

语义网 (Semantic Web)

吉建民

USTC
jianmin@ustc.edu.cn

2024 年 4 月 5 日

Used Materials

Disclaimer: 本课件采用了部分网络资源

Table of Contents

语义网概览

XML 结构化网络文档

用 RDF 描述网络资源

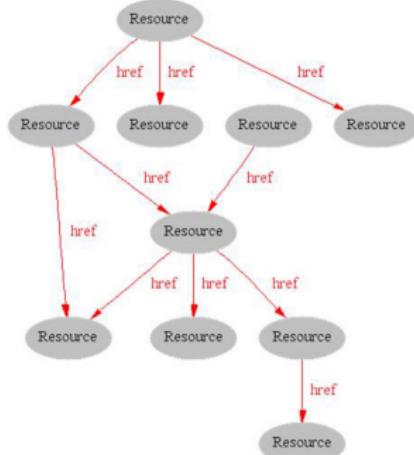
网络本体语言 OWL

语义网历史

- ▶ 语义网，最早由国际万维网联盟（World Wide Web Consortium, W3C）的 director Tim Berners-Lee (WWW 的发明者) 在 1998 (2000) 年提出。
- ▶ Tim Berners-Lee 的设想：
 - ▶ “... a goal of the Web was that, if the interaction between person and hypertext could be so intuitive that the **machine-readable** information space gave an accurate representation of the state of people's thoughts, interactions, and work patterns, then **machine analysis** could become a very powerful management tool, seeing patterns in our work and facilitating our working together through the typical problems which beset the management of large organizations.”
 - ▶ 通过给 WWW 上的文档添加能够被计算机所理解的语义 (Meta data)，将当前网络上无结构或半结构化的文档转换为“web of data”，从而使计算机与人更好的合作（机器可自动处理，机器可理解）。
- ▶ semantics + web + AI = more useful web

当今网络：“the Syntactic Web”

- ▶ 大多数 web 是设计给人浏览的
- ▶ 计算机负责演示 (presentation) ,
人负责联系 (linking) 和解释
(interpreting)
- ▶ 计算机不理解 web 的内容



“the Syntactic Web” 的不足

- ▶ 复杂查询涉及背景知识 (background knowledge)
 - ▶ 查找信息：“animals that use sonar but are not either bats or dolphins”, e.g. Barn Owl.
- ▶ 在数据存储库中定位信息
 - ▶ 旅游查询
 - ▶ 商品和服务价格
 - ▶ 人类基因组实验结果
- ▶ 寻找和使用 “网络服务” (web services)
 - ▶ 将两个蛋白质之间的表面相互作用可视化
- ▶ 委托复杂任务给 web “agents”
 - ▶ 将我的下个周末安排在温暖的、不太远、说英语或法语的地方

问题在哪里？

The screenshot shows the homepage of the WWW 2002 conference. At the top left is the URL "http://www2002.org". The main title "WWW 2002" is in large yellow letters. Below it, the subtitle "THE ELEVENTH INTERNATIONAL WORLD WIDE WEB CONFERENCE" is displayed. To the right is the logo for "CONFERENCE ORGANIZER International World Wide Web Conference Committee". On the left, there's a logo for "W3C 2002 HAWAII". The center features a yellow banner with the text "1 LOCATION. 5 DAYS. LEARN. INTERACT.". Below the banner, a sidebar lists various conference sections: Conference Proceedings, Call for Participation, Program, Registration Information, Hotel Accommodation, Conference Committee, Sponsorship/Exhibition Opportunities, Volunteer Information, Information about Hawaii, and Previous & Future WWW Conferences. The main content area includes a list of registered participants from countries like Australia, Canada, Chile, Denmark, France, Germany, Ghana, Hong Kong, India, Ireland, Japan, Malta, New Zealand, The Netherlands, Norway, Singapore, Switzerland, The United States, Vietnam, and Zambia. A "REGISTER NOW" button is present. A section titled "FEATURED SPEAKERS (CONFIRMED)" lists Tim Berners-Lee and Richard A. DeMillo, along with logos for various sponsors.

- ▶ 页面中标记包括：
 - ▶ 渲染信息：如字体大小、颜色
 - ▶ 到相关内容的超链接
- ▶ 语义内容可以被人获得，但无法很容易的由计算机获得

我们所看到的内容

WWW2002

The eleventh international world wide web conference

Sheraton waikiki hotel

Honolulu, hawaii, USA

7-11 may 2002

1 location 5 days learn interact

Registered participants coming from

australia, canada, chile denmark, france, germany, ghana, hong kong, india, ireland, italy, japan, malta, new zealand, the netherlands, norway, singapore, switzerland, the united kingdom, the united states, vietnam, zaire

Register now

On the 7th May Honolulu will provide the backdrop of the eleventh international world wide web conference. This prestigious event ...

Speakers confirmed

Tim bernes-lee

Tim is the well known inventor of the Web, ...

Ian Foster

Ian is the pioneer of the Grid, the next generation internet

需要加入“语义”

- ▶ 通过额外的定义，统一注解 (annotations) 的含义
 - ▶ 对所有的注解标签 (annotation tags) 给出统一的规范定义
 - ▶ 这种方式的问题：不灵活，只能规范有限的内容
- ▶ 通过本体 (ontology) 来刻画注解的含义
 - ▶ 本体提供了术语的词汇表
 - ▶ 可以通过老的术语来定义新的术语
 - ▶ 这些术语的含义（语义，semantics）是形式化定义的
 - ▶ 可以在不同的本体间定义术语之间的关系

本体的概念

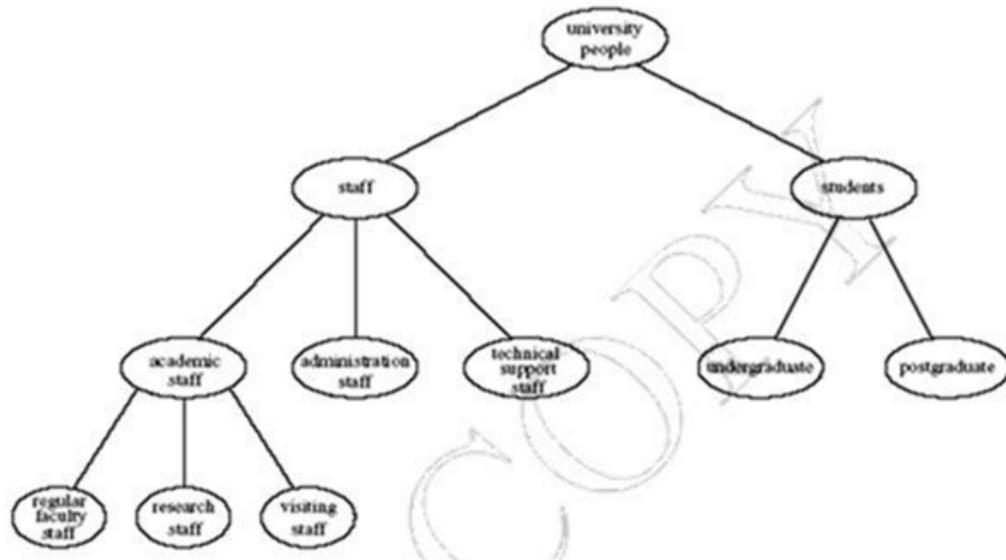
- ▶ 最早是一个哲学概念，从哲学的范畴来说，Ontology 是客观存在的一个系统的解释或说明，关心的是客观现实的抽象本质（Science of Being）
 - ▶ What characterizes being?
 - ▶ Eventually, what is being?
- ▶ 在人工智能界，Ontology：“给出构成相关领域词汇的基本术语和关系，以及利用这些术语和关系构成的规定这些词汇外延的规则的定义”
- ▶ 通常认为，Ontology 是共享概念模型的明确的形式化规范说明

本体的结构

本体刻画一个特定领域形式化描述，通常包含两部分：

- ▶ 领域中重要概念和关系的名字：
 - ▶ Elephant 是成员为一种特定动物的概念
 - ▶ Herbivore（食草动物）是成员为只吃植物或植物的部分的动物的概念
 - ▶ Adult_Elephant 是成员为年龄大于 20 的 Elephant 的概念
- ▶ 领域的背景知识或约束：
 - ▶ Adult_Elephant 至少重 2,000 kg
 - ▶ 所有的 Elephant 要么为 African_Elephant 要么为 Indian_Elephant
 - ▶ 不存在同时为 Herbivore 和 Carnivore（食肉动物）的个体

类别体系结构 (Class Hierarchy) 例子



语义网：第一步

语义网的目的：使网络资源更适合自动处理

- ▶ 用语义标记（semantic markup）扩展现有的渲染标记
 - ▶ 用元数据注解（metadata annotation）描述网上资源的内容或功能
- ▶ 用本体提供这些注解的词汇表（vocabulary）
 - ▶ 机器可处理的“形式化描述”
- ▶ 前提是需要一个标准的网络本体语言（web ontology language）
 - ▶ 遵循统一的语法（syntax）
 - ▶ 类似的，syntactic web 也基于特定的标准，如 HTTP 和 HTML

本体库的设计与发展

- ▶ 本体在语义网中起关键作用，用来提供工具（tools）和服务（services）：
 - ▶ 设计和维护高质量的本体库：
 - ▶ Meaningful：所有类型都存在实例
 - ▶ Correct：符合领域专家的直觉
 - ▶ Minimally redundant：不存在故意的同义词
 - ▶ Richly axiomatised：尽可能详细的描述
 - ▶ 存储本体类别中大量实例（instances），例如：网页中的注解（annotation）
 - ▶ 回答关于本体类别与实例的问题（queries），例如：
 - ▶ 找到更一般的类别
 - ▶ 检索一个符合给定描述的注解/页面
 - ▶ 整合（integrate）和调整（align）多个本体库

本体库例子

File Log Reasoner Help Export

Classes Properties Individuals Axioms Container Namespaces Imports

SubclassOf
 SameClassAs

Name: african_elephant

Documentation: "Elephants from Africa."

Classes: elephant

Restrictions:

type	property	filler
has-class	comes_from	(one-of Africa)

Find:

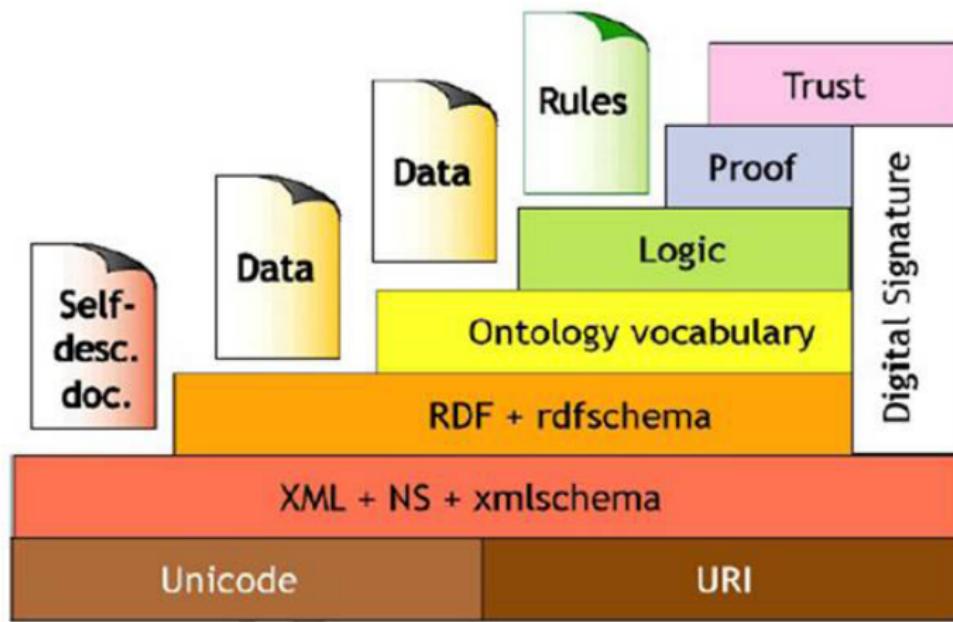
/home/horrocks/excalibur/systems/OilEd/Ontologies/elephants

Navigation icons: back, forward, search, etc.

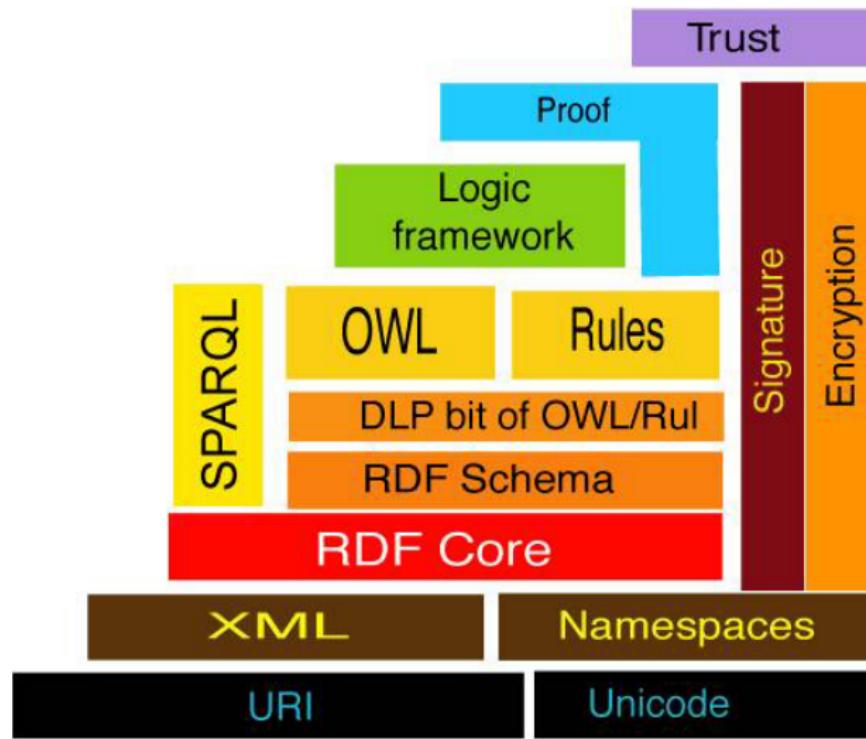
语义网层次结构

通过层次结构发展语义网，在另一层基础上定义新的一层。

- ▶ 向下计算容易
- ▶ 向上表达能力强



另一种语义网栈 (semantic web stack)



小结

- ▶ “Syntactic Web” 机器只负责 presentation，不理解页面的内容
- ▶ semantics + web + AI = more useful web
- ▶ “Semantic Web” 让机器更多的了解页面内容，更多的自动处理网络资源
- ▶ 用 annotation 来给页面内容加标签，再利用 ontology 形式化定义 annotation 的含义
- ▶ Ontology 刻画领域的公共知识：基本概念和关系，以及基于它们的规则
- ▶ 语义网通过层次结构发展，向下计算容易，向上表达能力越强

Table of Contents

语义网概览

XML 结构化网络文档

用 RDF 描述网络资源

网络本体语言 OWL

从 HTML 说起

- ▶ Berners-Lee 开发 HTML 的初衷是使用超文本作为组织分布式文档系统的一种方式，HTML 的标签主要是面向显示的，但其一直在努力增加一些标签来提供语义：
- ▶ HTML2.0 引入了 META 元素和 REL 属性。META 元素以名称、值的形式规定了元数据。META 一个流行的用法是表示关键字，比如 <META name = “kewyords” content = “Semantic Web”>，这样会帮助搜索引擎标引这个页面。
- ▶ HTML3.0 增加了 Class 属性，可以被任何标签使用来建立该元素的子类，不过这个语义标记很少被使用，不过即使被使用了，他们所提供的语义也是很有限的。
- ▶ 为了解决 HTML 的语义局限性，Dobson and Burrill 试着将其同 ER 关系模型结合。这就是超级 HTML，它由一系列简单的标签定义了文档中的实体，文档体的标记部分作为这些实体的属性，然后定义从实体的外部实体的关系。这是正式为 Web 页面增加结构数据的首次尝试，从而为解决这个问题提供了一种方法，也是之后 XML 设计的动机。

XML 概述

- ▶ XML (eXtensible Markup Language) 让每个人都能创建自己的标签，例如 < 姓名 >，从而支持应用程序将这些标签运用到复杂的应用中。也即 XML 允许用户在文档中加入了任意的结构
- ▶ 由于结构任意，XML 交换的双方需要在使用上一致，这样的一致性描述（结构信息）就是 DTD (Document Type Definition) 或 XML Schema
 - ▶ DTD 较老，限制较多
 - ▶ XML Schema 通过定义数据类型，提供了可扩充性
- ▶ 然而，XML 并不提供标签的意义。标签 <p> 可能意味着分段 (paragraph)，也可能意味着一部分 (part)。这需要通讯双方事先达成理解的一致

XML 特点和运用

- ▶ 可扩展性是 XML 最主要的特点
- ▶ 另外 XML 是一种元标记语言（它只表示数据本身，没有显示的样式）
 - ▶ 和 HTML 比较：HTML 不仅表示数据而且还有显示的样式
- ▶ 真正的数据和显示分离
- ▶ 基于这样的特点，XML 主要用来交换数据
 - ▶ 用交换双方都可以明白的标签来封装数据
 - ▶ 比如 Web Services，就是在不同平台和不同语言之间交换数据
- ▶ 写配置文件

XML 文档构成

一个 XML 文档就是扩展名为 .xml 的文本文件

XML 声明 { <?xml version="1.0" encoding="utf-8"?>

文档类型声明 { <!DOCTYPE books SYSTEM "books.dtd" [
 <!ENTITY bookName "XML简明教程" >
]>

文档元素 { <books>
 <book>
 <name>Java超级编程</name>
 <author>比尔盖茨</author>
 </book>
</books>

结构信息

- ▶ 结构信息
 - ▶ 定义所有的元素和属性名
 - ▶ 元素: <lecture> </lecture>
 - ▶ 属性: <lecture name = "AI" room = "3B215" />
 - ▶ 定义结构
 - ▶ 属性可以取哪些值
 - ▶ 哪些元素可能或必须出现在哪些元素中
- ▶ 利用结构信息可以检验 XML 文档是否满足要求

DTD 概述

- ▶ 若 XML 文件是 well-formed, 则不需要 DTD
- ▶ DTD: XML 文件的文件型别定义 (Document Type Definition) 可以看成一个或者多个 XML 文件的模板, 在这里可以定义 XML 文件中的元素、元素的属性、元素的排列方式、元素包含的内容等等。
- ▶ DTD 的作用:
 - ▶ 每个 XML 文件可以携带一个自身格式的描述
 - ▶ 不同组织的人可以使用一个通用 DTD 来交换数据
 - ▶ 应用程序可以使用一个标准 DTD 校验从外部世界接受来的 XML 数据是否有效
 - ▶ 可以使用 DTD 校验自己的 XML 数据
- ▶ 尽管 DTD 为 XML 文档提供了一个语法规范, DTD 并不提供语义信息。也就是说, DTD 中的一个元素的意义, 或者是由人根据在 DTD 中的自然语言描述的名称和注释来理解, 或者在 DTD 之外再编写一个文档中来专门描述意义
 - ▶ 这样, XML 文档的交换就必须要求交换的实体事先在 DTD 的使用和意义理解上都达成一致
 - ▶ 如果只是固定的合作实体之间, 可能不会有太大问题, 但是如果是在 Web 上……

- ▶ 一个 DTD 的组成部分可以定义在一个独立的文件（外部 DTD）中，也可以定义在该 XML 文档自身（内部 DTD）中
- ▶ DTD 由四部分构成
 - ▶ 元素 (Elements): `<!ELEMENT 元素名称 元素内容>`。按元素内容分有如下 4 类：
 - ▶ 空元素类型: `<!ELEMENT 元素名称 EMPTY>`
 - ▶ ANY 元素类型: `<!ELEMENT 元素名称 ANY>`
 - ▶ 父元素类型: 元素内容可以包含子元素，通过正则表达式规定子元素出现的顺序和次数
 - ▶ 混合元素类型: 元素内容可以包含文本，同时文本之间可以选择的插入子元素，但子元素的顺序和次数不受限制
`<!ELEMENT 元素名称 (#PCDATA | 子元素名 1 | 子元素名 2 | ...)>`
 - ▶ 属性 (Attribute): `<!ATTLIST 元素名称 属性名称 属性值型态 属性的内定值>`
 - ▶ 实体 (Entities): `<!ENTITY 实体名称 实体内容>`
 - ▶ 注释 (Comments): `<!-- 注释内容 -->`

XML Schema 概述

- ▶ DTD 仅仅提供了一个简单的结构描述：他们定义了元素出现的结构，位置，可能的属性等等
- ▶ XML Schema 被设计来代替 DTD。XML Schema 有几个优于 DTD 之处：
 - ▶ XML Schema 提供了一个丰富的语法来描述元素的结构，比如你可以定义元素出现的次数，默认值
 - ▶ XML Schema 提供支持数据类型。比如你可以定义电话号码是一个五位数字
 - ▶ XML Schema 提供了包含和继承机制，使你可以重用共同的元素定义，也可以将存在的定义运用于新的应用
 - ▶ XML Schema 以 XML 作为其编码的语法（因为 XML 是一个元语言），使得工具的开发变简单了，因为文档和文档定义都使用了相同的语法
- ▶ 通过数据类型 (data types) 定义元素和属性类型
 - ▶ 数据类型包括：built-in data types 和 user-defined data types

XML Schema 例子

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Address">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Recipient" type="xs:string" />
        <xs:element name="House" type="xs:string" />
        <xs:element name="Street" type="xs:string" />
        <xs:element name="Town" type="xs:string" />
        <xs:element name="County" type="xs:string" minOccurs="0" />
        <xs:element name="PostCode" type="xs:string" />
        <xs:element name="Country" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="IN" />
              <xs:enumeration value="DE" />
              <xs:enumeration value="ES" />
              <xs:enumeration value="UK" />
              <xs:enumeration value="US" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

小结

- ▶ XML 是一种元语言，允许用户借助标签为文档定义标记
- ▶ 标签的嵌套引入了结构。可以用 XML schema 或 DTD 来规定文档的结构
- ▶ 机器不能解读 XML 文档的语义
- ▶ 如果对词汇表存在共同的理解，则 XML 将支持协作和交流。因此，在小的专业领域，XML 有良好的实用性，但不适应全球性交流

Table of Contents

语义网概览

XML 结构化网络文档

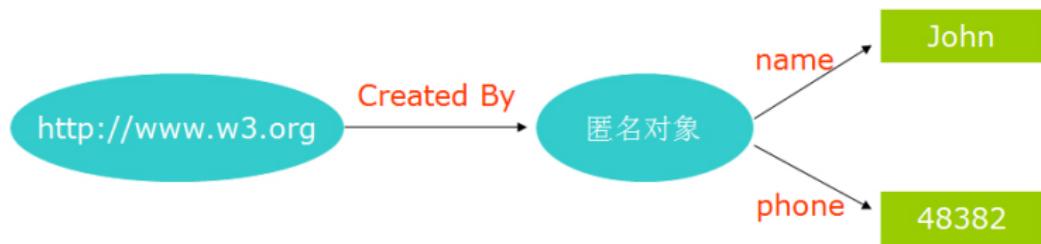
用 RDF 描述网络资源

网络本体语言 OWL

RDF 概述

- ▶ RDF (Resource Description Framework) 定义了一个简单的模型，用于描述资源，属性和值之间的关系。资源是可以用 URI 标识的所有事物，属性是资源的一个特定的方面或特征，值可以是另一个资源，也可以是字符串。总的来说，一个 RDF 描述就是一个三元组：对象 属性 值
 - ▶ 主体 谓词 客体
- ▶ 在 RDF 中，文档中的声明通常是某个事物 人、网页或其他任何东西对于某些值 另一个人、另一网页拥有某些属性（例如……是……的父母，……是……的作者）
- ▶ RDF 是一个机制，用于描述数据。它不是一个语言，而是一个模型，用于描述 Web 上数据
- ▶ RDF 是忽略语法的，它仅仅提供一个模型用于表达元数据。这种可能的表达可以是有向图，列表或其他，当然 XML 也是一种可选的表达

RDF 的有向图表示示例



RDF 的列表表示示例

Object	Attribute	Value
http://w3.org/# anonymous	created_by	#anonymous
# anonymous	name	“John”
# anonymous	phone	“477738”

RDF 的元组表示示例

```
hasName
( "http://www.ddddddddddd",
  "jime lenr" )
authorof
( "http://www.ddddddddddd",
  "http://book/ieiei" )
hasPrice
( "http://book/ieiei",
  "$83" )
```

RDF 的 XML 表示示例

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:uni="http://www.mydomain.org/uni-ns">

  <rdf:Description rdf:about="949318">
    <uni:name>David Billington</uni:name>
    <uni:title>Associate Professor</uni:title>
    <uni:age rdf:datatype="&xsd:integer">27<uni:age>
  </rdf:Description>

  <rdf:Description rdf:about="CIT1111">
    <uni:courseName>Discrete Maths</uni:courseName>
    <uni:isTaughtBy>David Billington</uni:isTaughtBy>
  </rdf:Description>

  <rdf:Description rdf:about="CIT2112">
    <uni:courseName>Programming III</uni:courseName>
    <uni:isTaughtBy>Michael Maher</uni:isTaughtBy>
  </rdf:Description>

</rdf:RDF>
```

- ▶ 虽然可以有很多种方式来表示 RDF 数据，RDF 数据的交换必须由一个固定有序的语法来支持。XML 是一个选择，而 RDF 规范使用的正是它
- ▶ 然而，RDF 数据模型并没有被绑定到一个特定的语法上，它可以用任何语法来表示，它也可以从非 RDF 的数据资源中抽取。用 XML 序列语法来表示的 RDF 很难理解，而 RDF 应用程序接口使开发者可以不管序列语法的具体细节，而把 RDF 数据当作是图表来进行处理

- ▶ RDF 被设计用来为元数据提供一个基本的对象、属性、值的数据模型
- ▶ 对于这些语义，RDF 并没有预先建模，同 XML 一样，RDF 数据模型没有提供声明属性名的机制
- ▶ RDF Schema 和 XML Schame，提供了一个词汇定义的方式，还可以定义哪些属性可以应用到哪些对象上。换句话说，RDF Schema 为 RDF 模型提供了一个基本的类型系统

RDF Schema 概述

- ▶ RDF Schema，使用了一些预先定义的词汇集，比如 class, subpropertyof, subclassof，来指定特定的 schema
- ▶ RDF Schema 是一个有效的 RDF 表达，就像 XML Schema 是一个有效的 XML 表达
- ▶ subclassof 允许开发者去定义每一个类的继承机制，
subpropertyof 对属性是一样的。属性的限制可以用 domain 和 range 结构来实现，这个结构可以用来扩展词汇表
- ▶ RDF 和 RDF Schema 可以用谓词逻辑刻画（为其子集），可以进行高效的自动推理

RDF Schema 示例

```
<rdfs:Class rdf:ID="lecturer">
<rdfs:comment>
The class of lecturers. All lecturers are academic staff members.
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#academicStaffMember"/>
</rdfs:Class>

<rdfs:Class rdf:ID="course">
<rdfs:comment>The class of courses</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="isTaughtBy">
<rdfs:comment>
Inherits its domain ("course") and range ("lecturer")
from its superproperty "involves"
</rdfs:comment>
<rdfs:subPropertyOf rdf:resource="#involves"/>
</rdf:Property>

<rdf:Property rdf:ID="phone">
<rdfs:comment>
It is a property of staff members
and takes literals as values.
</rdfs:comment>
<rdfs:domain rdf:resource="#staffMember"/>
<rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
```

RDF(S) 表达的局限

- ▶ RDF 局限于二元常谓词，RDFS 局限于子类分层和属性分层
- ▶ 不能表达的内容
 - ▶ 等价（不等价）关系
 - ▶ 枚举
 - ▶ 布尔运算：与、或、非
 - ▶ 数量限制：
 - ▶ 单值/多值
 - ▶ 可选/必选值
 - ▶ 自反、对称、传递
 - ▶

小结

- ▶ RDF 描述的 statement 为三元组：资源 属性 值，主体 谓词 客体
- ▶ RDF 有一个基于 XML 的语言
- ▶ RDF 是领域无关的，RDF Schema 提供了一种描述具体领域知识的机制
- ▶ RDF Schema 是最原始本体语言，提供了一套具有固定语义的建模原语
- ▶ RDF Schema 的核心概念是类、子类关系，属性、子属性关系
- ▶ RDF 和 RDF Schema 可以用谓词逻辑形式化刻画，存在高效的自动推理机

Table of Contents

语义网概览

XML 结构化网络文档

用 RDF 描述网络资源

网络本体语言 OWL

为什么需要 Ontology

- ▶ Ontology 为人类和应用程序系统提供了一个对于主题的共同理解
- ▶ Ontology 为了不同来源的信息的合成，提供了一个共同的相关领域的理解
- ▶ Ontology 为了在不同的应用程序之间共享信息和知识（用于互操作），描述应用程序的领域，定义术语及其关系
- ▶ Ontology 刻画领域的公共知识：基本概念和关系，以及基于它们的规则

本体的推理

▶ 本体所需的推理：

- ▶ 类属关系：若 x 是类 C 的实例， C 是 D 的子类，则可推出 x 是 D 的实例
- ▶ 类等价：若类 A 与 B 等价，类 B 与 C 等价，则 A 与 C 等价
- ▶ 相容：假设 x 是类 A 的实例， A 是 $B \cap C$ 的子类，也是 D 的子类。若 $B \cap D = \emptyset$ ，则本体不相容、出错
- ▶ 分类：如果声明一组特定的属性 值关系是称为类 A 成员的充分条件，那么，当个体 x 满足所有条件时，可推出 x 是 A 的实例

▶ 推理工具的作用：

- ▶ 检查本体和知识的相容性
- ▶ 检查类间的隐含关系
- ▶ 对实例进行自动分类

网络本体语言 OWL (Web Ontology Language) 概述

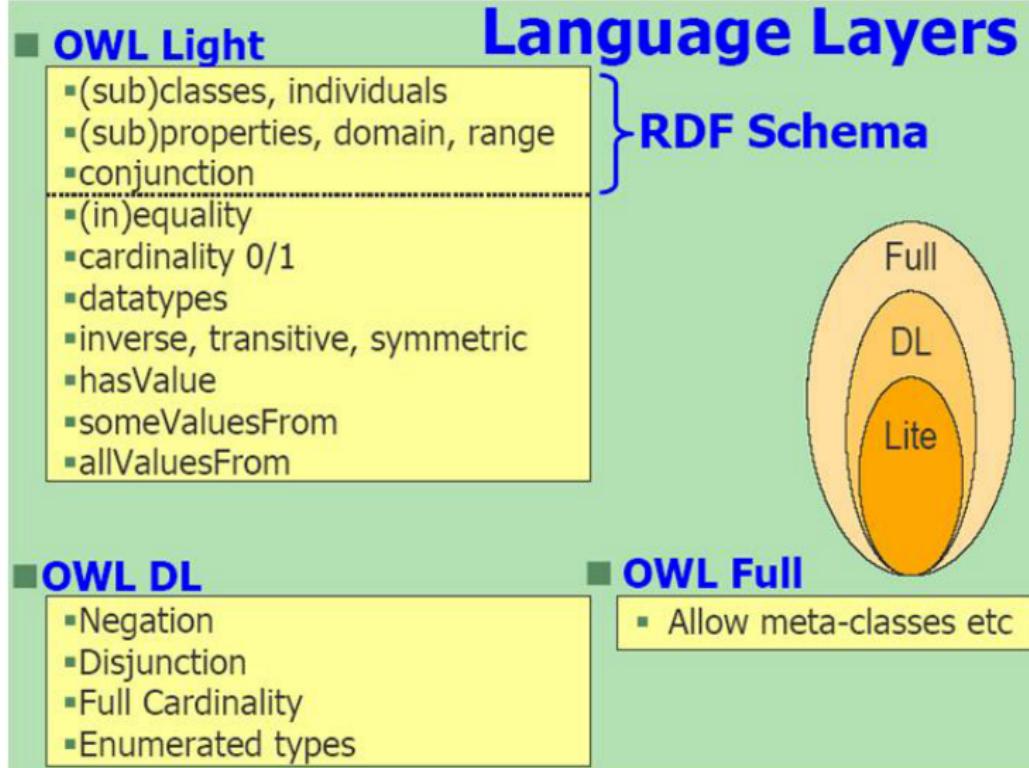
- ▶ OWL 建立在 RDF 之上
- ▶ OWL 是为处理网络的上信息
- ▶ OWL 被设计为便于被计算机理解
- ▶ OWL 并不便于被人阅读
- ▶ OWL 通过 XML 描述
- ▶ OWL 可以映射到描述逻辑 (description logic)，是一阶逻辑的子集，可判定，具有自动推理机，如 FaCT 和 RACER

OWL 的三个子语言

- ▶ OWL Lite
 - ▶ OWL DL 的一个子集，良好的可计算性
 - ▶ 提供给那些只需要一个分类层次和简单约束的用户
- ▶ OWL DL
 - ▶ 基于描述逻辑，与一部分 RDF 不兼容，旨在支持已有的描述逻辑商业处理（business segment）和具有良好计算性质的推理系统
 - ▶ 包括了 OWL 语言的所有成分，但有一定的限制，如类型的分离
 - ▶ 保证计算的完全性，可判定性，计算复杂度较高，暂时没有完全的推理机实现
- ▶ OWL Full
 - ▶ 最强的表达能力，不可判定
 - ▶ 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的用户

在表达能力和推理能力上，每个子语言都是前面的语言的扩展

OWL 三层语言



OWL: constructors

Constructor	Abbreviation	Example
intersectionOf	$C_1 \wedge \dots \wedge C_n$	Human \wedge Male
unionOf	$C_1 \vee \dots \vee C_n$	Doctor \vee Lawyer
complementOf	$\neg C$	\neg Male
oneOf	$\{x_1 \dots x_n\}$	{john, mary}
toClass	$\forall P.C$	\forall hasChild.Doctor
hasClass	$\exists P.C$	\exists hasChild.Lawyer
hasValue	$\exists P.\{x\}$	\exists citizenOf.{USA}
minCardinalityQ	$\geq n P.C$	≥ 2 hasChild.Lawyer
maxCardinalityQ	$\leq n P.C$	≤ 1 hasChild.Male
cardinalityQ	$= n P.C$	$= 1$ hasParent.Female

+ XML Schema datatypes:

- int, string, real, etc

OWL: Axioms

Axiom	Abbreviation	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \wedge Biped
sameClassAs	$C_1 \doteq C_2$	Man \doteq Human \wedge Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
samePropertyAs	$P_1 \doteq P_2$	cost \doteq price
sameIndividualAs	$x_1 \doteq x_2$	President_Bush \doteq G_W_Bush
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
inverseOf	$P_1 \doteq P_2^-$	hasChild \doteq hasParent $^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor $^+$ \sqsubseteq ancestor
uniqueProperty	Thing $\sqsubseteq \leqslant 1 P$	Thing $\sqsubseteq \leqslant 1$ hasMother
UnambiguousProperty	Thing $\sqsubseteq \leqslant 1 P^-$	Thing $\sqsubseteq \leqslant 1$ isMotherOf $^-$

☞ Axioms (mostly) reducible to subClass/PropertyOf

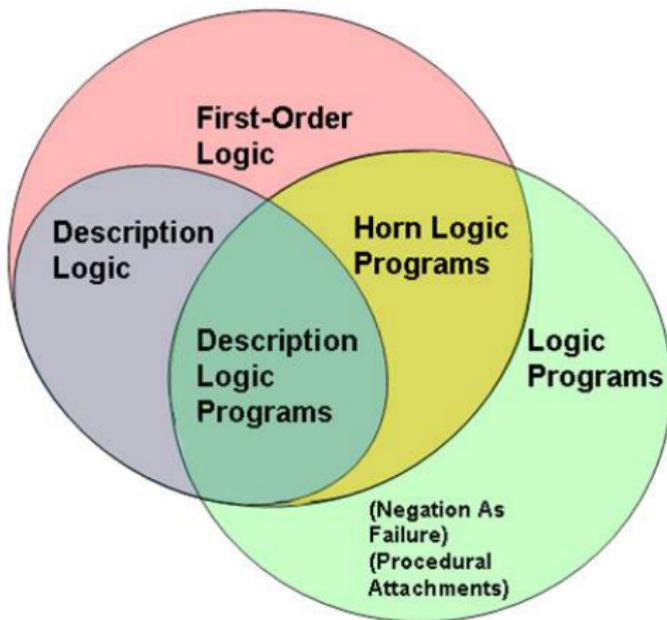
OWL 示例

```
<?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#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xml:base="http://wasp.cs.vu.nl/sekt/ontology/animal">
<owl:Ontology rdf:about= "animal"/><owl:Class rdf:ID="Eagle">
    <rdfs:subClassOf><owl:Class rdf:about="#Bird"/>
</rdfs:subClassOf></owl:Class><owl:Class rdf:ID="Animal"/>
<owl:Class rdf:ID="Fly"><owl:disjointWith>
    <owl:Class rdf:about="#Penguin"/></owl:disjointWith>
    <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class><owl:Class rdf:ID="Bird">
    <rdfs:subClassOf rdf:resource="#Fly"/>
</owl:Class>
<owl:Class rdf:ID="Penguin">
    <rdfs:subClassOf rdf:resource="#Bird"/>
    <owl:disjointWith rdf:resource="#Fly"/>
</owl:Class>
</rdf:RDF>
```

描述逻辑与 OWL 关系

语言构造成份	SHOQ(D)语法	OIL关键字	DAML+OIL关键字	OWL关键字
原子概念	A	class-def	Class	Class
关系	R	slot-def	rdfs:Property	rdfs:Property
概念合取	C \wedge D	and	intersectionOf	intersectionOf
概念析取	C \vee D	or	unionOf	unionOf
概念取反	$\neg C$	not	complementOf	complementOf
存在约束	$\exists R.C$	domain	hasClass	someValuesFrom
值约束	$\forall R.C$	range	toClass	allValuesFrom
全概念	\top	top	Thing	Thing
空概念	\perp	Bottom	Nothing	Nothing
最小基数约束	$(\geq n S.C)$	min-cardinality	minCardinalityQ	minCardinality
最大基数约束	$(\leq n S.C)$	max-cardinality	maxCardinalityQ	maxCardinality
命名	O	has-Value	hasValue	hasValue
数据类型	D	Value-type	rdf:datatype	rdf:datatype

KR 语言之间关系



小结

- ▶ OWL 是网络本体的建议标准，用于描述知识的语义以便供机器处理
- ▶ OWL 以 RDF 和 RDFS 为基础；使用基于 XML 的 RDF 语法，用 RDF 描述定义实例，使用了大部分 RDFS 建模原语
- ▶ OWL 可以映射到谓词逻辑和描述逻辑，从而具有形式语义，可以自动推理

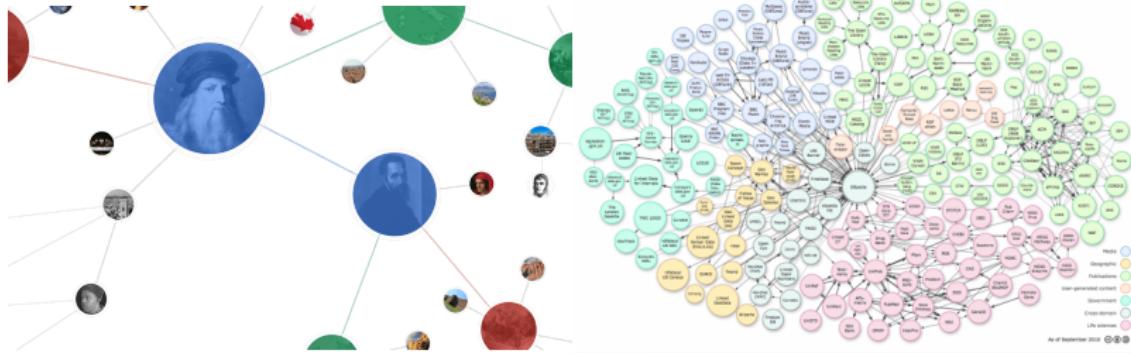
知识图谱 (Knowledge Graph)

- ▶ 知识图谱 (Knowledge Graph, KG) 是结构化的语义知识库，以符号形式描述客观世界中概念、实体及其关系，提供了一种更好地组织、管理和理解海量信息的能力
- ▶ 知识图谱的基本组成形式为“实体-关系-实体”三元组，以及实体及其相关“属性-值”对，实体间通过关系相互联结，构成网状的知识结构
- ▶ 知识图谱最早由 Google 于 2012 年 5 月发布，用于智能化搜索功能，对搜索结果进行系统的知识整理
- ▶ 知识图谱既来源于知识表示与推理领域中语义网络 (Semantic Networks)、描述逻辑 (Description Logics)、概念图 (Conceptual Graphs)，也来源于数据库系统中网状数据库 (Network Databases) 和三元组存储 (Triple Stores)
- ▶ 知识图谱既可以从 KR 的角度，看作 RDF (Resource Description Framework) 三元组的知识库，采用 RDF 查询语言如 SPARQL 进行查询和推理；也可以从图数据库 (Graph Database) 的角度，看作属性图 (Property Graph) 表示，在图数据库如 Neo4J 中存储，并用相应的查询语言 Cypher 进行查询

知识图谱

- ▶ 知识图谱既可以从 RDF 的角度进行 KR 知识库的构建和形式化推理，也可以从属性图的角度进行高效存储和查询，并采用一系列学习方法进行知识图谱的构建、推理、融合和补全，甚至可以进行知识图谱嵌入 (KG Embedding)，将知识图谱知识转换为向量表示，进一步与深度学习技术融合
- ▶ 区别于大部分 KR 方式，知识图谱具有以下特点：
 - ▶ (1) 可大规模扩展 (Scalable)，在保持高效查询和推理的条件下支持大规模知识库。例如，GDELT 全局知识图谱 (Global Knowledge Graph) 拥有 35 亿条知识，描述 3.64 亿个实体
 - ▶ (2) 自底向上构造 (Bottom-up construction)，可以从底层数据中提取知识，自底向上的构造更符合问题需求的知识图谱，不过度依赖人工进行自顶向下的设计
 - ▶ (3) 支持多种混合方式构造 (Mixed models of construction)，知识图谱既可以基于人工规则和数据库进行冷启动，也可以采用多种学习方法，灵活的从结构化数据、文本、图像等数据来源进行自动生成和扩展

知识图谱



Google search results for "the hobbit":

The Hobbit: An Unexpected Journey (2012) - IMDb
★★★★★ Rating: 8.3/10 (1,046,400 votes)
A young hobbit, Bilbo Baggins, sets out on a "unexpected journey" to the Lonely Mountain with a powerful group of Dwarves to reclaim their home from the dragon Smaug.
Full cast and crew | Release dates | The Desolation of Smaug: Trivia and more | Watch on Amazon Prime Video

The Hobbit - Wikipedia, the free encyclopedia
The Hobbit is a series of three epic fantasy adventure films directed, co-written and produced by Peter Jackson and based on J. R. R. Tolkien's 1954 fantasy novel *The Hobbit*, minus

The Hobbit: An Unexpected Journey (2012) - Wikipedia
The Hobbit, or There and Back Again, better known by its abbreviated title *The Hobbit*, is a fantasy novel by J. R. R. Tolkien. The first of three novels in his epic Middle-earth legendarium.

The Hobbit (film series) - Wikipedia
The Hobbit is a series of three epic fantasy adventure films directed, co-written and produced by Peter Jackson and based on J. R. R. Tolkien's 1954 fantasy novel *The Hobbit*. It is a prequel to the 2001 film *The Lord of the Rings: The Fellowship of the Ring*.

The Hobbit (film series) - The Lord of the Rings - Hobbit (1977 film) - Adaptations
The Hobbit (film series) | The Lord of the Rings - Hobbit (1977 film) | Adaptations

The Hobbit (film series) - Wikipedia, the free encyclopedia
The Hobbit (film series) is a series of three epic fantasy adventure films directed, co-written and produced by Peter Jackson and based on J. R. R. Tolkien's 1954 fantasy novel *The Hobbit*. It is a prequel to the 2001 film *The Lord of the Rings: The Fellowship of the Ring*.

The Hobbit: An Unexpected Journey (2012) - Wikipedia, the free encyclopedia
The Hobbit: An Unexpected Journey is a 2012 epic fantasy adventure film directed by Peter Jackson and based on J. R. R. Tolkien's 1954 fantasy novel *The Hobbit*.

The Hobbit: An Unexpected Journey (2012) - Google Play Books
The Hobbit: An Unexpected Journey is a 2012 epic fantasy adventure film directed by Peter Jackson and based on J. R. R. Tolkien's 1954 fantasy novel *The Hobbit*.

Google search results for "the hobbit book":

the hobbit book
the hobbit book
the hobbit book online
the hobbit book summary
the hobbit book review

Ad related to the hobbit book
Buy The Hobbit (film series) by J.R.R. Tolkien, ISBN 9780007314414
Up to 50% Discount, Free Worldwide Delivery, Order Now!

The Hobbit - Wikipedia, the free encyclopedia
The Hobbit is a series of three epic fantasy adventure films directed, co-written and produced by Peter Jackson and based on J. R. R. Tolkien's 1954 fantasy novel *The Hobbit*, minus

The Hobbit (film series) - The Lord of the Rings - Hobbit (1977 film) - Adaptations
The Hobbit (film series) | The Lord of the Rings - Hobbit (1977 film) | Adaptations

Five things changed/expanded from the book for 'The Hobbit' film
www.cnn.com/2012/12/14/... - hobbit-book-movie/index.html
"One Day" author Jojo Moyes says she was surprised by the longer length of the first instalment of "The Hobbit" film trilogy because the source book is a children's book.

Images for the hobbit book

The Hobbit or There and Back Again, J.R.R. Tolkien, Peter Jackson
www.amazon.com/... - hobbit-book-movie/index.html
I purchased the Kindle edition of The Hobbit and it was a really nice read for a change

This are all listed in Google SERPs