

Advanced web technology

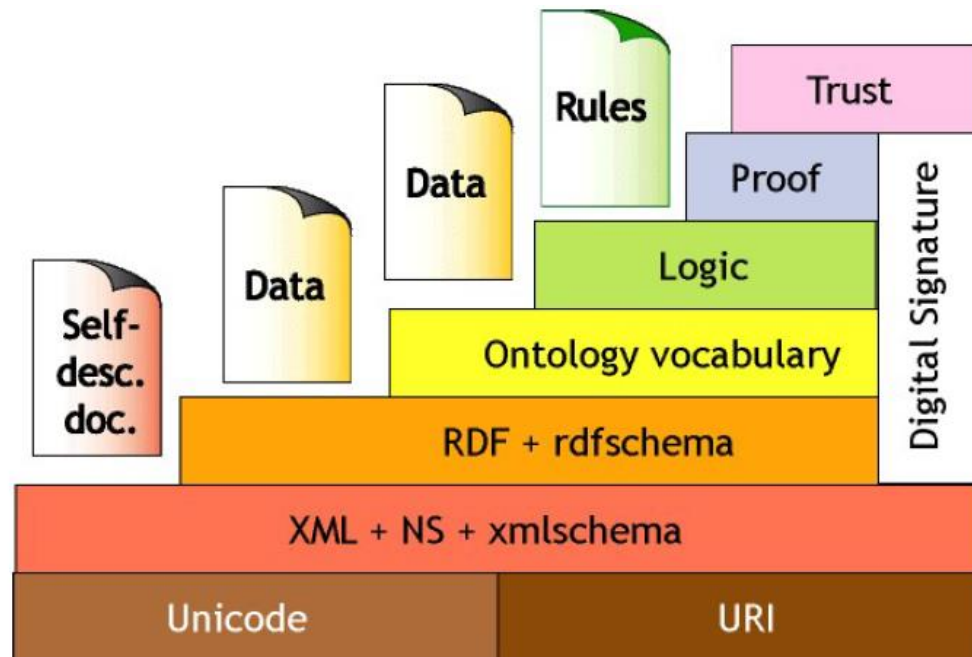
高级Web技术

语义Web概述

语义Web 简介

语义Web协议栈

Wikipedia 定义**语义网**："a project that intends to create a universal medium for information exchange by putting documents with computer-processable meaning (semantics) on the World Wide Web"。



语义Web 简介

▪ 语义Web协议栈

– **URI**(Uniform Resource Identifiers)

- 作为资源(任何东西都可以看作资源)标识机制，提供对资源的标准化的名字描述。

– **Unicode**

- 提供世界上各种语言的统一的字符编码标准。

– **XML**(Extensible Markup Language)

- 定义了结构化的数据描述方式，是数据互操作的语法基础，但没有包含任何特殊的语义。

– **Namespace**

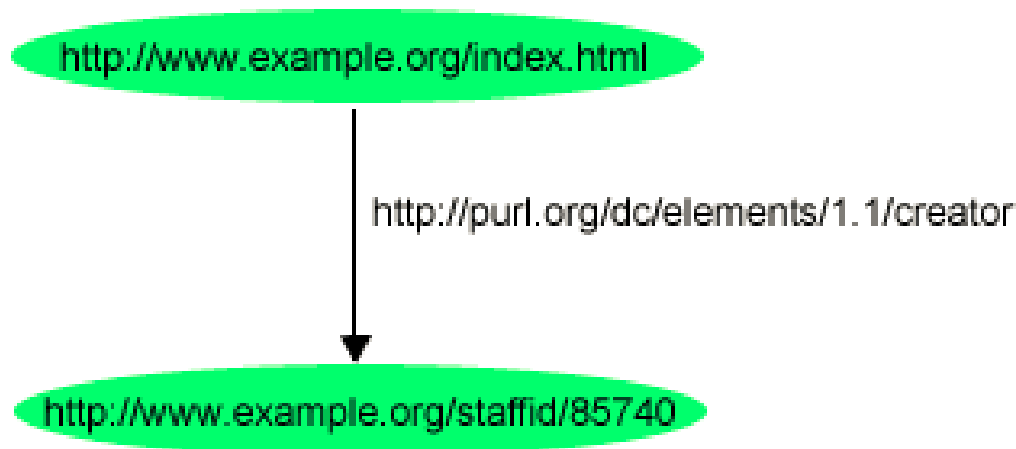
- 提供将名字分类的机制，使得重名但含义不同的资源能够一起使用。

语义Web 简介

■ 语义Web协议栈

– RDF, RDF Schema

- RDF(Resource Description Framework)是描述数据语义的基础。它定义了描述资源以及陈述事实的基本方式：主语、谓语、宾语的三元组。



- RDF Schema是一种RDF词汇描述语言，在RDF之上定义了一个最小的语义模型(词汇集)支持复杂词汇的建模，包括Resource, Class, Property, subClassOf, subPropertyOf, range, domain等基本元素。

语义Web 简介

- 语义Web协议栈

- **Ontology Vocabulary**

- Ontology提供了一种明确的定义语义的方式，通过Ontology的定义的语义，使机器能够进行互操作，使机器能够理解数据的语义。

- **Logic Layer**

- 逻辑层在本体所描述的知识之上提供逻辑推理能力(基于规则)。

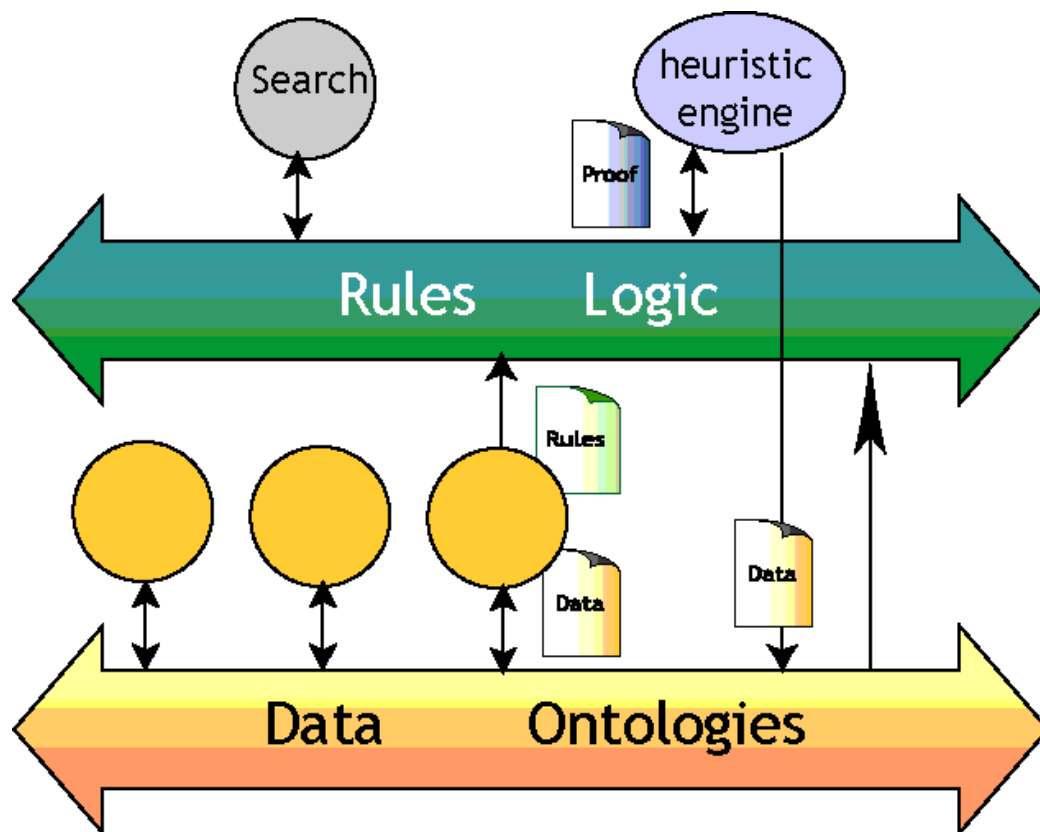
- **Proof Layer**

- 有了对事实的逻辑描述，就能够提供对事实的复杂的“证明”。

语义Web 简介

■ 语义Web协议栈

■ 语义Web总线(Semantic Web Bus)



语义Web 简介

■ 语义Web协议栈

– 可信层

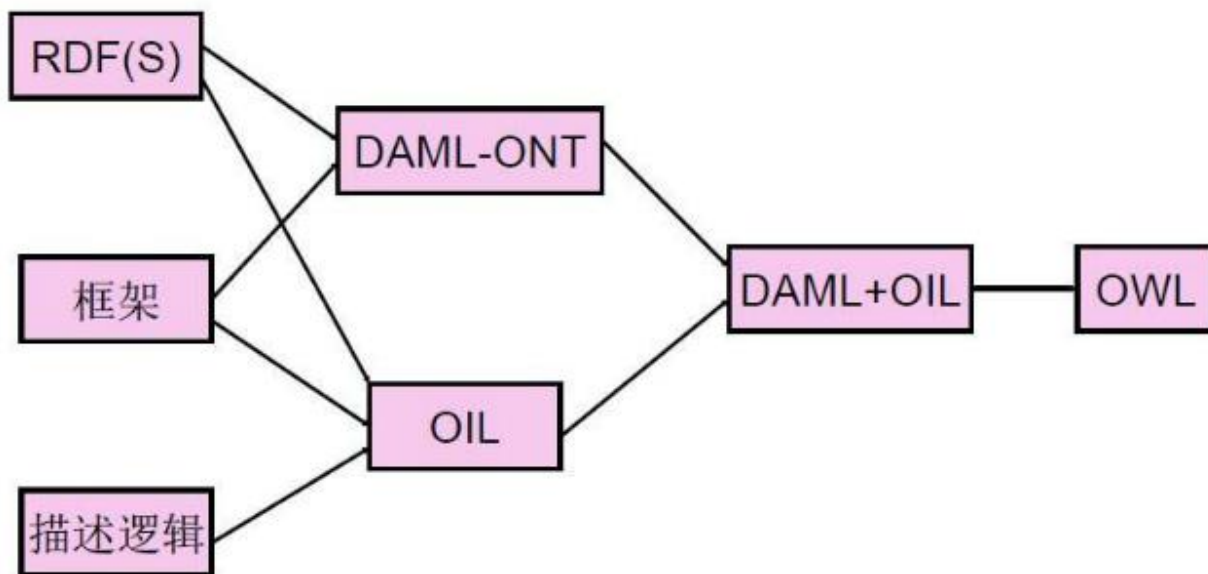
- 在之前所有层次的基础之上，会形成无数对于某一事实的陈述，这些陈述合理与否依赖于它们所处的上下文环境。
- 因此当人或计算机访问这些陈述时，需要根据上下文和自己的需求自行判定该陈述是否可信（Trust）。
- 采用加密技术和数字签名技术（渗透到每个层次的规范中去）是实现判定可信性的一个重要手段，并可以由此形成一个可信的Web。

，John声明：所有一级代理商有权委托二级代理商，而Jane是我公司的一级代理商之一。Jane的声明是：所有与我公司有三年以上合作关系的，将获得二级代理商资格，Joe是我公司一个有4年合作关系老伙伴。同时，所有这些声明都有数字签名保证了是可信的，于是，可以得出结论：Joe将获得John公司的二级代理商资格

语义Web

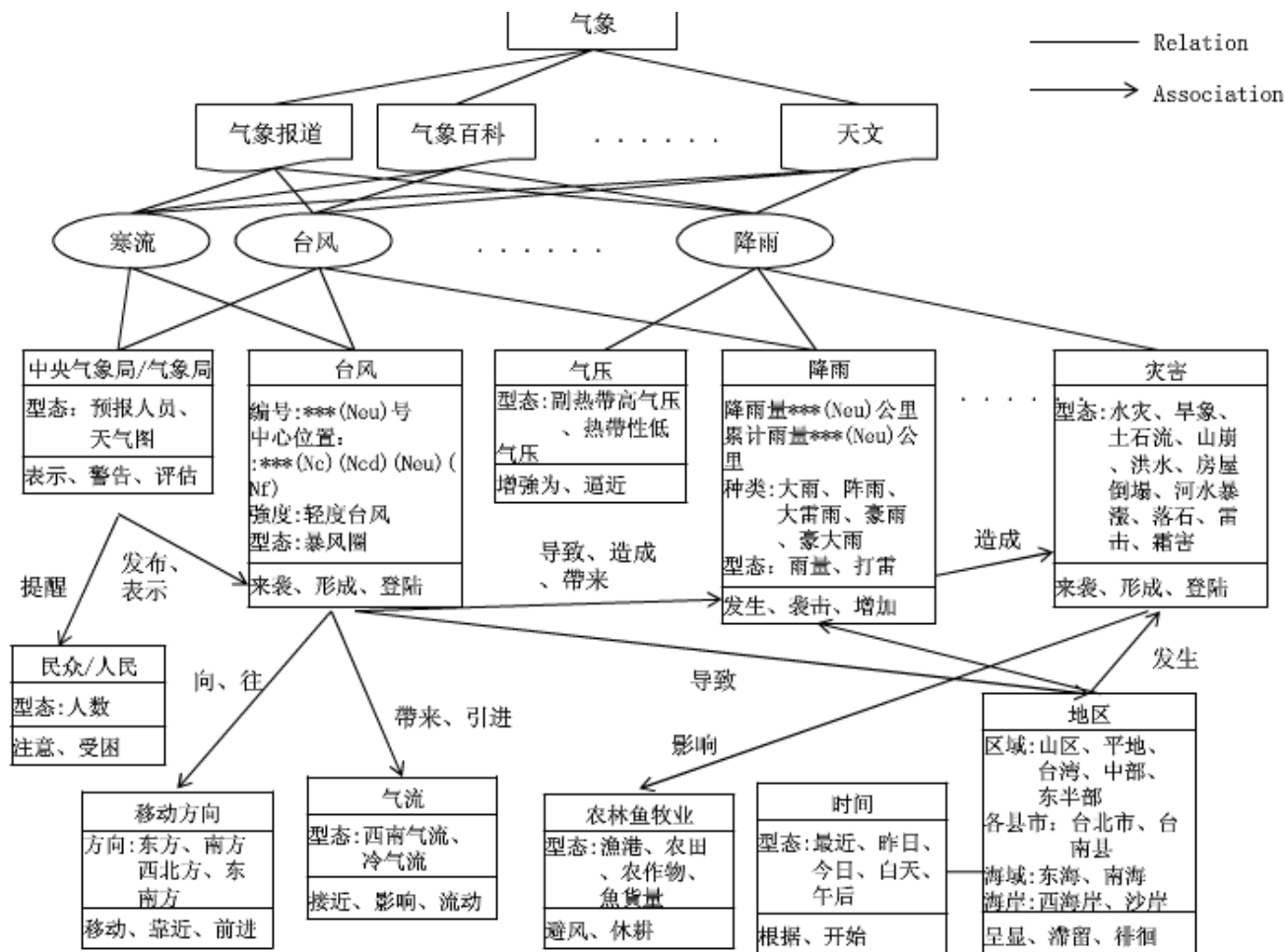
■ Ontology表示模型和语言关系图

Ontology is a formal, explicit specification of a shared conceptualization.



语义Web

■ Ontology实例



语义Web

■ W3C的Ontology语言栈

名称	描述
XML	结构化文档的表层语法，对文档没有任何语义约束。
XML Schema	定义XML文档的结构约束的语言。
RDF	对象（或者资源）以及它们之间关系的数据模型，为数据模型提供了简单的语义，这些数据模型能够用XML语法进行表达。
RDF Schema	描述RDF资源的属性和类型的词汇表，提供了对这些属性和类型的普遍层次的语义。
OWL	添加了更多的用于描述属性和类型的词汇，例如类型之间的不相交性（disjointness），基数（cardinality），等价性，属性的更丰富的类型，属性特征（例如对称性，symmetry），以及枚举类型（enumerated classes）。

RDF

■ RDF简介

- 资源描述框架(Resource Description Framework, 简称 RDF)是一个用于表达关于万维网(World Wide Web)上的资源的信息的语言。
- 将“Web资源 (Web resource)”这一概念一般化后, RDF可被用于表达关于任何可在Web上被标识的事物的信息, 即使有时它们不能被直接从Web上获取。
- RDF基于这样的思想: 用Web标识符 (URIs) 来标识事物, 用简单的属性 (property) 及属性值来描述资源。
- RDF用于信息需要被应用程序处理而不是仅仅显示给人观看的场合。RDF提供了一种用于表达这一信息、并使其能在应用程序间交换而不丧失语义的通用框架。

RDF

▪ RDF

– 元数据

- 是关于数据的数据或关于信息的信息
- 元数据的使用，可以大大提高系统的检索和管理的效率。
- RDF 是W3C提出的用于描述Web资源的元数据解决方案

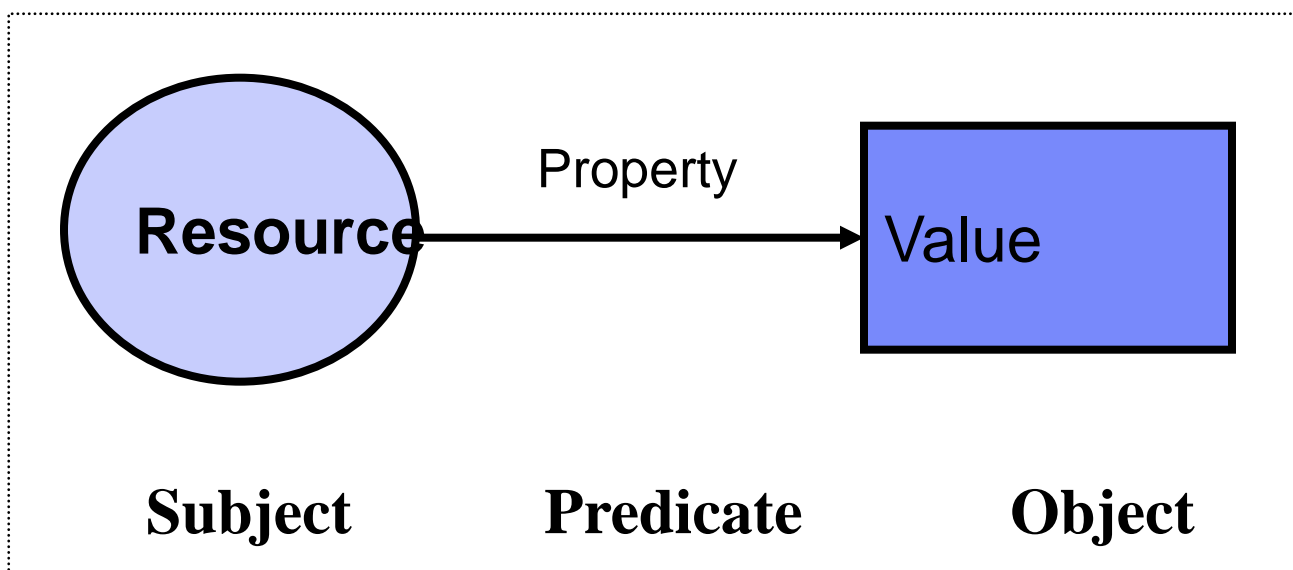
– 资源描述框架（Resource Description Framework）

- **资源（Resource）**：所有在Web上被命名、具有**URI**的东西。如网页、XML文档中的元素等；
- **描述（Description）**：对资源属性（Property）的一个陈述（Statement），以表明资源的特性或者资源之间的联系；
- **框架（Framework）**：与被描述资源无关的通用模型，以包容和管理资源的多样性、不一致性和重复性。
- 综合起来，RDF就是定义了一种通用的框架，即资源—属性—值的三元组，来描述Web上的各种资源。

RDF

▪ RDF基本模型

每个RDF陈述可以看作一个逻辑断言 (Assertion)。

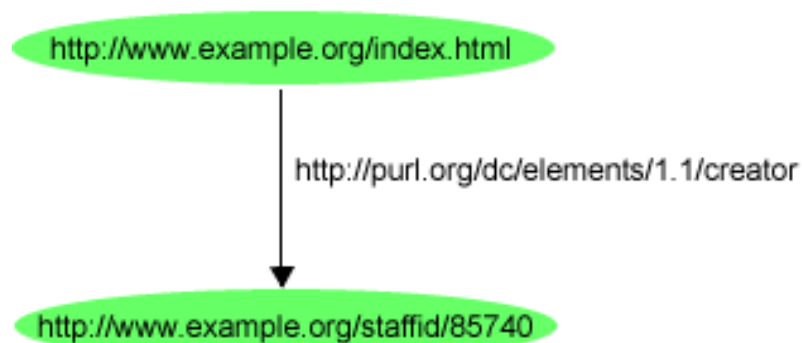


RDF表示了一个称述

RDF

■ RDF

- 每个陈述都是由主体（subject），谓词（predicate），客体（object）组成的
 - 用URIref标识RDF陈述中涉及的事物
 - 用RDF/XML作为一种机器可处理的方式来表示RDF陈述
- **RDF的图模型**
 - 一个表示主体的节点；
 - 一个表示客体的节点；
 - 一个由主体节点指向客体节点的表示谓词的弧；



RDF

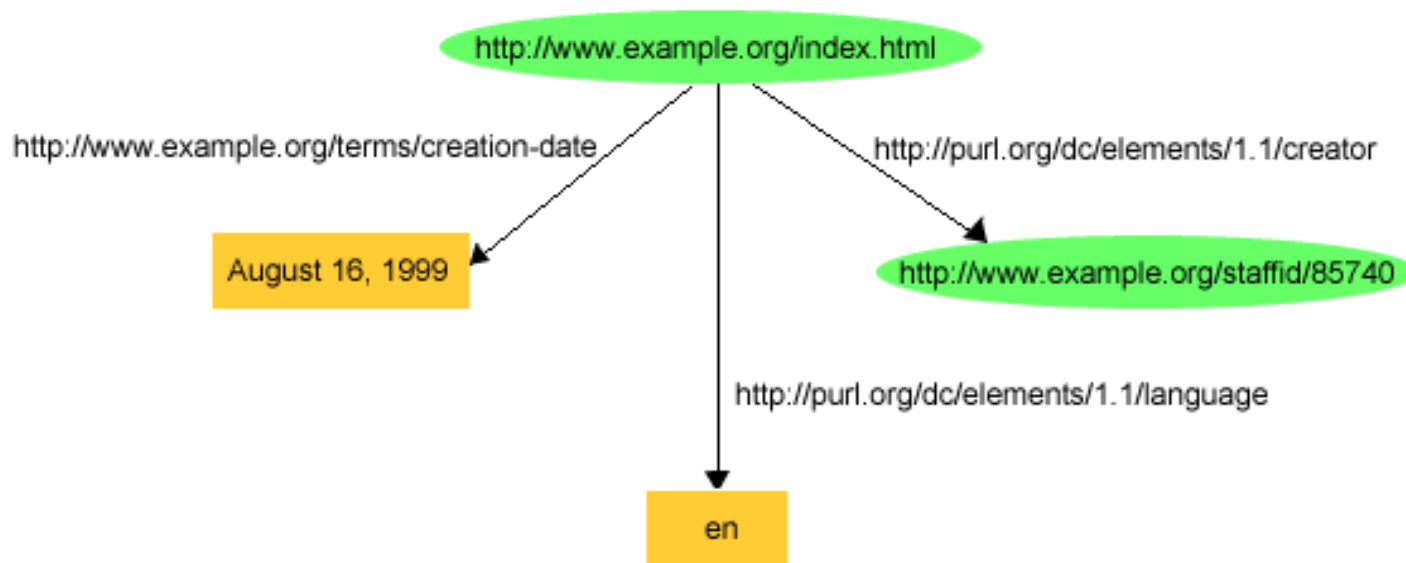
▪ RDF实例

有一个人由<http://www.w3.org/People/EM/contact#me> 标识, 他的名字是Eric Miller, 他的电子邮件地址是em@w3.org, 他的头衔是Dr.



RDF

图与三元组



`<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator>`
`<http://www.example.org/staffid/85740> .`
`<http://www.example.org/index.html> <http://www.example.org/terms/creation-date> "August 16, 1999" .`
`<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/language> "en" .`

RDF

■ 表示RDF的XML语法: RDF/XML

- **rdf:type**属性来描述一个RDF资源时, 该属性的值就是被看作表达这种事物的种类 (**kings**) 或类别 (**categories**) 的资源
- **rdf:about**属性代表被描述的资源

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:exterms="http://www.example.org/terms/">
    <rdf:Description rdf:about="http://www.example.org/index.html">
      <exterms:creation-date>August 16, 1999</exterms:creation-date>
      <dc:language>en</dc:language>
      <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
    </rdf:Description>
  </rdf:RDF>
```

RDF

■ 其他RDF表达能力

– RDF容器

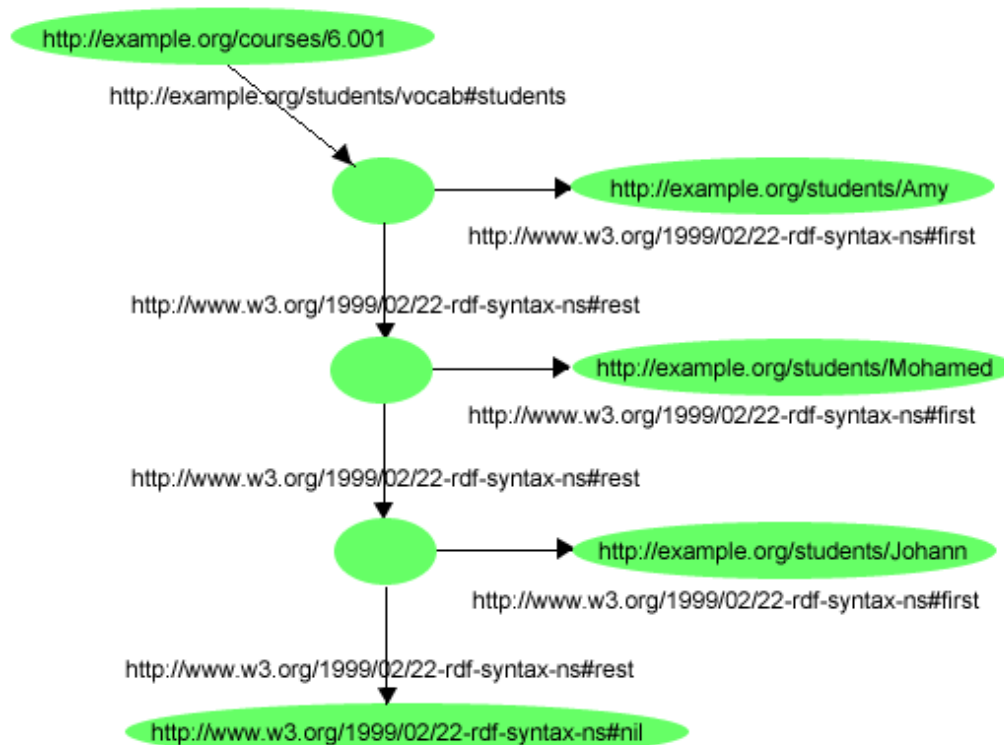
- **rdf:Bag**
可能包含重复成员的资源或文字，且成员之间是无序的
- **rdf:Seq**
可能有重复的成员，而且成员之间是有序的
- **rdf:Alt**
表示了一组可以选择的资源或文字

RDF

■ 其他RDF表达能力

– RDF集合

- RDF的集合词汇包括属性`rdf:first` 和`rdf:rest`, 和资源`rdf:nil`



RDF

■ RDF

– 词汇集

■ 允许

■ 流行

– 以

– 定

– 定

■ 使用

```
<rdf :RDF
```

```
  xmlns :rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://www.purl.org/DC/"
  xmlns:nm="http://www.metalab.unc.edu/xml/names/">
```

```
<rdf :Description about="http://www.metalab.unc.edu/xml" >
```

```
<dc:CREATOR parsetype="Literal">
```

```
<nm:FirstName> Elliotte</nm:FirstName>
```

```
<nm:MiddleNmae> Rusty</nm:MiddleName>
```

```
<nm:LastName> Harold</nm:LastName>
```

```
</dc:CREATOR>
```

```
</rdf :Description>
```

```
</rdf :RDF>
```

RDF

■ RDF Schema

- RDF Schema为RDF提供了一个类型系统。RDF Schema 类型系统在某些方面类似于Java这样的面向对象编程语言的类型系统
- RDFS通常把类组织成为一种分级结构（类的分层）
- RDF Schema 所具有的这些能力本身也是以 RDF词汇形式提供的,这些资源的（RDF Schema词汇）URI带有前缀 <http://www.w3.org/2000/01/rdf-schema#> （QName通常采用前缀 rdfs:）

RDF

■ RDF Schema

– 描述类

- RDF类可以用来表示事物的任何分类
- 类可以通过RDF Schema中的资源（`rdfs:Class`和`rdfs:Resource`）以及特性（`rdf:type`和`rdfs:subClassOf`）来表示
- 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

```
ex:MotorVehicle    rdf:type    rdfs:Class
```

– 描述实例

- 特性`rdf:type`用来表明一个资源是某个类的实例

```
exthings:companyCar    rdf:type    ex:MotorVehicle
```

RDF

■ RDF Schema

– 描述类之间的关系

- 两个类之间的特化关系（specialization relationship）可以用预定义的特性 `rdfs:subClassOf` 来描述。`rdfs:subClassOf` 特性具有传递性

```
ex:Van    rdfs:subClassOf    ex:MotorVehicle
```

- 一个类可以是一个或多个类的子类
- RDF Schema 规定：所有的类总是 `rdfs:Resource` 的子类（因为任何类的实例都是资源）

RDF

▪ RDF Schema

– 描述特性

- 在RDF schema中，特性是用RDF类`rdf:Property`以及RDF Schema特性`rdfs:domain`（定义域）、`rdfs:range`（值域）以及`rdfs:subPropertyOf`来描述的
- RDF中的所有特性都被描述为类`rdf:Property`的实例

```
ex:terms:weightInKg    rdf:type    rdf:Property
```

- `rdfs:subPropertyOf`描述特性之间的特化关系

ex:primaryDriver是ex:driver的特化



ex:driver	rdf:type	rdf:Property
ex:primaryDriver	rdf:type	rdf:Property
ex:primaryDriver	rdfs:subPropertyOf	ex:driver

RDF

▪ RDF Schema

– 描述特性

- `rdfs:range` 用于表明某个特性的值（定义域）是给定类的实例

特性 `ex:author` 的值是类 `ex:person` 的实例



<code>ex:Person</code>	<code>rdf:type</code>	<code>rdfs:Class</code>
<code>ex:author</code>	<code>rdf:type</code>	<code>rdf:Property</code>
<code>ex:author</code>	<code>rdfs:range</code>	<code>ex:Person</code>

- `rdfs:range` 特性也可用于表明特性的值是一个类型文字

特性 `ex:age` 的值来自 XML Schema 数据类型 `xsd:integer`



<code>ex:age</code>	<code>rdf:type</code>	<code>rdf:Property</code>
<code>ex:age</code>	<code>rdfs:range</code>	<code>xsd:integer</code>

RDF

■ RDF Schema

- **rdfs:domain**用于表明某个特性应用于指定的类（定义域）

特性`ex:author`应用于类`ex:Book`的实例上

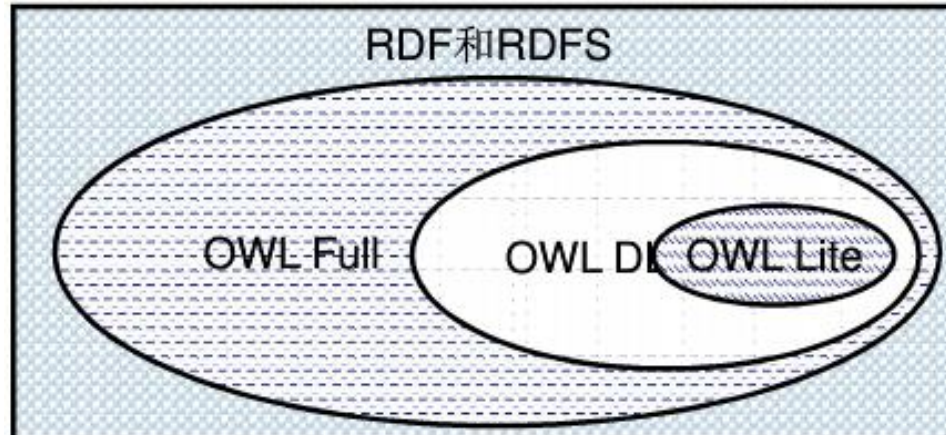


<code>ex:Book</code>	<code>rdf:type</code>	<code>rdfs:Class</code>
<code>ex:author</code>	<code>rdf:type</code>	<code>rdf:Property</code>
<code>ex:author</code>	<code>rdfs:domain</code>	<code>ex:Book</code>

语义Web

■ OWL

- OWL(Web Ontology Language)是W3C提出的一种本体描述语言，OWL源自于DAML+OIL。
- OWL DL得名于它的逻辑基础——描述逻辑，兼顾表达能力和可计算性



语义Web

■ OWL

子语言	描述	例子
OWL Lite	用于提供给那些只需要一个分类层次和简单的属性约束的用户。	支持基数 (cardinality), 只允许基数为0或1。
OWL DL	支持那些需要在推理系统上进行最大程度表达的用户, 这里的推理系统能够保证计算完全性 (computational completeness, 即所有地结论都能够保证被计算出来) 和可决定性 (decidability, 即所有的计算都在有限的时间内完成)。它包括了OWL语言的所有约束, 但是可以被仅仅置于特定的约束下。	当一个类可以是多个类的一个子类时, 它被约束不能是另外一个类的实例。
OWL Full	支持那些需要在没有计算保证的语法自由的RDF上进行最大程度表达的用户。它允许在一个Ontology在预定义的 (RDF、OWL) 词汇表上增加词汇, 从而任何推理软件均不能支持OWL FULL的所有feature。	一个类可以被同时表达为许多个体的一个集合以及这个集合中的一个个体。

从语法上来说, OWL-Lite是三个之中最简单的一个

OWL-Full是OWL的三种子语言中表达能力最强的一个, 但不太关心可判定性。

不过也正是由于表达能力太强这个原因, 用OWL-Full表示的本体是不能进行自动推理的。

语义Web

■ OWL语法简介

- **Classes** : 提供了组织具有相似特征的资源的一种抽象方式。
- 可以使用`rdfs:subClassOf`将Classes组成一个层次化的结构。

```
<owl:Class rdf:ID="Grape"> ... </owl:Class>
```

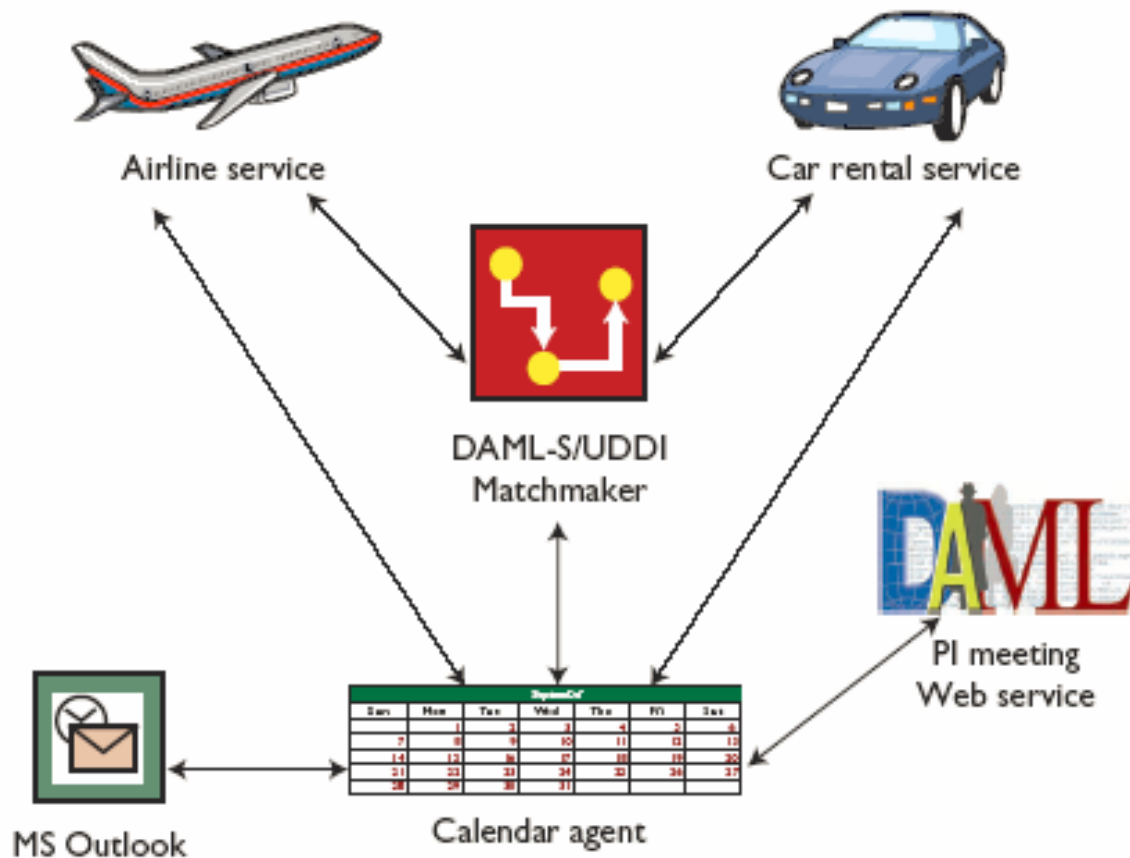
```
<owl:Class rdf:ID="WineGrape">
```

```
    <rdfs:subClassOf rdf:resource="&food;Grape" />
```

```
<owl:Class>
```

语义Web

■ 语义web服务



语义Web服务的一个应用场景