



# 语义网与知识图谱

上海大学计算机学院

主讲：刘 炜

2020/9/7



# 课程目标

本课程为智能科学与技术专业选修课，通过课程学习，达到以下目标：

- （1）了解知识表示、语义网基本概念。
  - （2）掌握RDF, RDFs, OWL和Rules等语义知识表示语言，熟练掌握描述逻辑的语法和语义，掌握本体概念和本体建模方法。
  - （3）了解知识图谱基本概念，理解知识图谱的技术内涵和外延。
  - （4）掌握知识图谱的表示和关联建模、存储和关联查询，知识图谱的获取和语义关系抽取技术，知识图谱的简单推理等。
  - （5）掌握领域本体和知识图谱构建技术，并开展简单的应用。
-

# 课程安排

第一讲：语义网概述

第二讲：RDF和RDFs

第三讲：本体论与OWL语言

第四讲：描述逻辑与描述逻辑语义

第五讲：知识图谱概览

第六讲：知识图谱的表示与关联建模

第七讲：知识图谱的存储与查询

第八讲：知识图谱的获取与语义抽取

第九讲：知识图谱推理（可选）

第十讲：项目报告

---

# 实验安排

**实验一、二：**RDF，RDFS表示语言的使用

**实验三：**本体建模工具Protege使用及领域本体构建

**实验四：**基于描述逻辑的知识建模

**实验五：**玩转第一个知识图谱——RDF图数据库实例

( Jena/TDB/RDF/SPARQL )、属性图数据库实例

( Neo4J/Property Graph/Cypher )。

**实验六：**实体抽取和关系抽取

**实验七、八、九：**知识图谱应用课程项目

**实验十：**项目验收与报告

---

# 课程考核

总评成绩=出勤（10%）+ 实验与作业（10%）+ 课程项目（10%）  
+ 笔试（70%）

注：出勤1次得1分，迟到和早退得0.5分，满分10分，低于6分计0分。

实验与作业：每周布置的实验课作业与任务完成情况

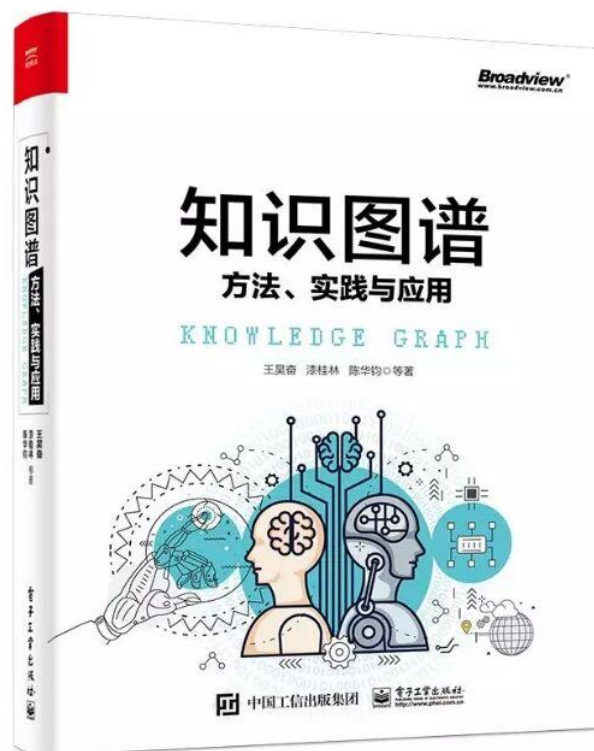
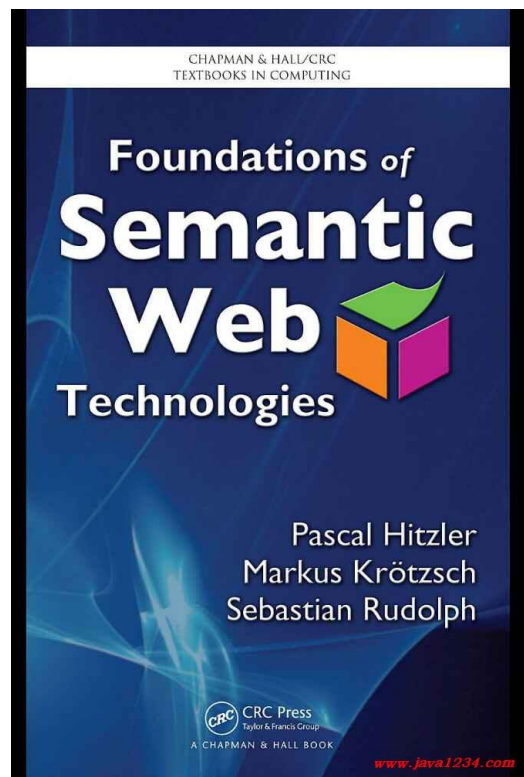
课程项目：包括项目验收与报告

笔试：闭卷

---

# 学习资源

- ❖ [www.elearning.shu.edu.cn](http://www.elearning.shu.edu.cn)
- ❖ [www.openkg.org](http://www.openkg.org)
- ❖ 语义网与知识图谱 课程空间





## 语义网概述

上海大学计算机学院 刘炜

2020/9/7

# 内容概览

---

**一、语义网概述**

**二、RDF**



# WWW渗透到社会各个方面

---

- 社会联系 (社交网络平台, 微博, 微信, Facebook等)
  - 经济生活 (购买, 销售, 广告 ...)
  - 管理(电子政务)
  - 教育(eLearning, Mooc, ...)
  - 工作(OA系统, ERP, workflows)
  - 娱乐 (游戏、音乐、 视频等 ...)
- 网络1.0: 文件网  
[Web1.0: Web of documents](#)
  - 网络2.0: 人际/社会网  
[Web2.0: Web of persons](#)
  - 网络3.0: 数据网  
[Web3.0: Web of data \(semantics\)](#)



# 当前的Web

---

- 取得了巨大的成功(Immensely successful)
- 产生了海量的数据(Huge amounts of data)
- 有了用于结构化数据传输的标准语法 ( XML ) (Syntax standards for transfer of structured data)
- 大量机器可处理，人类可读的文档(Machine-processable, human-readable documents).

## 但是：

- Content/knowledge cannot be accessed by machines.
- Meaning (semantics) of transferred data is not accessible.

# 当前的Web存在的问题

---

- **信息太多但是结构化太少**
- **信息多为满足人类消费（使用）**
  - – 内容搜索非常简单
  - – 未来需要更好的方法
- **Web内容多为异构**
  - in terms of content 内容异构
  - in terms of structure 结构异构
  - in terms of character encoding 编码方式异构
  - future requires intelligent information integration 未来需要智能化的信息集成
- **人类可以从给定的信息片段演绎推理出隐藏的信息**
- **但是目前的Web只能处理文档中的语法**
  - – 需要自动推理技术

# 几个例子

---

- Find that landmark article on data integration written by an Indian researcher in the 1990s.  
[如果你不知道答案，如何去处理这个问题？]
- Are lobsters spiders?  
[这个问题目前回答起来比较简单，但是在若干年前是无法回答的。它仍然需要在不同的网站去发现和集成相关的背景知识]
- Which car is called a “duck” in German?  
[需要能够从不同网站智能地集成网页信息，加上一定的背景知识]

**语义网就是一种描述网络Web数据，具有模型语义，并且在一定程度上支持语义推理的知识表示方法。**

# 语义Web的基本组成

---

- 描述Web信息的开放标准 ( Open Standards for describing information on the Web )
- 从Web描述信息中进一步获取语义的方法 ( Methods for obtaining further information from such descriptions )

# 语义Web的基本组成

---

- 从Web描述信息中进一步获取语义的方法 ( Methods for obtaining further information from such descriptions )

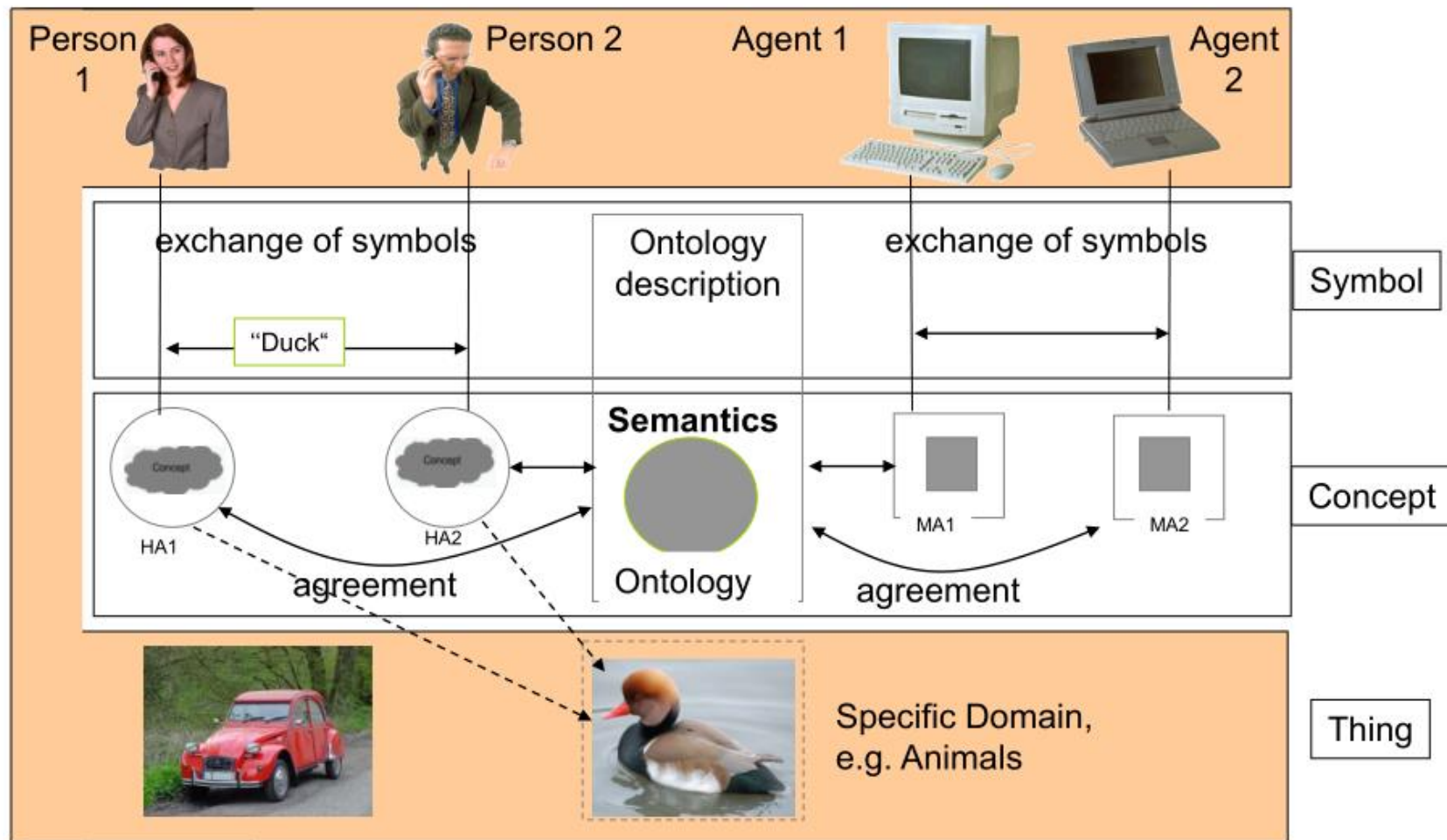
**主要方法: 逻辑演绎 (又称自动推理——automated reasoning)**

**E.g.,  
D.C. is a capital  
Every capital is a city**

-----  
**Hence: D.C. is a city**

**基于谓词逻辑，需要明确什么结论是有效的，另外，我们需要相应的算法。**

# 语义Web基本思想



# 语义Web基本思想

---

**人类之所以能用自然语言思维，能用自然语言互相交流，关键在于每个人的大脑中，有一个结构相同、内容近似、涵盖丰富、查询和推理功能强大并能方便提升的知识库系统。**

**对机器而言，也是同样的，机器之间要相互理解和沟通。也需要一个这样的知识库，这个知识库我们通常称之为**本体**！**

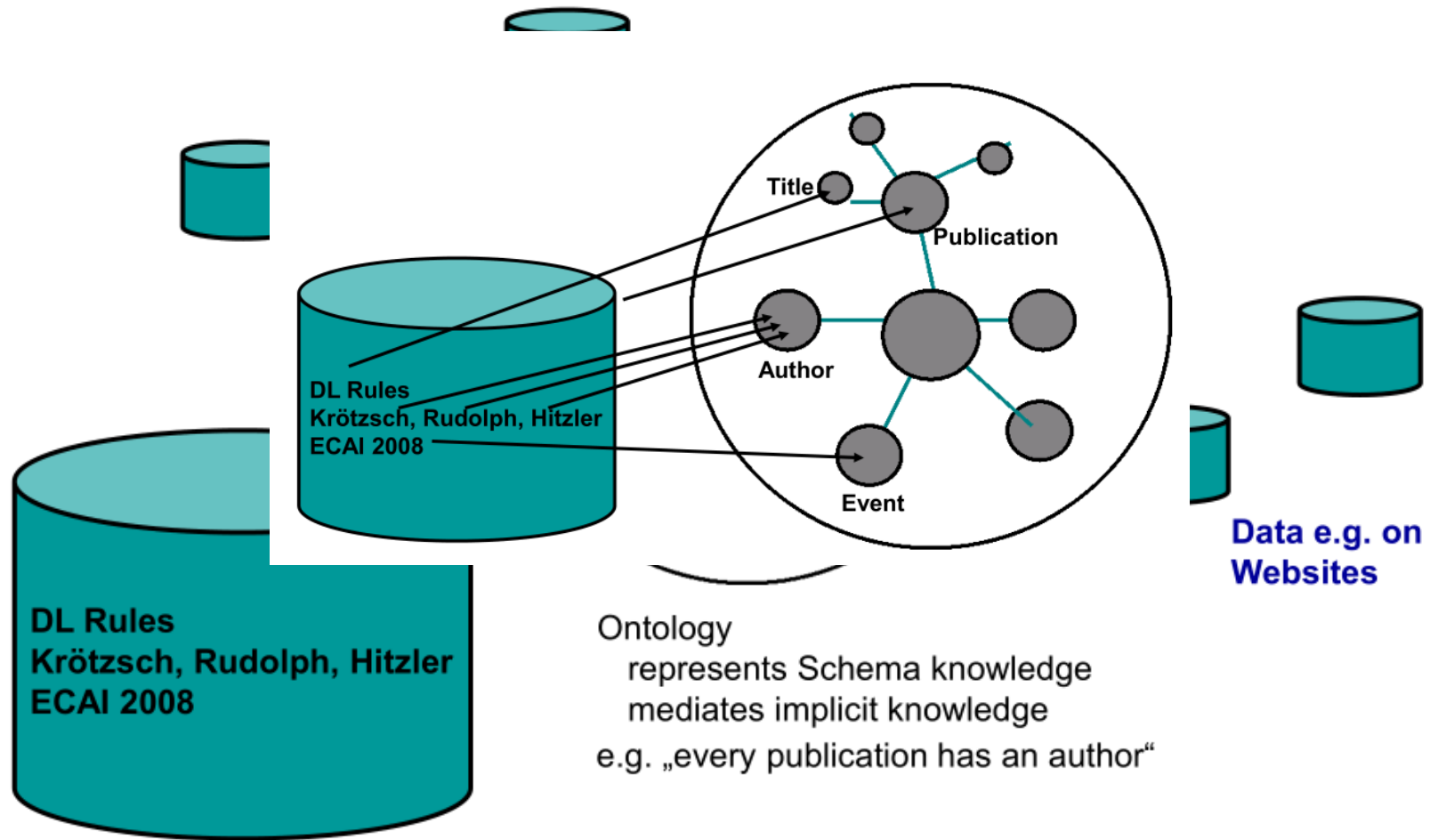


# 什么是本体（ Ontology ）

---

- 本体是指一种 “**形式化**的，对于共享概念体系的明确而又详细的说明” 。
- 本体提供的是一种**共享词表**，也就是特定领域之中那些存在着的对象类型或概念及其属性和相互关系；
- 或者说，本体就是一种特殊类型的术语集，具有结构化的特点，且更加适合于在计算机系统之中使用；
- 或者说，本体实际上就是对特定领域之中某套概念及其相互之间关系的形式化表达（ formal representation ）。

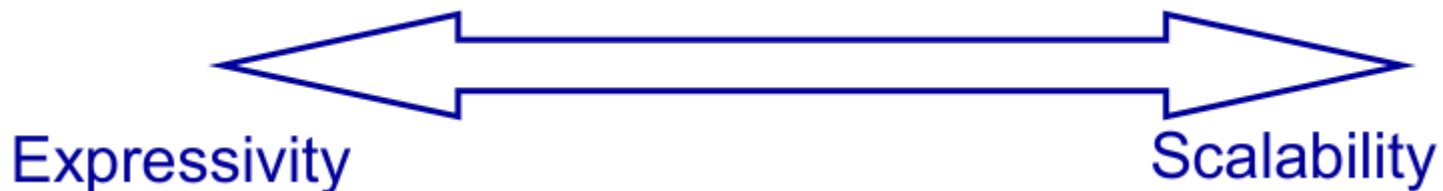
# 语义Web基本思想



# 本体语言

---

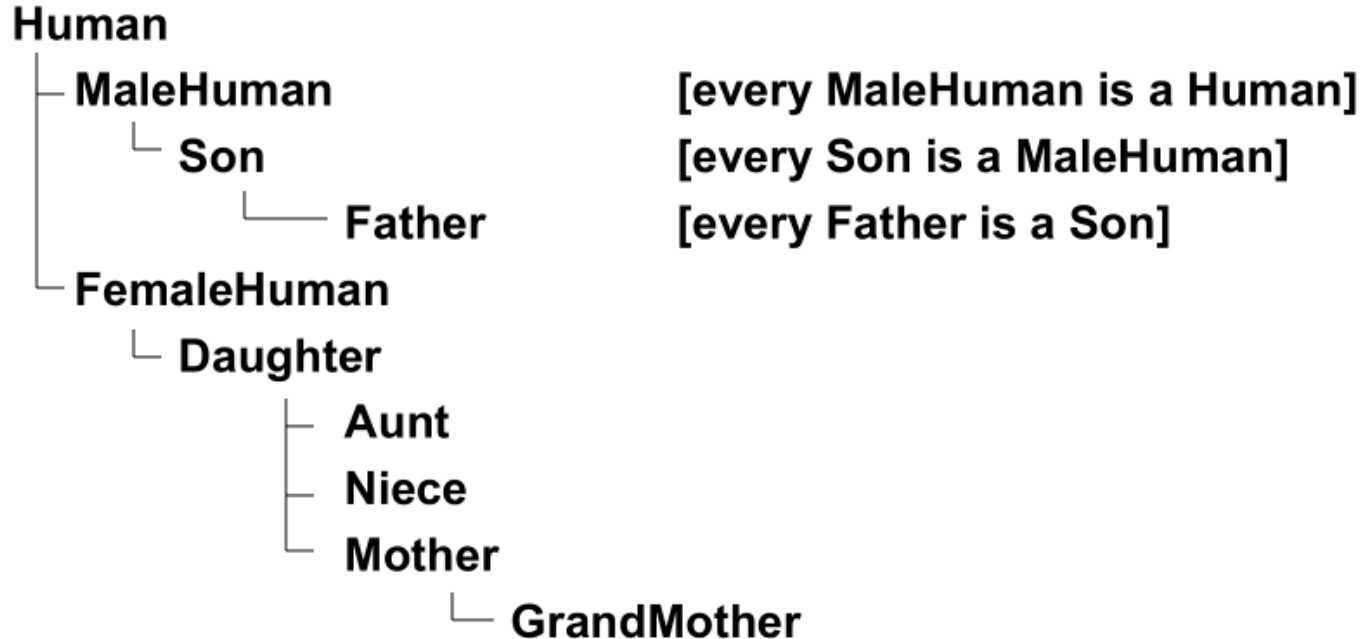
- 实现语义技术最重要的是有合适的知识表示语言（ Of central importance for the realisation of Semantic Technologies are suitable representation languages ）
- 含义（ Meaning,语义 ） 通常通过逻辑和推理算法实现。  
Meaning (semantics) provided via logic and deduction algorithms (automated reasoning).
- 本体语言的可伸缩性（ Scalability ） 具有挑战。 Scalability is a challenge.



# 本体

---

- 本体的核心通常是分类体系（Taxonomy）：
  - – 事物的类，以层次结构进行组织（classes of things, arranged in a hierarchy）



# Partonomy vs. Taxonomy

---

- 以下是部分-整体分类体系 (Partonomy , not a taxonomy):
  - 事物的类，按照“部分-整体”形式构造层次结构— classes of things, arranged in a hierarchy of “part-of” relationships

## America

└ LatinAmerica

└└ SouthAmerica

└└└ Brazil

[Brazil is *part of* South America]

└ NorthAmerica

└└ USA

└└└ Indiana

└└└ Florida

└└└ Ohio

└└└└ GreeneCounty

# Partonomy vs. Taxonomy

---

- **Partonomy:**

**A is part of B**

**hand is part of body**

**Germany is part of Europe**

**Wing is part of aircraft**

**Engine is part of car**

- **Taxonomy**

**every A is a B**

**every father is a man**

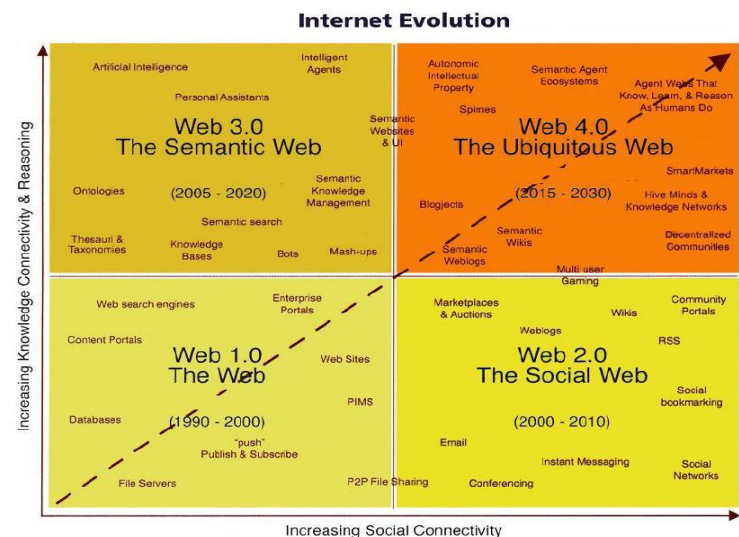
**every dog is a mammal**

**every bottle is a container**

**every arm is a limb**

# 语义网发展历史

- 1998年由Tim Berners-Lee提出
- 1999,2000年: W3C metadata activity (提出 RDF(S))
- W3C 语义Web activity: chartered 2001.
  - USA: DAML-Programme 2000-2005 , approx. \$90M.
  - Many large scale EU projects since 2002 and ongoing.
- 欧盟第六框架、欧盟第七框架 FP6/FP7 ( LarKC项目 , 大规模知识加速器 )
- 大量IT公司和投资机构对语义网投入资金。



# 各种Semantic Web项目

---

 Freebase

PKUBASE

 LINKING OPEN DATA  
W3C SWEO Community Project

ZHISHI.me

schema.org

....the new SEO?

NELL

 DBpedia

X LORE

  
WIKIDATA

WEB CHILD

CN-DBpedia

WEBKB

Herbnet

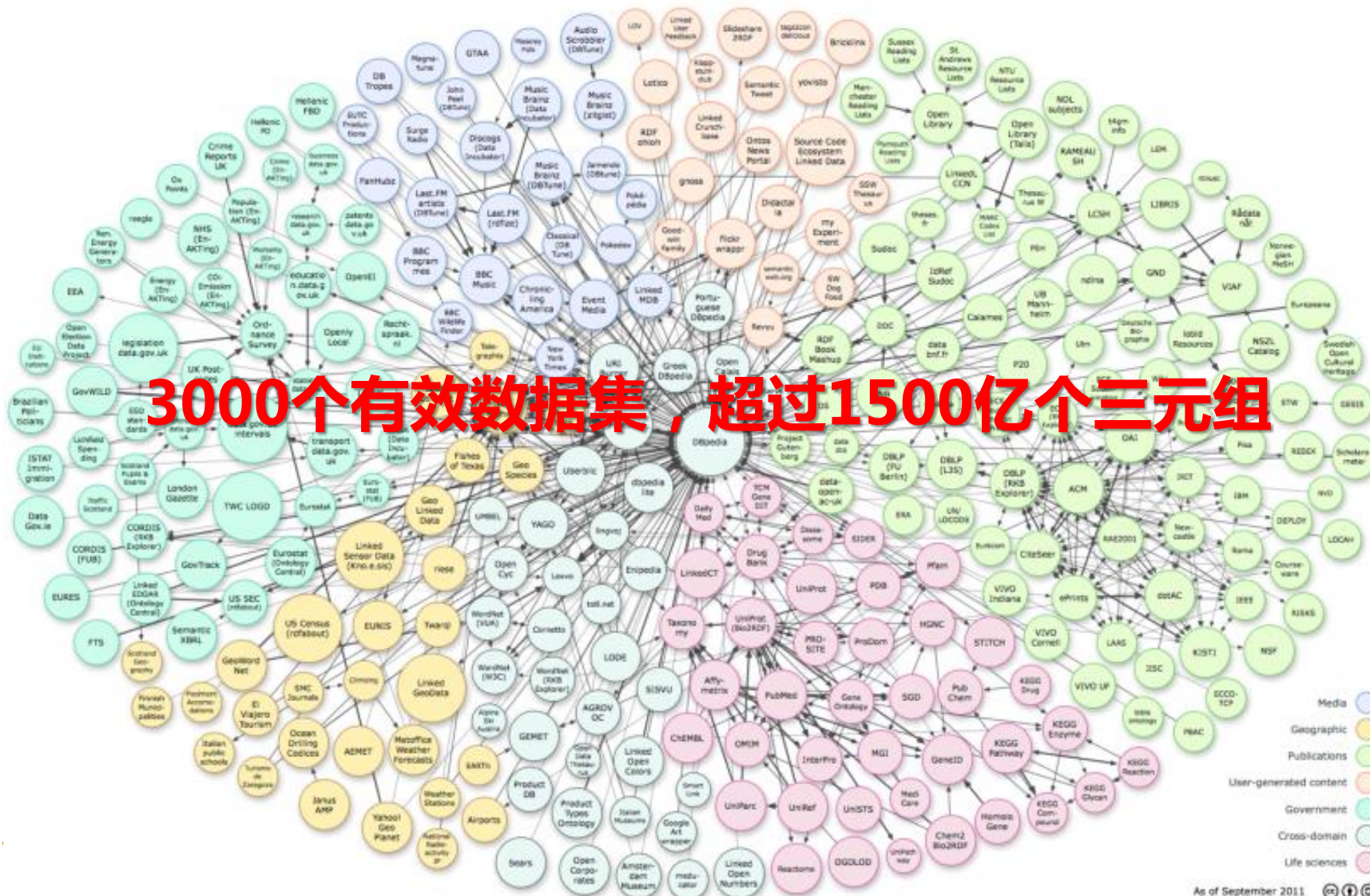
yago  
select knowledge

 LinkedGeoData.org

linked life data  




# Linked Open Data



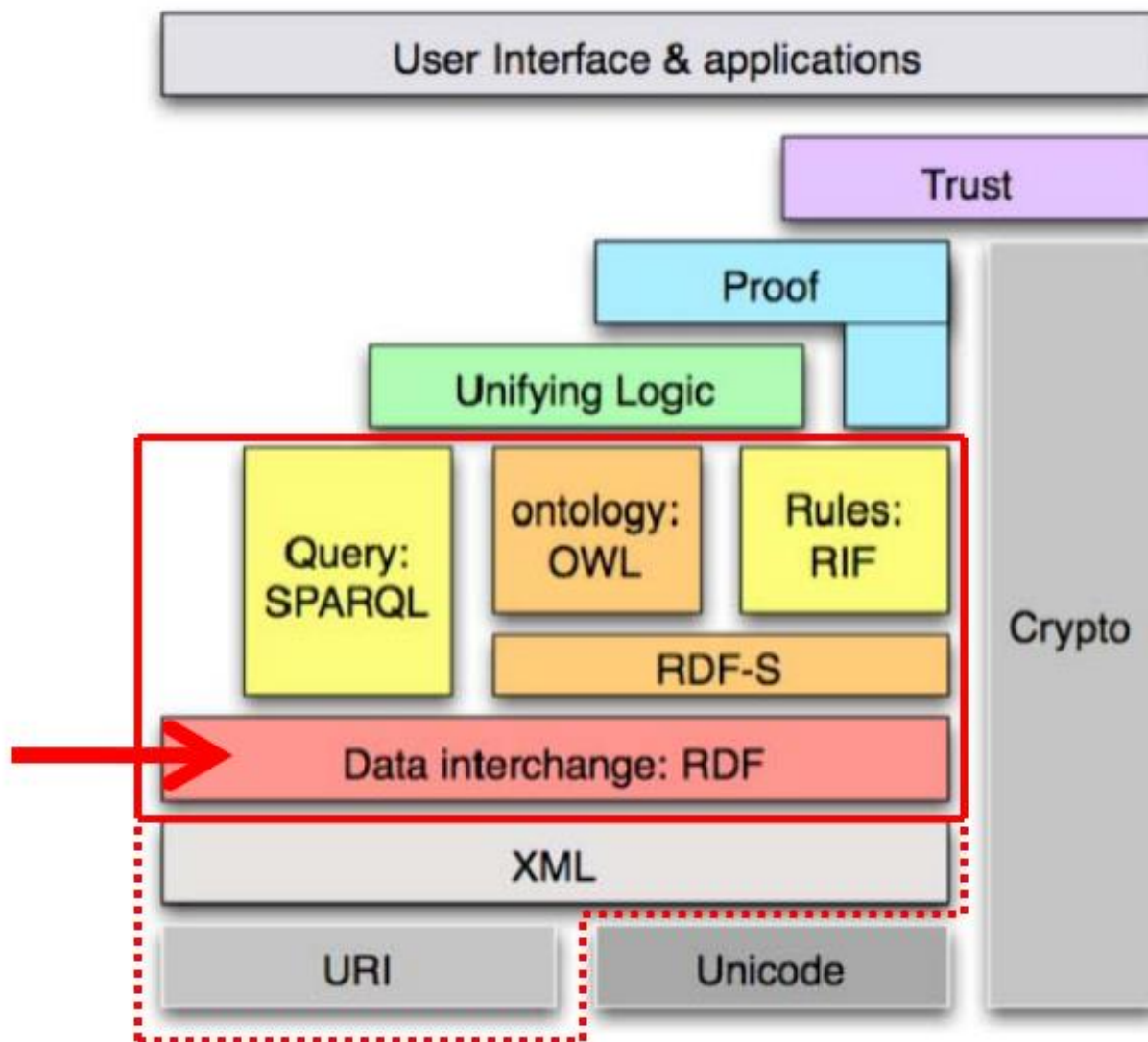
# 内容概览

---

一、语义网概述

二、RDF(S)

# RDF 在语义Web中的位置



# RDF知识点

---

- 1. 动机Motivation
- 2. 三元组和图Triples and Graphs
- 3. RDF语法 RDF syntaxes: Turtle and RDF/XML
- 4. 数据类型Datatypes
- 5. n元关系n-ary relationships
- 6. 空节点Empty nodes
- 7. 列表Lists

# 为什么使用RDF?

---

用XML描述信息（知识），被编码成树结构，严格的树形层次结构，适合对文档中的信息进行表示，而且易于处理。

```
<book>
  <title>FOST</title>
  <publisher>CRC Press</publisher>
  <author>Pascal</author>
</book>
```

# 为什么使用RDF?

---

当把XML中的树状信息进行合并时，会变得累赘笨重，不易理解。

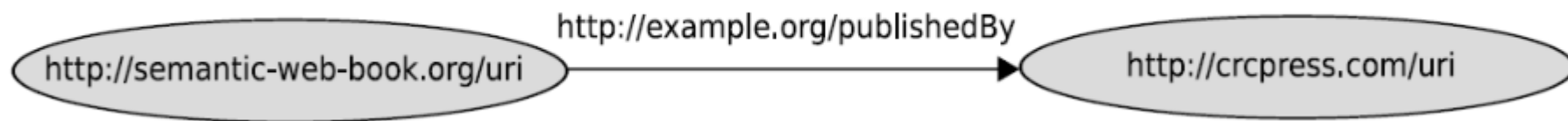
```
<publisher>
  <name>CRC Press</name>
  <book>
    <title>FOST</title>
  </book>
  <book>
    <title>...</title>
  </book>
</publisher>
```

在上例中可以看出，在任何情况下，出版社和书之间都不是层次上的从属关系。

# 为什么使用RDF?

---

## RDF使用（有向）图作为数据模型



RDF把对象之间的关系作为构建信息的基础，大量这样的实体关系很自然地构成了图，而不是层次化的树结构。RDF是为Web和其他电子网络的数据设计的一种描述语言。在这样的环境中，信息通常被分散存储和管理，合并不同来源的RDF数据变得更加容易。

# RDF

---

- **“Resource Description Framework”**
- **W3C Recommendation 2004**  
**<http://www.w3.org/RDF/>**
- **RDF is a data model**
  - originally for describing metadata for web pages, but has grown beyond that
  - structured information
  - universal, machine-readable data exchange format
  - main syntax uses XML for serialization



# RDF知识点

---

- 1. 动机Motivation
- **2. 三元组和图Triples and Graphs**
- 3. RDF语法 RDF syntaxes: Turtle and RDF/XML
- 4. 数据类型Datatypes
- 5. n元关系n-ary relationships
- 6. 空节点Empty nodes
- 7. 列表Lists

# RDF

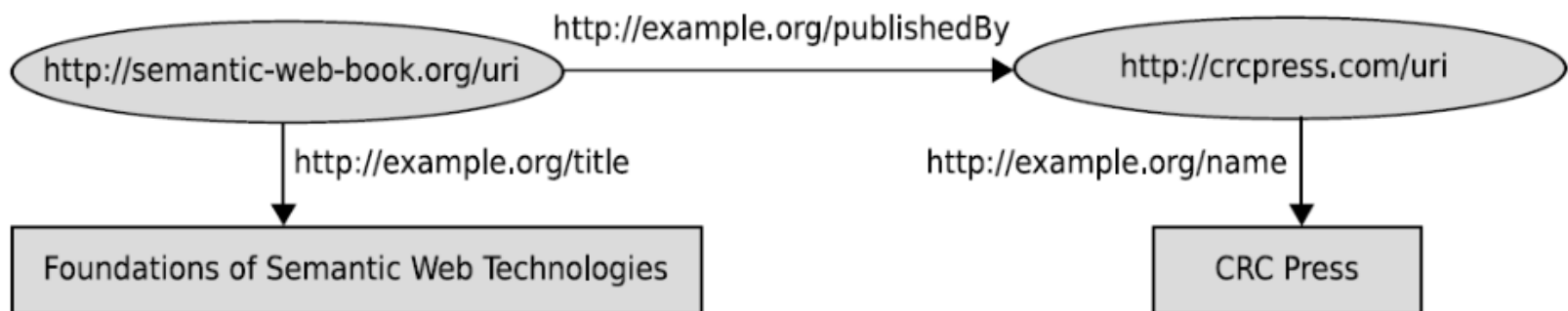
---

- **URIs**
  - for referencing resources 可以清楚地区分资源
- **Literals 文字**
  - data values 数据值
- **Empty nodes 空节点**
  - 没有命名的节点(or the name of which isn' t known)

# RDF— Literals

---

- 表示数据的值 for representing data values
- 采用字符串进行编码 encoded as strings
- 值通过数据类型进行解释 interpreted by means of datatypes
- 没有数据类型的字面体当做字符串进行处理 literals without datatype are treated the same as strings



# RDF

---

- 表示图有多种可能
- 一个图表示一系列的三元组（点-边-点）
- 一个三元组包含：



规则:

- 主语：URIs 和空节点
- 谓语: URIs (通常被称为属性properties)
- 宾语: URIs 、空节点或文字Literals

注：可以从一系列三元组重构图

# RDF知识点

---

- 1. 动机Motivation
- 2. 三元组和图Triples and Graphs
- **3. RDF语法 RDF syntaxes: Turtle and RDF/XML**
- 4. 数据类型Datatypes
- 5. n元关系n-ary relationships
- 6. 空节点Empty nodes
- 7. 列表Lists

# Turtle — 简要的RDF三元组语言

---

- 为RDF提供简单语法
- 三元组表示方法：
  - URIs 放在<>中
  - 文字用双引号 ""
  - 三元组用英文句号结束
  - 忽略空格

```
<http://semantic-web-book.org/uri>
  <http://example.org/publishedBy> <http://crcpress.com/uri> .
<http://semantic-web-book.org/uri>
  <http://example.org/title>
    "Foundations of Semantic Web Technologies" .
<http://crcpress.com/uri>
  <http://example.org/name>      "CRC Press" .
```

# Turtle

- 前缀缩写

```
@prefix book: <http://semantic-web-book.org/> .  
@prefix ex: <http://example.org/> .  
@prefix crc: <http://crcpress.com/> .  
  
book:uri    ex:publishedBy    crc:uri .  
book:uri    ex:title          "Foundations of Semantic Web Technologies" .  
crc:uri     ex:name            "CRC Press" .
```

- 进一步简化  
组合相同主语的三元组  
组合具有相同主语和谓语的三元组

```
@prefix book: <http://semantic-web-book.org/> .  
@prefix ex: <http://example.org/> .  
@prefix crc: <http://crcpress.com/> .  
  
book:uri    ex:publishedBy    crc:uri ;  
            ex:title          "Foundations of Semantic Web Technologies" .  
crc:uri     ex:name            "CRC Press", "CRC" .
```

**问题：上述turtle描述了几个三元组？**

# RDF的XML序列化

---

- Turtle 易于读写，但是不是实践中最常用的RDF语法。XML是web进行数据传输的基础,支持XML的工具或编程包很多。
- 因此，RDF的主要语法是基于XML的。
- Turtle不是W3C推荐的。
- RDF的规范化语法是其XML语法。



# RDF的XML-based语法

---

- 名字空间用来对标签进行消歧
- RDF语言中的标签通常有固定的名字空间，缩写为rdf

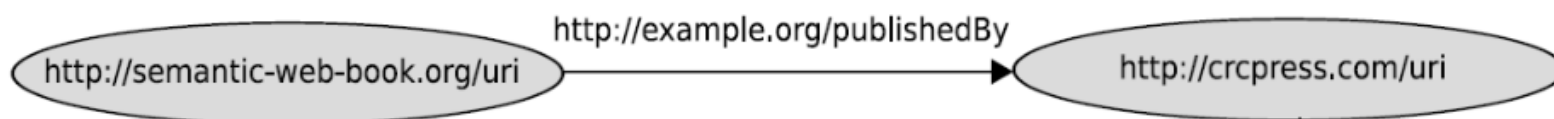
```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/">

  <rdf:Description rdf:about="http://semantic-web-book.org/uri">
    <ex:publishedBy>
      <rdf:Description rdf:about="http://crcpress.com/uri">
        </rdf:Description>
      </ex:publishedBy>
    </rdf:Description>

  </rdf:RDF>
```

# RDF的XML-based语法

---



**subject node**

**URI of the subject**

**property**

**object node**

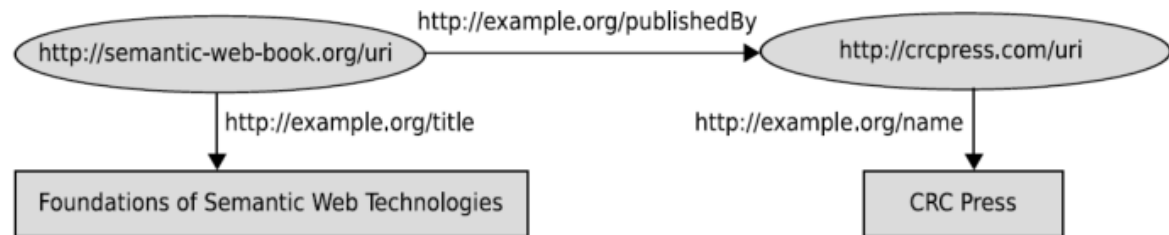
**URI of the object**

```
<rdf:Description rdf:about="http://semantic-web-book.org/uri">
  <ex:publishedBy>
    <rdf:Description rdf:about="http://crcpress.com/uri">
      </rdf:Description>
    </ex:publishedBy>
  </rdf:Description>
```

# RDF的XML-based语法

---

- 无类型的文字 ( Literals ) 可以是自由文本
- 一个主语可以包含多个属性
- 宾语的描述可用作更深层次三元组的主语 ( 嵌套 ) , 即宾语也可以描述为一个三元组。



```
<rdf:Description rdf:about="http://semantic-web-book.org/uri">
  <ex:title>Foundations of Semantic Web Technologies</ex:title>
  <ex:publishedBy>
    <rdf:Description rdf:about="http://crcpress.com/uri">
      <ex:name>CRC Press</ex:name>
    </rdf:Description>
  </ex:publishedBy>
</rdf:Description>
```

# RDF的XML-based语法

---

- 利用XML的属性值代替文字的表达
  - 属性名就是属性URI
- 通过属性标签内rdf:resource值表示的URIs表示宾语

```
<rdf:Description rdf:about="http://semantic-web-book/uri"  
    ex:title= "Foundations of Semantic Web Technologies">  
    <ex:publishedBy rdf:resource="http://crcpress.com/uri" />  
</rdf:Description>  
<rdf:Description rdf:about="http://crcpress.com/uri"  
    ex:Name="CRC Press" />
```

# RDF的XML语法

---

- 命名空间的使用是必要的，因为在XML名字中不允许使用冒号“:”，除非它与命名空间一起使用。
- 问题: 命名空间不能出现在XML的属性值，例如，在XML属性出现'book'将会出错，因为会被解释为URI模式。
- 解决方案: 使用 XML ENTITYs.

```
<?xml version="1.0" encoding="utf-8"?> <!DOCTYPE rdf:RDF[  
    <!ENTITY book 'http://semantic-web-book.org/'>  
]>  
  
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   
    xmlns:ex ="http://example.org/">  
  
    <rdf:Description rdf:about="&book;uri">  
        <ex:title>Foundations of Semantic Web Technologies</ex:title>  
    </rdf:Description>  
  
</rdf:RDF>
```

# RDF的XML语法

---

- 基础命名空间 ( base namespace ) 的使用。

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex ="http://example.org/"
  xml:base ="http://semantic-web-book.org/" >

  <rdf:Description rdf:about="uri">
    <ex:publishedBy rdf:resource="http://crcpress.com/uri" />
  </rdf:Description>

</rdf:RDF>
```

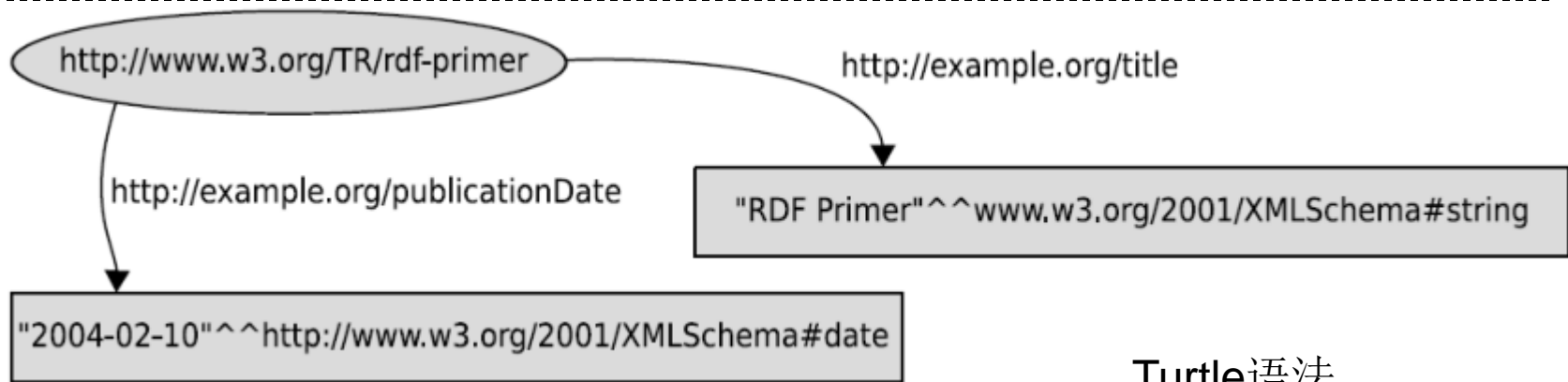
rdf:about="uri" 等同于rdf:about="http://semantic-web-book.org/uri"

# RDF知识点

---

- 1. 动机
- 2. 三元组和图
- 3. RDF语法：Turtle and RDF/XML
- **4. 数据类型Datatypes**
- 5. 多值关系 (n-ary relationships)
- 6. 空节点 (Empty nodes)
- 7. 列表Lists

# RDF的数据类型 ( Datatypes )



Turtle语法

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://www.w3.org/TR/rdf-primer>
  <http://example.org/title> "RDF Primer"^^xsd:string ;
  <http://example.org/publicationDate> "2004-02-10"^^xsd:date .
```

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-primer">
  <ex:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    RDF Primer
  </ex:title>
  <ex:publicationDate
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    2004-02-10
  </ex:publicationDate>
</rdf:Description>
```



# RDF的数据类型 ( Datatypes )

---

- 数据类型通常使用XML Schema
- 相同的数据值会有不同的表示方式：

`"3.14"^^xsd:decimal` 等同 `"+03.14"^^xsd:decimal`

但是

`"3.14"^^xsd:string` 不同于 `"+03.14"^^xsd:string`

- 大多数常见的XML数据类型允许在RDF中得到一个有意义的解释。
- `Rdf:XMLLiteral`是RDF唯一内嵌数据类型，它允许嵌入形式良好的XML片段作为RDF的文字值。

```
<rdf:Description rdf:about="http://semantic-web-book/uri">
  <ex:title rdf:parseType="Literal">
    Foundations of
    <br />
    <b>Semantic Web Technologies</b>
  </ex:title>
</rdf:Description>
```

# RDF知识点

---

- 1. 动机
- 2. 三元组和图
- 3. RDF语法：Turtle and RDF/XML
- 4. 数据类型Datatypes
- **5. 多值关系 (n-ary relationships)**
- 6. 空节点 (Empty nodes)
- 7. 列表Lists

# 哪里有错误？

---

```
@prefix ex: <http://example.org/> .  
ex:Chutney ex:hasIngredient "1lb green mango",  
                           "1tsp. Cayenne pepper" .
```

结构性不好，不易查询

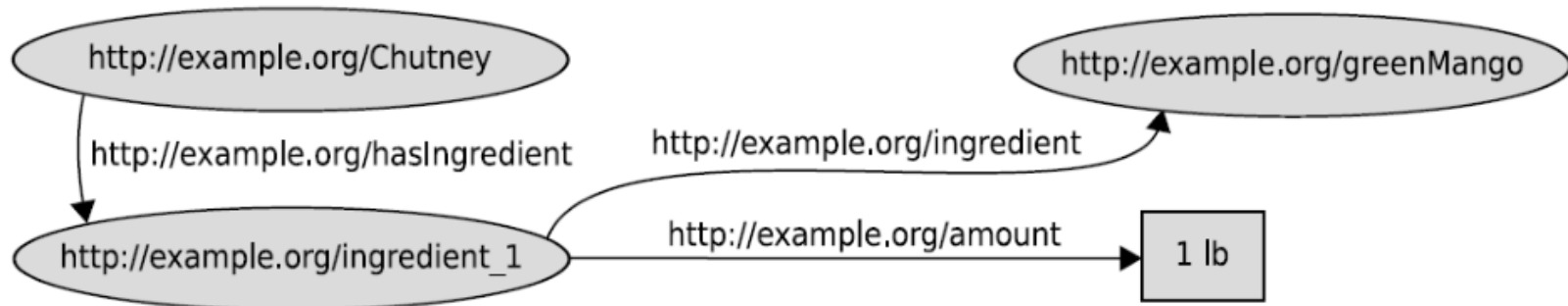
```
@prefix ex: <http://example.org/> .  
ex:Chutney ex:ingredient ex:greenMango;    ex:amount "1lb" ;  
           ex:ingredient ex:CayennePepper; ex:amount "1tsp." .
```

配料和用量分开了，三元组之间没有关系，引发歧义！

---

# 这是一个三元关系

---



```
@prefix ex: <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:Chutney          ex:hasIngredient  ex:ingredient1 .
ex:ingredient1      rdf:value         ex:greenMango;
                    ex:amount         "1lb" .
```

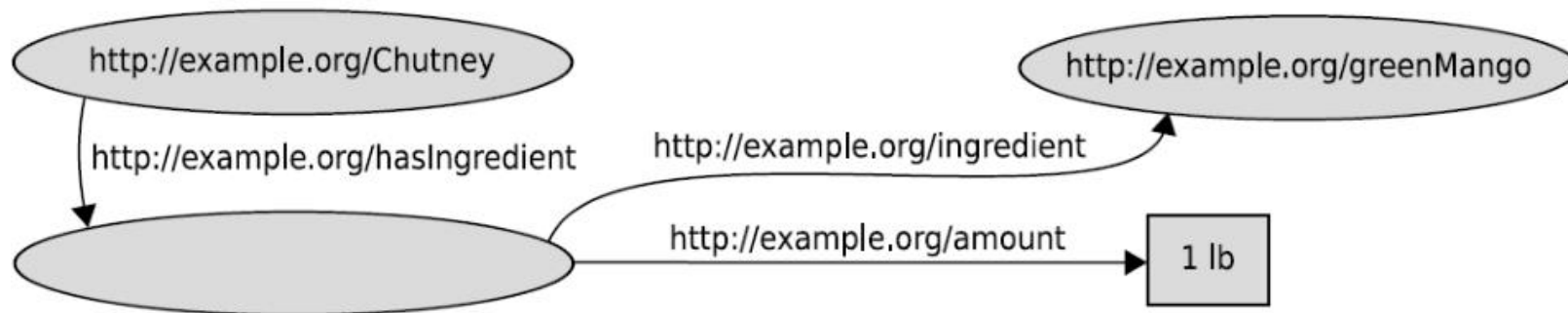
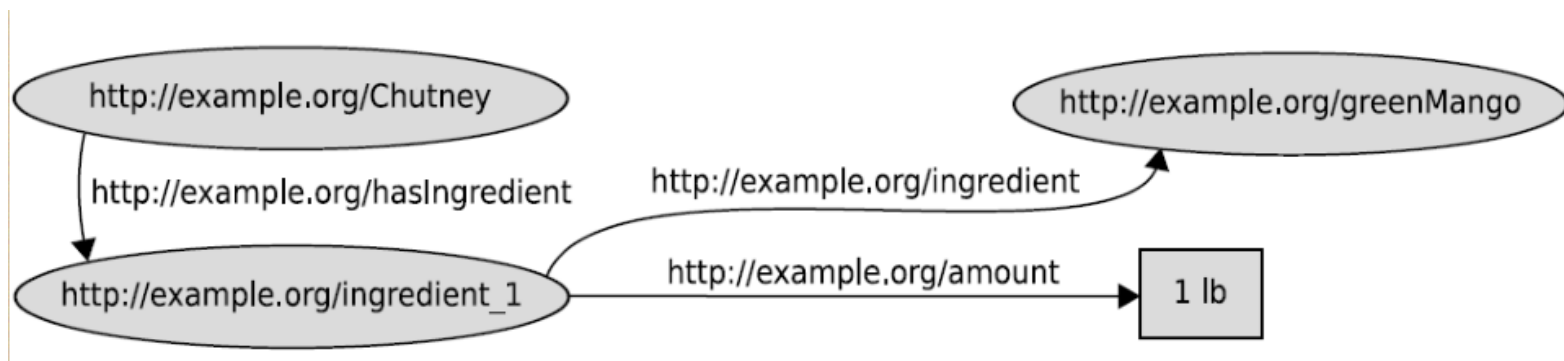
# RDF要点

---

- 1. 动机
- 2. 三元组和图
- 3. RDF语法：Turtle and RDF/XML
- 4. 数据类型Datatypes
- 5. 多值关系 (n-ary relationships)
- **6. 空节点 (Empty nodes)**
- 7. 列表Lists

# 空节点不需要名字

---



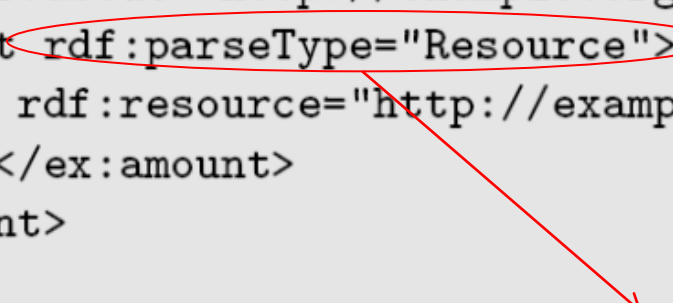
# 空节点语法

---

```
<rdf:Description rdf:about="http://example.org/Chutney">
  <ex:hasIngredient rdf:nodeID="id1" />
</rdf:Description>
<rdf:Description rdf:nodeID="id1">
  <ex:ingredient rdf:resource="http://example.org/greenMango" />
  <ex:amount>1lb</ex:amount>
</rdf:Description>
```

精简：

```
<rdf:Description rdf:about="http://example.org/Chutney">
  <ex:hasIngredient rdf:parseType="Resource">
    <ex:ingredient rdf:resource="http://example.org/greenMango" />
    <ex:amount>1lb</ex:amount>
  </ex:hasIngredient>
</rdf:Description>
```



表示属性值是一个空节点，将自动创建一个新的空白节点

# 空节点语法

---

```
<rdf:Description rdf:about="http://example.org/Chutney">
  <ex:hasIngredient rdf:nodeID="id1" />
</rdf:Description>
<rdf:Description rdf:nodeID="id1">
  <ex:ingredient rdf:resource="http://example.org/greenMango" />
  <ex:amount>1lb</ex:amount>
</rdf:Description>
```

## Turtle:

```
@prefix ex: <http://example.org/> .
ex:Chutney    ex:hasIngredient _:id1 .
_:id1         ex:ingredient ex:greenMango;    ex:amount  "1lb" .
```

空白节点由一个下划线代替命名空间前缀。

## 精简:

```
@prefix ex: <http://example.org/> .
ex:Chutney    ex:hasIngredient
               [ ex:ingredient ex:greenMango;    ex:amount  "1lb" ] .
```



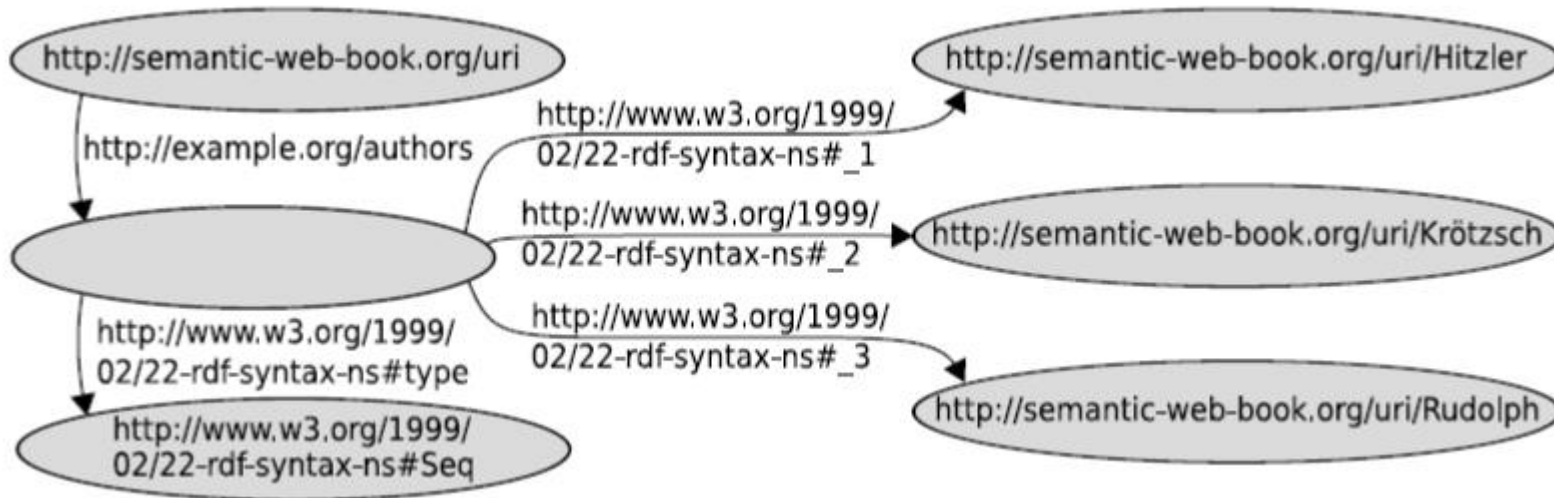
# RDF要点

---

- 1. 动机
- 2. 三元组和图
- 3. RDF语法：Turtle and RDF/XML
- 4. 数据类型Datatypes
- 5. 多值关系 (n-ary relationships)
- 6. 空节点 (Empty nodes)
- **7. 列表Lists**

# Open Lists(containers)

---



```
<rdf:Description rdf:about="http://semantic-web-book/uri">
  <ex:authors>
    <rdf:Seq>
      <rdf:li rdf:resource="http://semantic-web-book.org/uri/Hitzler" />
      <rdf:li rdf:resource="http://semantic-web-book.org/uri/Kröttsch" />
      <rdf:li rdf:resource="http://semantic-web-book.org/uri/Rudolph" />
    </rdf:Seq>
  </ex:authors>
</rdf:Description>
```

# Open Lists(containers , 容器)

---

- “open” : 可添加新元素.
- rdf:Seq – 顺序列表
- rdf:Bag – 无序
- rdf:Alt – 为替代 , 是一组可选择的资源或文字集合
- Lists实际难以体现形式化语义

# Closed lists (collections)

---

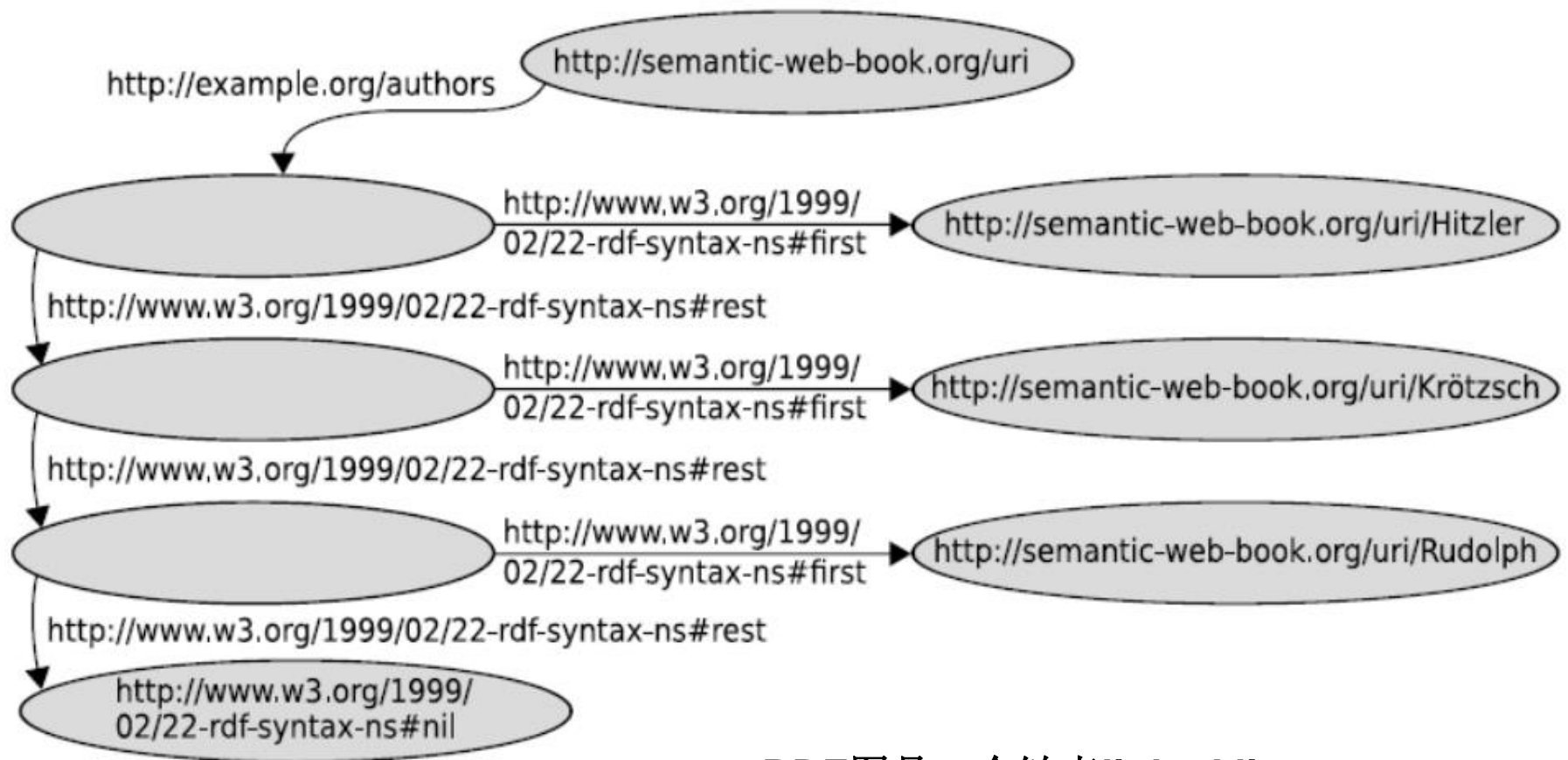
```
<rdf:Description rdf:about="http://semantic-web-book/uri">
  <ex:authors rdf:parseType="Collection">
    <rdf:Description
      rdf:about="http://semantic-web-book.org/uri/Hitzler" />
    <rdf:Description
      rdf:about="http://semantic-web-book.org/uri/Krötzsch" />
    <rdf:Description
      rdf:about="http://semantic-web-book.org/uri/Rudolph" />
  </ex:authors>
</rdf:Description>
```

```
@prefix book: <http://semantic-web-book.org/> .
book:uri <http://example.org/authors>
  ( book:uri/Hitzler book:uri/Krötzsch book:uri/Rudolph ) .
```

---

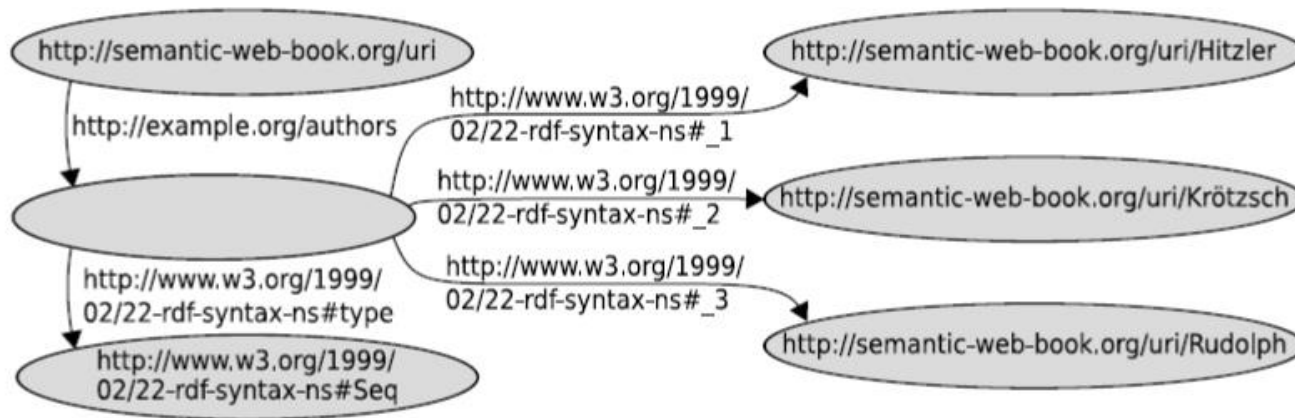
# Closed lists (collections)

---



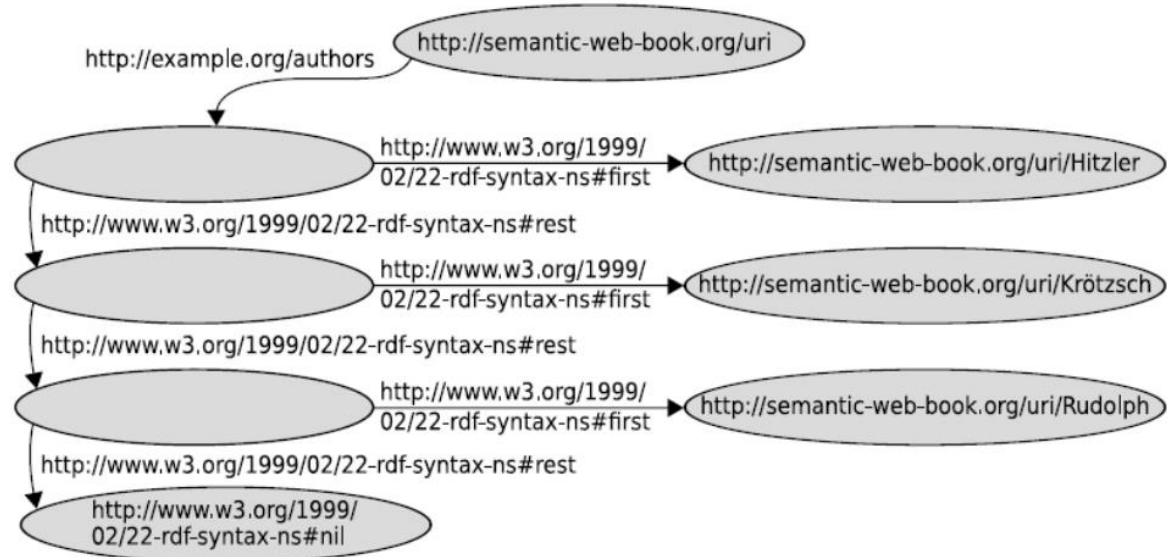
**RDF图是一个链表linked list**

# 对比



开放的

封闭的



# 小结

---

- 什么是语义网，从WEB到语义网
  - 什么是本体，本体的作用
  - 资源描述框架RDF，TURTLE语法与XML-based语法
- 
- 实验：掌握RDF框架语言的基本语法
-