	
Registered Users	200,045
protege-users list members	17,269
protege-discussion list members	2,501
protege-owl list members	2,312

Protégé is available from this site as a free download along with plug-ins and ontologies.

downloads	
Protégé 4.2 beta	Jul 12, 2012
Protégé 4.1	Jul 26, 2011
Protégé 3.5 beta	May 9, 2012
Protégé 3.4.8	Jan 12, 2012
WebProtégé 2.0 beta	June 4, 2012

go to protégé-owl



go to protégé-frames

go to WebProtégé

- plug-in architecture & Java-based Application Programming Interface (API)
- The latest version is 4.1 (October 26, 2010)
- Protégé 4.X is an OWL ontology development environment.
- Besides HermiT, there are other reasoners available for Protege 4.1, including Pellet and FaCT++. These reasoners are available for download from the File | Check for plugins... menu item.

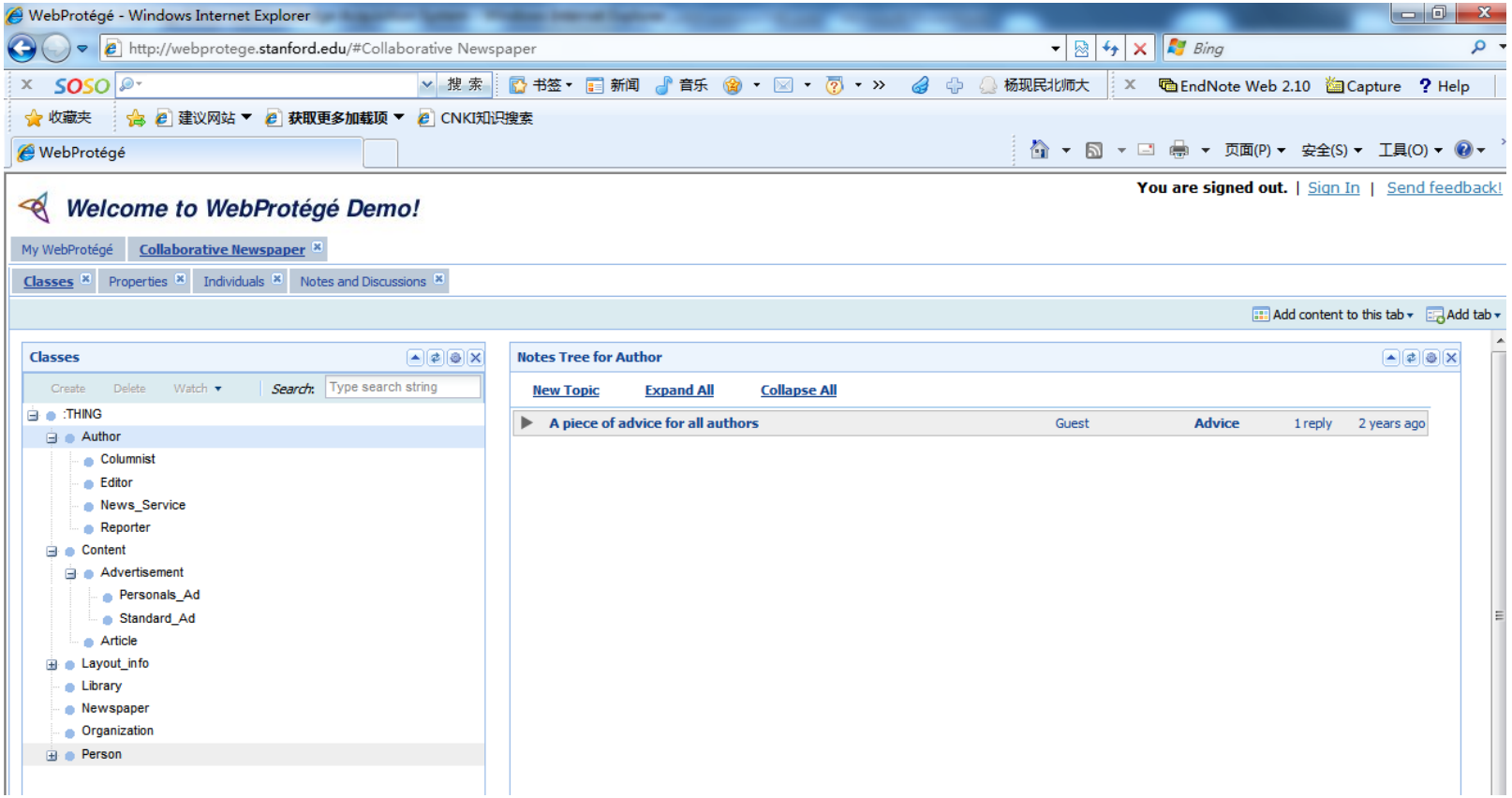
- System classes rooted in :THING
- 超类owl:Thing
- Deleting a class deletes the class and all of its subclasses
- Protégé提供本体概念类，关系，属性和实例的构建，并且屏蔽了具体的本体描述语言

Protégé-OWL editor

- The Protégé-OWL editor enables users to: Load and save OWL and RDF ontologies.
  - Edit and visualize classes, properties, and SWRL rules.
  - Define logical class characteristics as OWL expressions.
  - Execute reasoners such as description logic classifiers.
  - Edit OWL individuals for Semantic Web markup.

WebProtégé

- WebProtégé实现在线本体协同编辑
- http://webprotege.stanford.edu/#Collaborative Newspaper

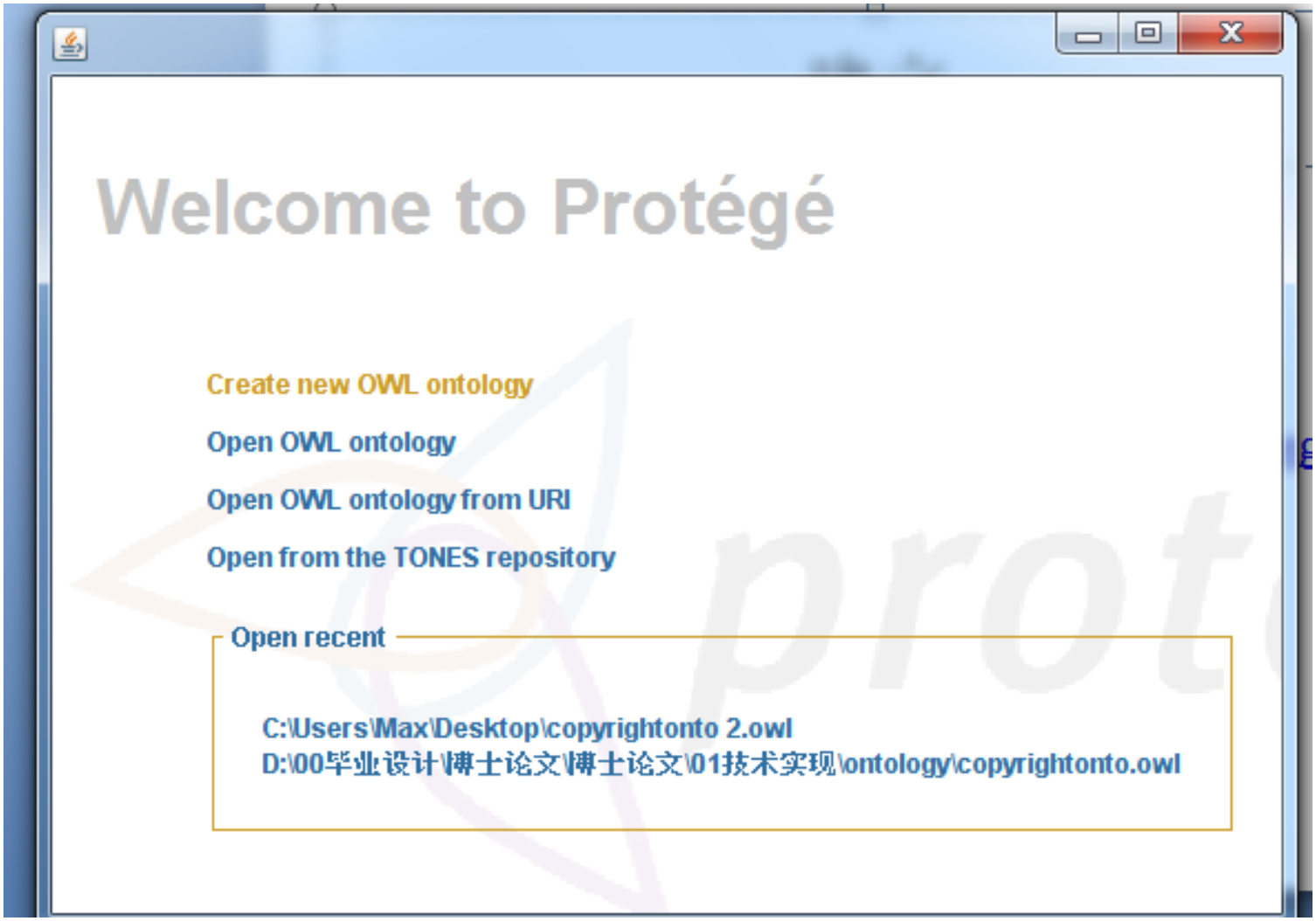


如何利用Protégé OWL Editor 创建本体?

安装Protégé

- 安装JDK,配置环境变量
- 安装protégé-OWL editor
  - http://protege.stanford.edu/download/protege/4.0/zip/?C=M;O=D
- 安装graphviz
  - http://www.pixelglow.com/graphviz/

创建OWL本体


Welcome to Protégé
Create new OWL ontology
Open OWL ontology
Open OWL ontology from URI
Open from the TONES repository
Open recent
C:\Users\Max\Desktop\copyrightonto 2.owl
D:\00毕业设计\博士论文\博士论文\01技术实现\ontology\copyrightonto.owl

创建OWL本体

- 实际操作
  - 类的增、删、改、查
  - 属性的增、删、改、查
  - 个体的增、删、改、查
  - OWLViz可视化查看类关系
  - DL query

JENA框架

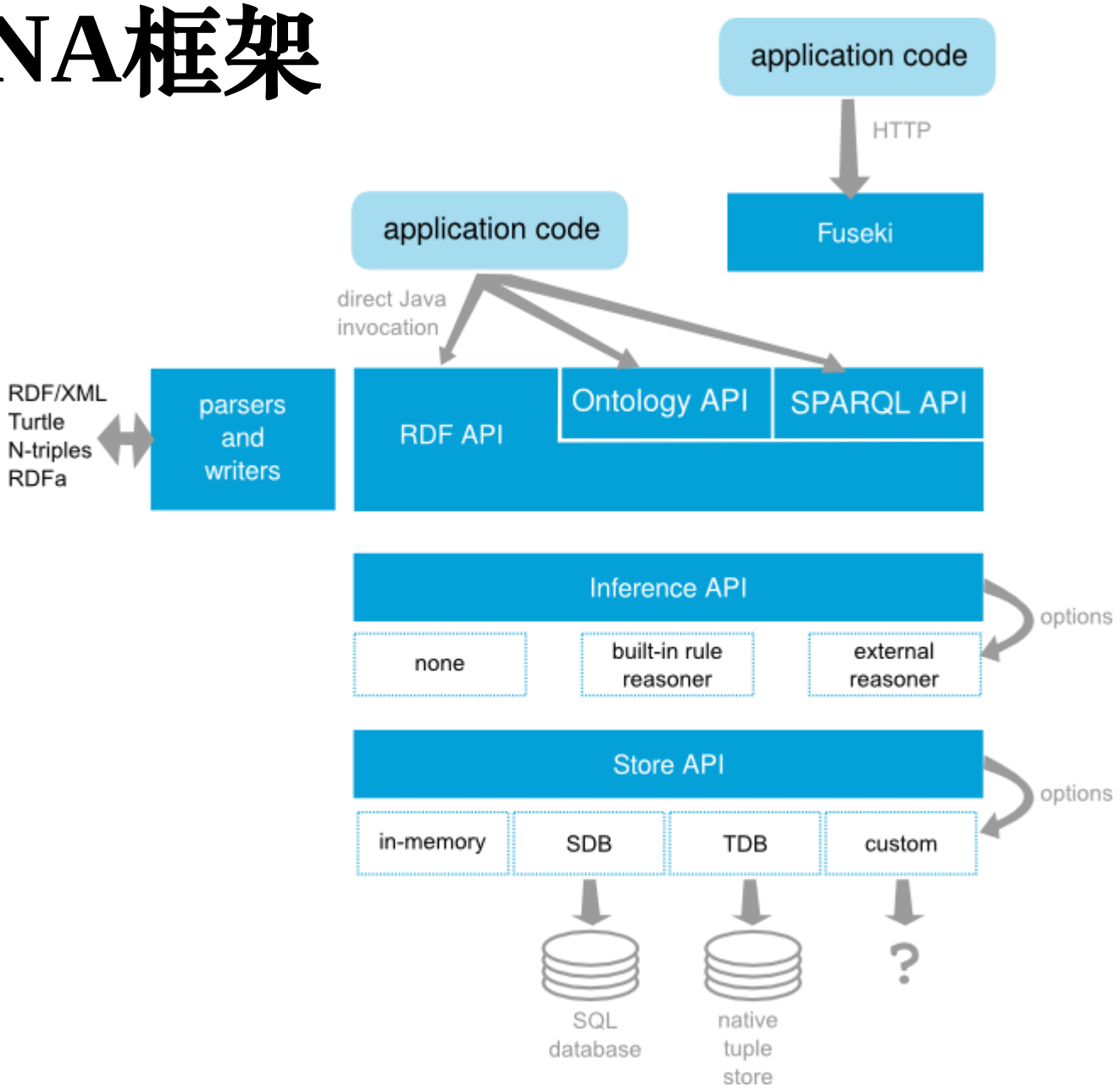
- 我们在Protege这样的编辑器里构建了本体，那么如何在应用程序里使用本体呢?

- 我们需要一些开发接口 ！！

- Jena由 HP Labs（http://www.hpl.hp.com）开发的Java开发工具包，用于Semantic Web(语义网)中的应用程序开发。

- Jena对应用程序就像Protégé对我们——我们使用Protégé操作本体，应用程序则是使用Jena来做同样的工作。

JENA框架


application code
HTTP
Fuseki
application code
direct Java
invocation
RDF/XML
Turtle
N-triples
RDFa
parsers
and
writers
RDF API
Ontology API
SPARQL API
Inference API
options
none
built-in rule
reasoner
external
reasoner
Store API
options
in-memory
SDB
TDB
custom
?
SQL
database
native
tuple
store

- The Jena Framework includes:
  - an API for reading, processing and writing RDF data in XML, N-triples and Turtle formats;
  - an ontology API for handling OWL and RDFS ontologies;
  - a rule-based inference engine for reasoning with RDF and OWL data sources;
  - stores to allow large numbers of RDF triples to be efficiently stored on disk;
  - a query engine compliant with the latest SPARQL specification
  - servers to allow RDF data to be published to other applications using a variety of protocols, including SPARQL

JENA功能

- 以RDF/XML、三元组形式读写RDF
- RDFS，OWL，DAML+OIL等本体的操作
- 利用数据库保存数据
- 查询本体模型
- 基于规则的推理

如何应用JENA

- 登录Jena的主页
  (http://jena.sourceforge.net/downloads.html)
- 下载Jena的最新版本
- 将下载的zip文件解压到任意路径
- 将lib下的jar文件全部加入CLASSPATH
- 如果使用Eclipse，则可以通过修改工程的Java创建路径的方法导入Jena jar文件

常用操作

- 创建本体模型
  - *OntModel ontModel = ModelFactory.createOntologyModel();*
- 从外部文件读取本体模型
  - *ontModel.read("file:D:/temp/Creatrue/Creature.owl")*
- 创建属性
  - *Property childOf = model.createProperty(relationshipUri,"childOf");*
- 创建类
  - *OntClass oc= ontModel .createClass(classURI);*
- 创建一个三元组
  - *Statement statement = model.createStatement(adam,parentOf,fran);*

常用操作

- **基于规则的推理**

  - //自定义推理规则
  - String rules="[Rule1: (?x "+travel+"hasPart ?y)(?x "+travel+"hasPart ?z)" +
  - "->(?y "+travel+"inSameDomain ?z)]" +
  - "[Rule2: (?x "+travel+"hasPart ?y)(?y "+travel+"hasPart ?z)" +
  - "->(?x "+travel+"hasPart ?z)]";
  -
  - Reasoner reasoner=new GenericRuleReasoner(Rule.*parseRules(rules));*

  - //把编写的规则加入到已有的推理规则中
  - reasoner=reasoner.bindSchema(model);
  - InfModel infmodel=ModelFactory.*createInfModel(reasoner,model);*

Any question, please contact me:

Xianmin Yang

Ph.D.

Jiangsu Normal University

TEL:15862183989

E-Mail: yangxianmin8888@163.com

Vita:http://lcell.bnu.edu.cn/TeamMember/Yang/index.html

MicroBlog:http://t.sina.com.cn/yangxianmin8888

School of Information and Communication, Jiangsu Normal University

NO 57, Heping Road, Jiangsu Normal University, Xuzhou 221009

* Actions speak louder than words *