北京郵電大學

分布计算环境

北京邮电大学计算机学院

主要内容

- Web技术的发展历程
- Web基本支撑技术
- XML技术
- 语义WEB
- Web 2.0
- Web Service
- 小结

HTML的缺点

♦ 在数据组织和表达能力上：

- 专为描述主页的表现形式而设计，不能适应日益增多的信息检索要求和存档要求
- 不支持结构化数据：不支持对信息语义及其内部结构的描述
- 不支持合法检查：HTML不支持应用程序检查输入数据的结构合法性
- 对形式的描述能力不够：无法描述矢量图形、科技符号和一些其它的特殊显示效果
- 缺乏可扩展性：HTML不允许用户定义私有的标记字或属性，这就无法把数据参数化，也无法从语义上进行数据检查
- 标记日益臃肿，文件结构混乱而缺乏条理，导致浏览器的设计越来越复杂

对文档组织的新要求

- 扩展能力，可以根据需要定义新标记
- 结构的表达能力，可以表示任意复杂程度的数据
- 校验，可以检查数据的结构正确性
- 媒体无关性，可以用多种格式发布内容
- 厂商和平台中立，可以用标准的商业软件甚至简单的文本工具处理任何满足标准的文档
- 安全性

The Answer ➔ XML

XML主要内容

- 基本概念
- XML 文档结构定义
- XML 名字空间
- XML的解析
- XML的标准体系
- XML技术小结

XML基本概念

XML, the eXtensible Markup Language

开发者一致推崇使用
独立于任何体系结构的数据格式
独立于任何语言的数据格式
已经被广泛使用

XML基本概念

- **XML: 可扩展标记语言**
  - 由万维网联合组织（W3C：World Wide Web Consortium）制定的描述语言标准
  - 用来对信息进行自描述
  - 关注于怎样描述数据内容的组织和结构，以便数据在网络上进行交流和处理
- **XML最重要的特征：**
  - 被标记的各个数据是保持其含义的，因此系统间交换数据的可能性极大提高
- **XML最基本的概念：**
  - 结构化标记数据，是实现 **“文档结构化”** 的语言规范

XML的结构化文档技术

- **例: HTML格式文档**
- **<HTML>**
- **<TITLE>联想电脑 </TITLE>**
- **<BODY>**
- **<UL>**
- **<LI> 联想**
- **<LI>台式机**
- **<LI>P4**
- **<LI>2.4GHz**
- **<LI>1 GB**
- **<LI>120GB**
- **<LI>9999元**
- **</UL>**
- **</BODY>**
- **</HTML>**

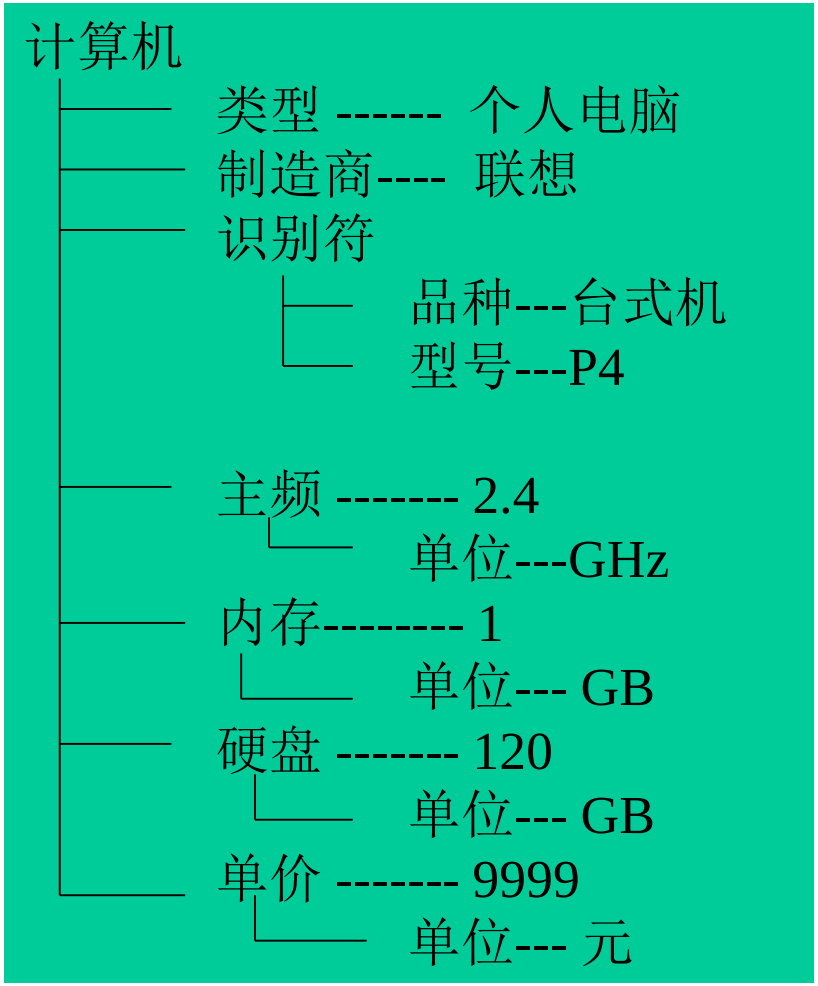
对应的XML格式文档：

<?xml version =”1.0”?>
<!DOCTYPE联想电脑>
<计算机 类型＝“个人电脑”>
<制造商>联想</制造商>
<识别符>
<品种>台式机</品种>
<型号>P4</型号>
</识别符>
<主频 UNIT＝“GHz”> 2.4 </主频>
<内存 UNIT＝“GB”> 1 </内存>
<硬盘 UNIT＝“GB”> 120 </硬盘>
<单价 UNIT＝“元”> 9999 </单价>
</计算机>

XML的结构化文档技术

- XML用标记表达文档结构
- XML标记：
  - 正确地传达按文档所标记的数据的“含义”：通过标记的含义
  - 正确地表达XML文档所具有的“结构”：通过标记间的嵌套
- XML文档数据采用“树结构”方式表达

例： 上例中XML文档的树结构形式



XML文档的有关术语

- **XML文档：**就是用XML标记写的XML源代码文件。XML文档是类似ASCII的纯文本文件，可用任何文本编辑器创建和修改
- XML文档包含三个部分：

  1. XML文档声明

  声明表示它是一个XML文档，它遵循的是哪个XML版本的规范。

  <?xml version="1.0“ standalone="no" encoding="UTF-8"?>

- 2. 关于文档类型和样式表的定义

<!DOCTYPE type-of-doc SYSTEM/PUBLIC "dtd-name">

<?xml-stylesheet type=“text/css” href=“jokes.css”?>

- 3. 用XML标记创建的内容
  - 文档内容主体部分

XML文档的有关术语

- Element(元素)：

  类似于HTML，XML中的元素由标记来定义，包括开始和结束标记以及其中的内容

  - <author>巴金</author>

- Tag(标记)

  标记是用来定义元素的。在XML中，标记必须成对出现，将数据包围在中间

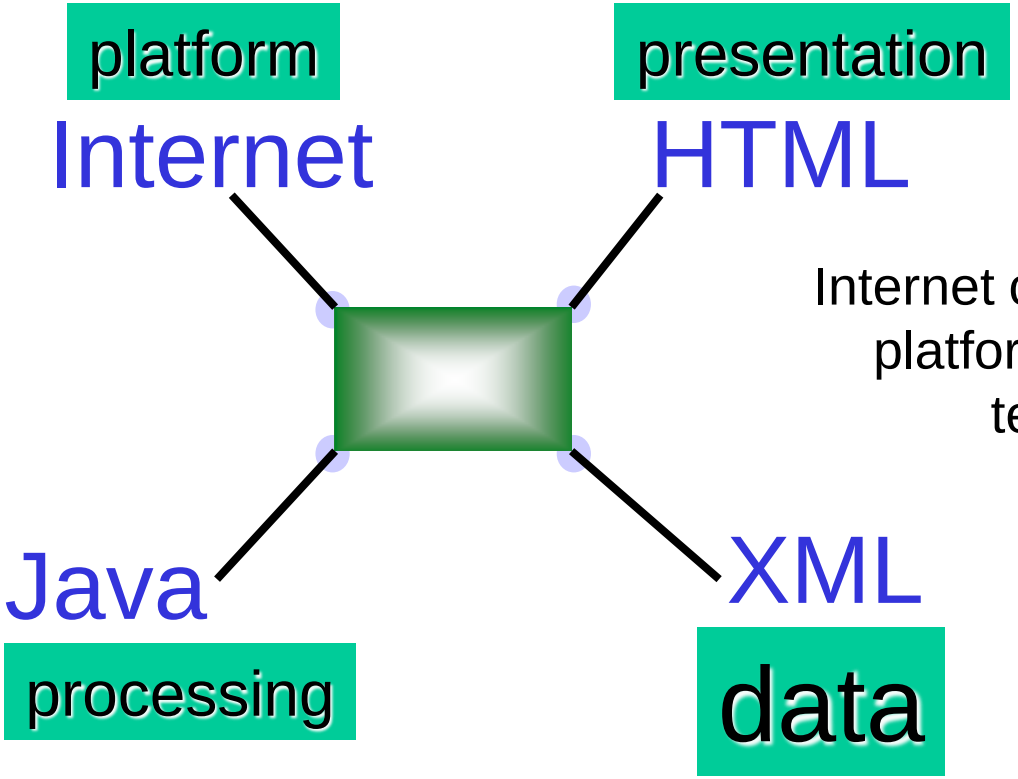
  - <author>

- Attribute(属性):

  与HTML中的属性类似， XML中的属性是对标记进一步的描述和说明，一个标记可以有多个属性。每个属性都有它自己的名字和数值，属性是标记的一部分

  - <author sex=“male” age=“101”>巴金</author>

Where Does XML fit in ?



Internet creates a need for platform-independent technology.

- HTML = Universal Rendering
- Java = Universal Code
- XML = Universal Data

XML数据与处理方式的分离

- XML强调数据与数据处理方式的分离
- XML文件的编写者集中精力于数据本身，而不受数据实际处理方式的细枝末节的影响
- XML文档和DTD以及Schema文件中并没有标记关于数据的显示方式或者处理方式的信息，XML文档的处理方式可根据用户要求而改变
  - 例如，可以为相同的数据定义不同的显示方式，从而适合于不同应用、不同媒体，使XML数据得到最大程度的重用性
- 对数据的处理方式由处理XML文档的程序负责
  - 已标准化的：XHTML
  - 未标准化的

XML主要内容

- 基本概念
- XML 文档结构定义
- XML 名字空间
- XML的解析
- XML的标准体系
- XML技术小结

文档结构定义DTD

- **DTD（Document Type Definition)：对一种数据文件中数据的组织存放结构的说明**
- 描述了一个标记语言的**语法和词汇表**，也就是定义了文件的整体结构以及文件的语法，规定了一个语法分析器为了解释一个XML文件所需要知道的所有规则的细节
  - 严格定义了某项数据应该在哪儿出现，规定了各种数据项之间的关系
  - 列出所有有效的元素，例如元素、标记、属性、实体
- 通过建立各种不同的文档类型定义，我们就拥有了不同格式的数据文件
- 不同格式的数据文件的DTD都是按照一定的标准给出的。这样即使遇到一种未知格式但带有DTD的数据文件，也可以通过分析它的DTD来知道数据在文件中的组织结构，进而提取数据

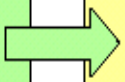
XML Schema

- DTD本身不是XML，只提供了非常有限的数据类型，也不支持名域机制；DTD中的内容模型是不开放的，不能随意扩充内容
- **XML Schema**
  - DTD的升级版
  - 使用XML格式来描述XML数据模型
  - 具有DTD所没有的强大特性：比如数据类型、子元素个数等
  - 业界标准的XML数据建模工具

XML Schema

- XML schema本身是规范的XML文档，支持的数据类型包括数字型、布尔型、整型、日期时间、 URI、十进制数等。而且它还支持由这些简单的类型生成更复杂的类型。

```
struct POINT {
  int x;
  int y;
};
```



```
<complexType name='POINT' >
  <sequence>
    <element name='x' type='int' />
    <element name='y' type='int' />
  </sequence>
</complexType>
```

XML主要内容

- 基本概念
- XML 文档结构定义
- XML 名字空间
- XML的解析
- XML的标准体系
- XML技术小结

XML Namespace

- 名域/命名空间： 命名空间用于区分具有相同名而又在相应的上下文中具有不同含义的元素和属性
- 当我们在一个XML文档中使用他人的或者多个DTD或Schema文件，因为XML中标记都是自己创建的，在不同的DTD或Schema文件中，标记名可能相同但表示的含义不同，这就可能引起数据混乱

  如在一个文档<table>wood table</table>中<table>表示桌子，而在另一个文档<table>namelist</table>中<table>表示表格。
- namespaces通过给标记名称加一个网址(URL)定位的方法来区别这些名称相同的标记

XML Namespace

- Namespaces在XML文档的开头部分声明，声明的语法为：
  <document xmlns:yourname='URL'>
  其中 yourname是自己选定的namespaces的名称，URL就是名字空间的网址
- 假设 "桌子<table>"文档来自http://www.zhuozi.com，就可声明为
  <document xmlns:zhuozi='http://www.zhuozi.com'>
- 然后在后面的标记中使用定义好的名字空间，就可以将这两个<table>区分开来
  <zhuozi:table>wood table</table>
- 需注意的是：设置URL并不是说这个标记真的要到那个网址去读取，仅仅作为一种区别的标志而已

XML主要内容

- 基本概念
- XML 文档结构定义
- XML 名字空间
- XML文档的解析
- XML的标准体系
- XML技术小结

XML文档解析

XML文档解析分析XML文档的内容是否符合XML标准，判断一个文档是否遵守DTD/Schema定义的结构，并对XML文档内容的访问提供支持

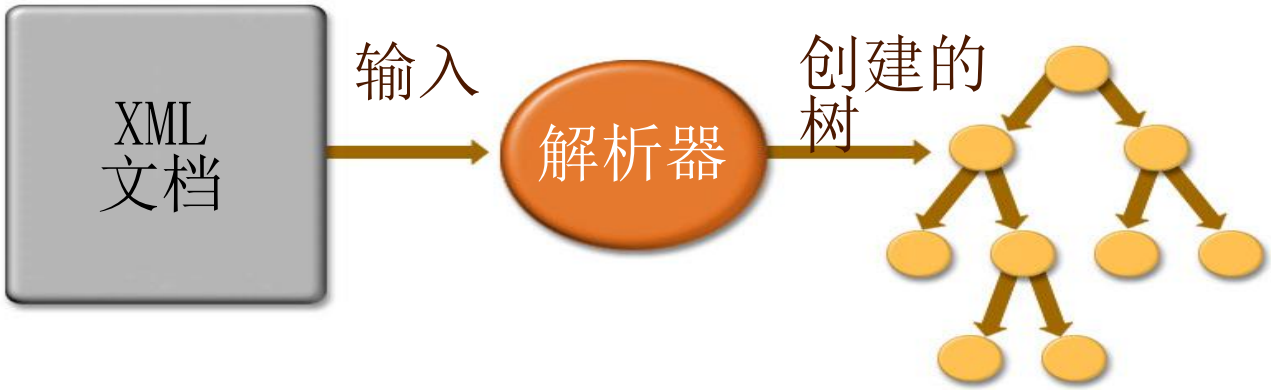
- **两种主要的解析方式**
  - DOM（Document Object Model）
    - ➔根据文档的内容建立一个层次的数据结构；可读可写可修改；W3C标准
  - SAX (Simple API for XML)
    - ➔基于事件的解析器；顺序读取,速度快,不可写入；业界的事实标准

XML APIs—DOM

DOM 是W3C为XML文档和HTML文档制定的一个独立于平台和语言的编程接口标准，使得程序和脚本能以标准的方式存取与更新文档的内容、结构和样式。

- The Document Object Model (DOM)
  - 基于文档的树型结构存取XML文档
  - 由元素节点和文本节点组成
  - 可以在树的某个节点上向前或向后移动
  - 与SAX相比需要更大的内存
  - org.w3c.dom.*

DOM 的行为



- DOM提供给用户一组接口来装载、操作并序列化XML文档
- DOM提供了对存储在内存中的XML文档的一个完全的表示，提供了可以随机访问整个文档的方法。
- API例：
  - doc = builder.parse("XXX.xml")
  - root = doc.getDoumentElment();
  - getChildNodes();getParentNode();removeChild();appendChild(); setNodeValue();
  - writer.write(root);

XML APIs — SAX

- The Simple API for XML (SAX)

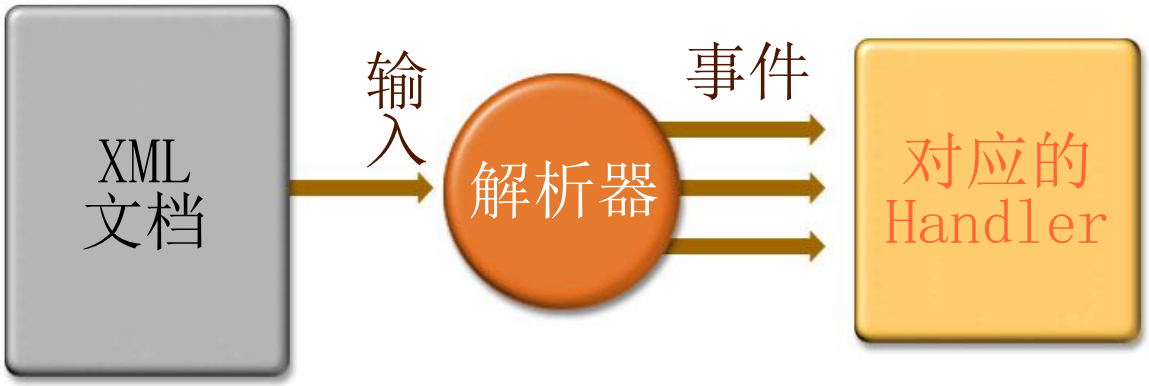
  SAX是一个接口集合，它允许开发人员在自己的应用中通过这个接口集合来读取和分析XML文档中的数据

  - 串行存取文档
  - 轻量级、快速
  - 是基于事件的，当它在一个XML文档中发现特殊的符号时，它会产生相关的事件
  - org.xml.sax.*

  SAX最初是为Java编程语言设计的，并使用了Java的接口定义

new Cheapest()

SAX的行为



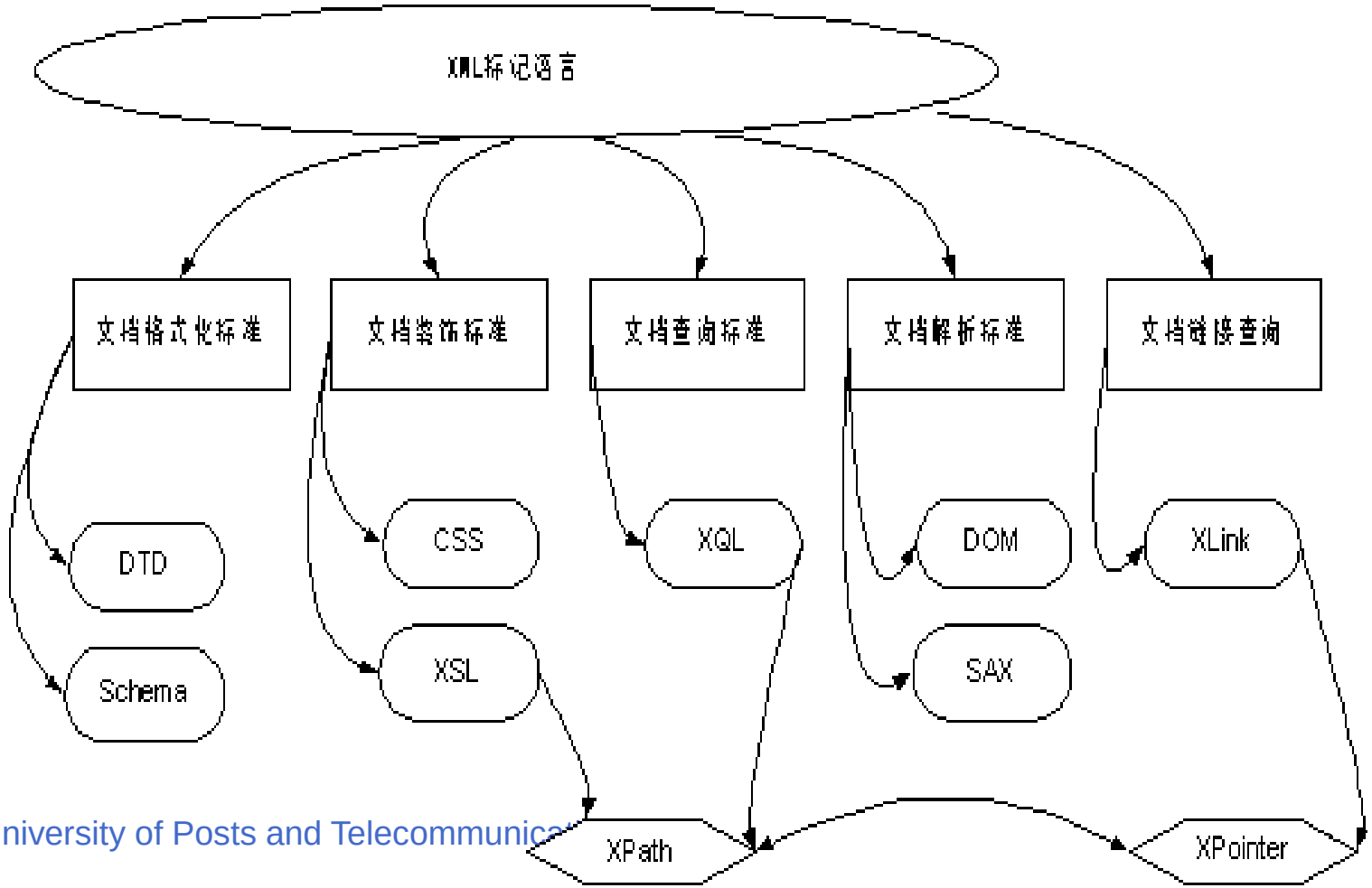
- SAX 是事件驱动的，在XML文档被解析的过程中，相应的事件被发送给设置的事件处理程序。事件如：
  - startDocument, startElement, endElement, characters, etc.
- 事件被送到设置的事件处理程序，由该程序决定怎么处理相应的事件
  - parser.setContentHandler(new Cheapest());

XML主要内容

- 基本概念
- XML 文档结构定义
- XML 名字空间
- XML文档的解析
- XML的标准体系
- XML技术小结

XML标准体系

- XML是由若干标准共同支撑起来的一个体系，这些标准组合起来分别完成数据集成、数据传输、数据操纵、数据显示等
- 基础标准

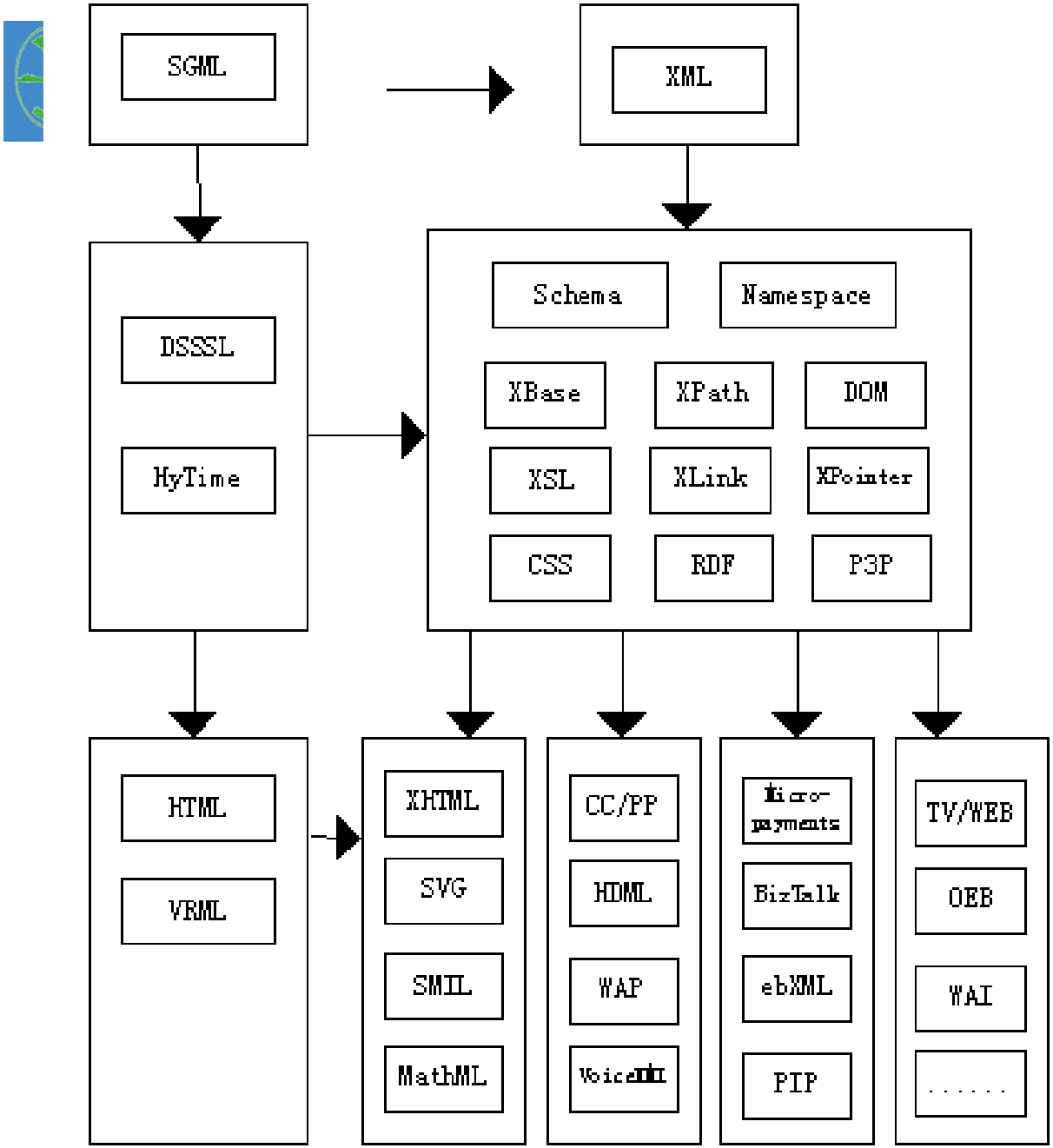


XML应用标准

- XML是一种元语言，是一种允许创建无限种不同用途的不同标记语言的技术。XML包含一组规则，任何人都可以根据这组规则创建标记语言
- 各行业领域纷纷采用XML制定针对Internet的应用标准，以适应本行业内部的信息交流和存档需要

- XML成功的关键：

  所有可用它定义的不同的特殊用途的语言都能被一个标准化的处理程序解析，这一程序小得可以内嵌入每一个Web浏览器中



元语言标准

基础标准

应用标准

XML标准体系框架

XML主要内容

- 基本概念
- XML 文档结构定义
- XML 名字空间
- XML文档的解析
- XML的标准体系
- XML技术小结

XML小结

- XML的基本思想：利用数据标识tag表示数据的含义，利用简单的嵌套和引用来实现数据元素之间的关系
  - 信息提供者能够根据需要，自行定义新的标识及属性名
  - 文件结构的嵌套可以复杂到任意程度
  - XML具有自描述性，文件可以包括一个语法结构描述，使应用程序可以对此文件进行结构确认

XML主要优点

- XML提供了一种功能强大、灵活高效地表达数据内容的方法
- XML具有很强的可扩展性(通过定义新的DTD/Schema)
- XML中的数据内容与具体应用无关，使得用它表示的数据具有很好的使用效率和可重用性
- XML通过DTD / Schema使得所包含的数据具有自描述性，可被不同程序用于不同目的
- XML采用一种开放的、以文本为基础的格式，可用从文本编辑器直到可视化开发环境的任何工具创建和编辑，使得程序可以更简单

XML主要优点（续）

- 数据和显示相分离，可由XSL指定如何显示数据。可以为同一数据指定不同的样式表用于不同输出，并很容易改变显示格式
- 具有很强的链接能力，可以定义双向链接、多目标链接、扩展链接和两个文档间的链接
- 易于处理。XML对格式的定义严格，并具有层次结构，处理起来更加容易
- 是与厂商无关的标准，可以任选一个解析器来处理

主要内容

- Web技术的发展历程
- Web基本支撑技术
- XML技术
- 语义WEB
- Web 2.0
- Web Service
- 小结

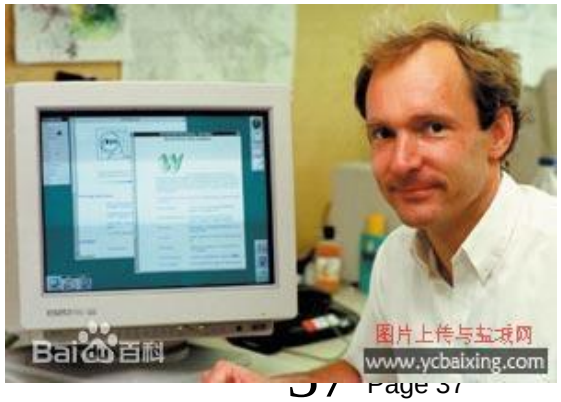
语义Web主要内容

- Why Semantic Web
- What's Semantic Web
- What's RDF
- What's Ontology
- 从语义网到知识图谱

万维网之父Berners-Lee的两个梦想

- 1、Web成为人们相互合作的强大工具

- 2、合作延伸到计算机，机器可以分析Web上的所有数据，包括数据、链接以及人与计算机之间的交互。
  - 语义Web（Semantic Web,也叫语义网）是实现Tim Berners-Lee梦想的关键。
    - ➔“语义”就是文本（数据）的含义，它和语法是一对对应的概念

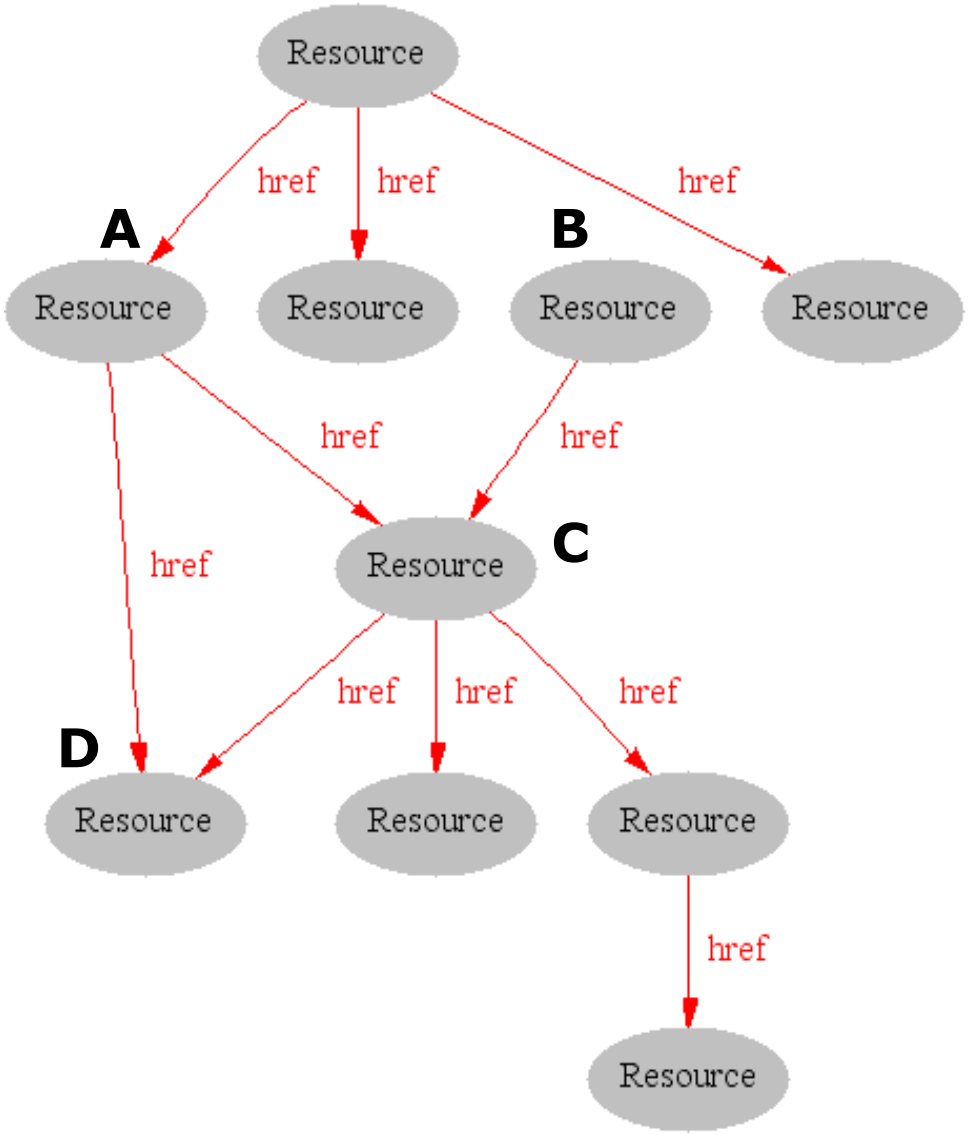


语义Web出现前的Web情况

- 网络的迅速普及已经成为信息时代的重要标志
- Web专门为人所设计，网页上的文本、图形、图像等各种媒体的涌现均是供人们阅读的，而网络和计算机只是作为一种简单的中介工具
  - 计算机并不能“理解” Web的内容，并在“理解”的前提下处理和利用这些信息
- 如何从浩瀚的信息海洋中快速准确地找到所需要的信息已经成为人们越来越关心的问题
- 由于Web数据的自身特点：动态性、分布性、无结构或半结构性等特点，使得实现上述任务非常困难

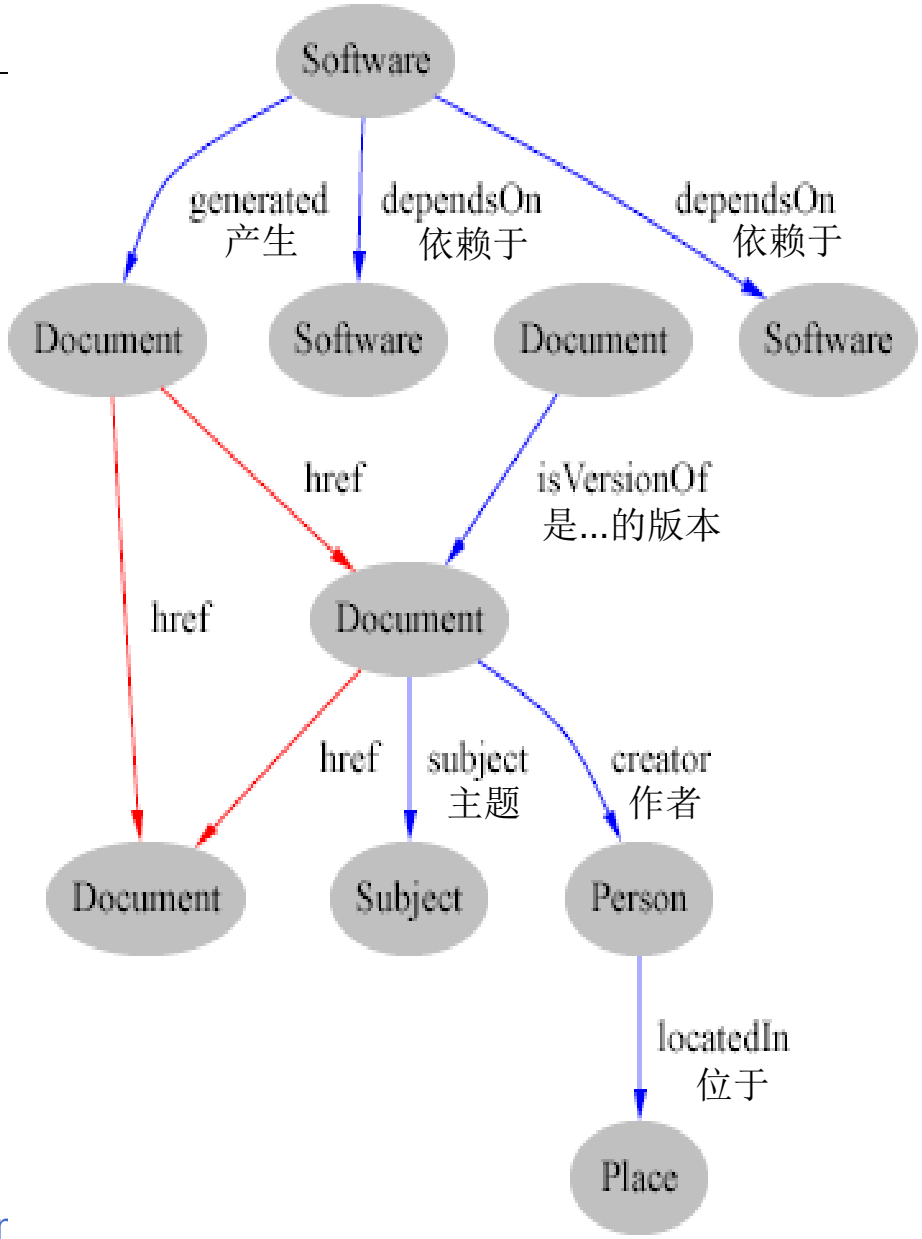
传统Web网络

- 资源通过极有限的语义相互链接（如：具有一定表达意义的文件名）
- 一个资源所在的系统无法自动判定其它系统中的资源含义（如：图中资源D通过href与资源A和C建立了链接，但是无法知道A和C的含义）



语义网

- 每个资源有清晰的定义，如软件，文件，人，地点
- 概念之间的关系有清晰的定义，如软件产生文件，软件依赖于软件，文件的版本，文件具有主题，人所在地点



语义Web主要内容

- Why Semantic Web
- What's Semantic Web
- What's RDF
- What's Ontology
- 从语义网到知识图谱

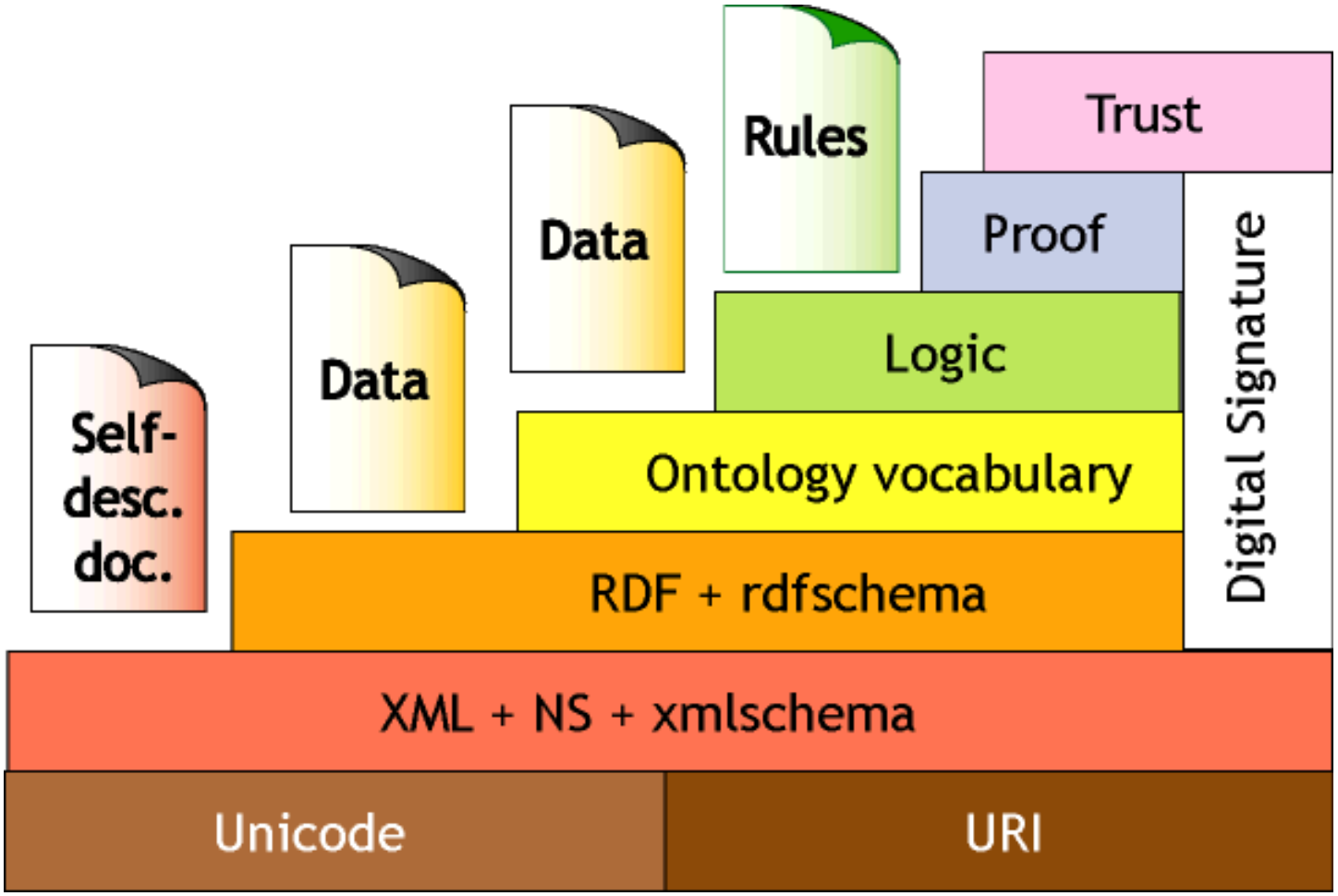
What is Semantic Web?

- Berners-Lee给出如下定义：语义Web是一个网，它包含了文档和文档的一些部分，描述了事物间的明显关系，且包含语义信息，以便于机器的自动处理
  - 语义网是对Web的一种扩展，是一个信息的 Web，这些信息被赋予明确定义的含义，是机器可处理的
    - ➔对信息来说，“Semantic”是信息的含义,也就是说表达了对信息能作哪些操作
  - 其目标：机器能够自动处理和集成网上可用的信息
    - ➔全局数据集成、信息共享、知识交换，发现新知识，实现面向内容的信息管理
  - 需要合适的数据表示方法

语义描述的基础：元数据

- 元数据：描述数据的数据。主要包括
  - 语法元数据：描述有关数据内容的非上下文信息。如文档的大小、位置、创建日期等
  - 结构元数据：提供有关内容结构的信息，如使用DTD或者XML Schema 可以定义XML文档用到的元素、属性和实体以及不同元素和属性之间的关系
  - 语义元数据：提供数据的语义信息，也能对语法和结构元数据添加关系、规则和约束条件

Semantic Web - Layers



Semantic Web－Layers

- 该体系中从低层到高层分别为：Unicode和URI、XML、RDF、Ontology、Logic、Proof、Trust
- 第一层Unicode和URI。该层是整个语义Web的基础，其中Unicode处理资源的编码，URI负责标识资源
- 第二层XML+NS+XML Schema，用于表示数据的内容和结构
- 第三层RDF+RDF Schema，用于描述Web上的资源及其属性，可简单描述资源间的关系
- 第四层Ontology vocabulary，用于描述各种资源之间的联系
- 第五层到第七层是在下面4层的基础上进行的逻辑推理操作，包括公理和推理规则、认证机制及信任机制
- 其中核心层为XML、RDF、Ontology，这3层用于表示Web信息的语义

语义Web主要内容

- Why Semantic Web
- What's Semantic Web
- What's RDF
- What's Ontology
- 从语义网到知识图谱

XML描述语义？

- 对于陈述“David Billington is a lecturer of Discrete Mathematics.”可以表示为：

```
<course name="Discrete Mathematics">
        <lecturer >David Billington </lecturer>
</course>
```

- 和

```
<lecturer name="David Billington">
        <teaches >Discrete Mathematics< /teaches>
</lecturer>
```

XML标记嵌套的含义没有标准化，不利于机器的理解

资源描述框架（RDF）

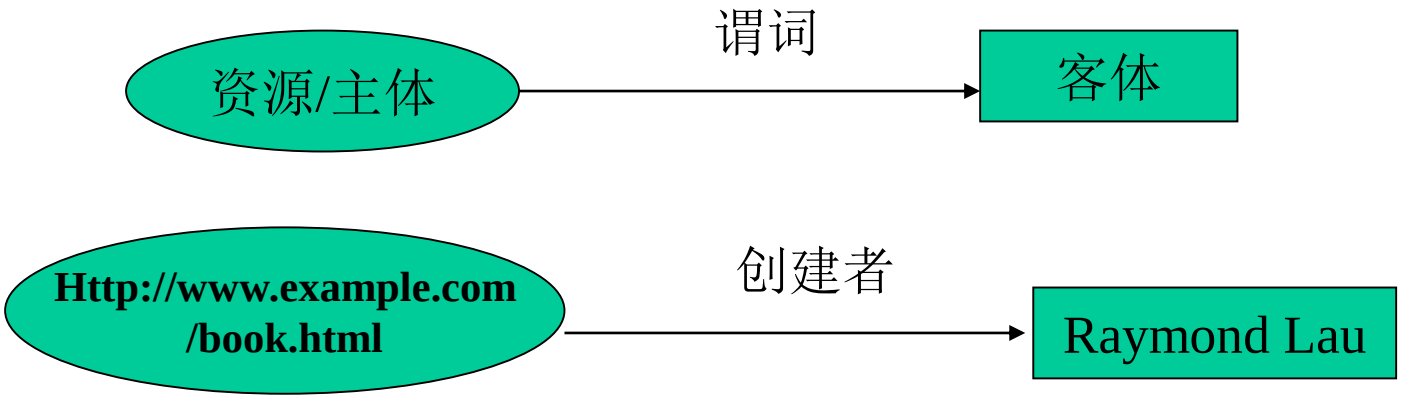
- RDF：一种描述资源的（属性）语言
  - 是W3C组织推荐的描述Web上元数据的标准
- 用Web标识符（URI）来标识事物，用简单的属性（property）及属性值来描述资源
- 主要包括三部分
  - RDF模型
  - RDF/XML语法
  - RDF Schema

RDF模型中的概念

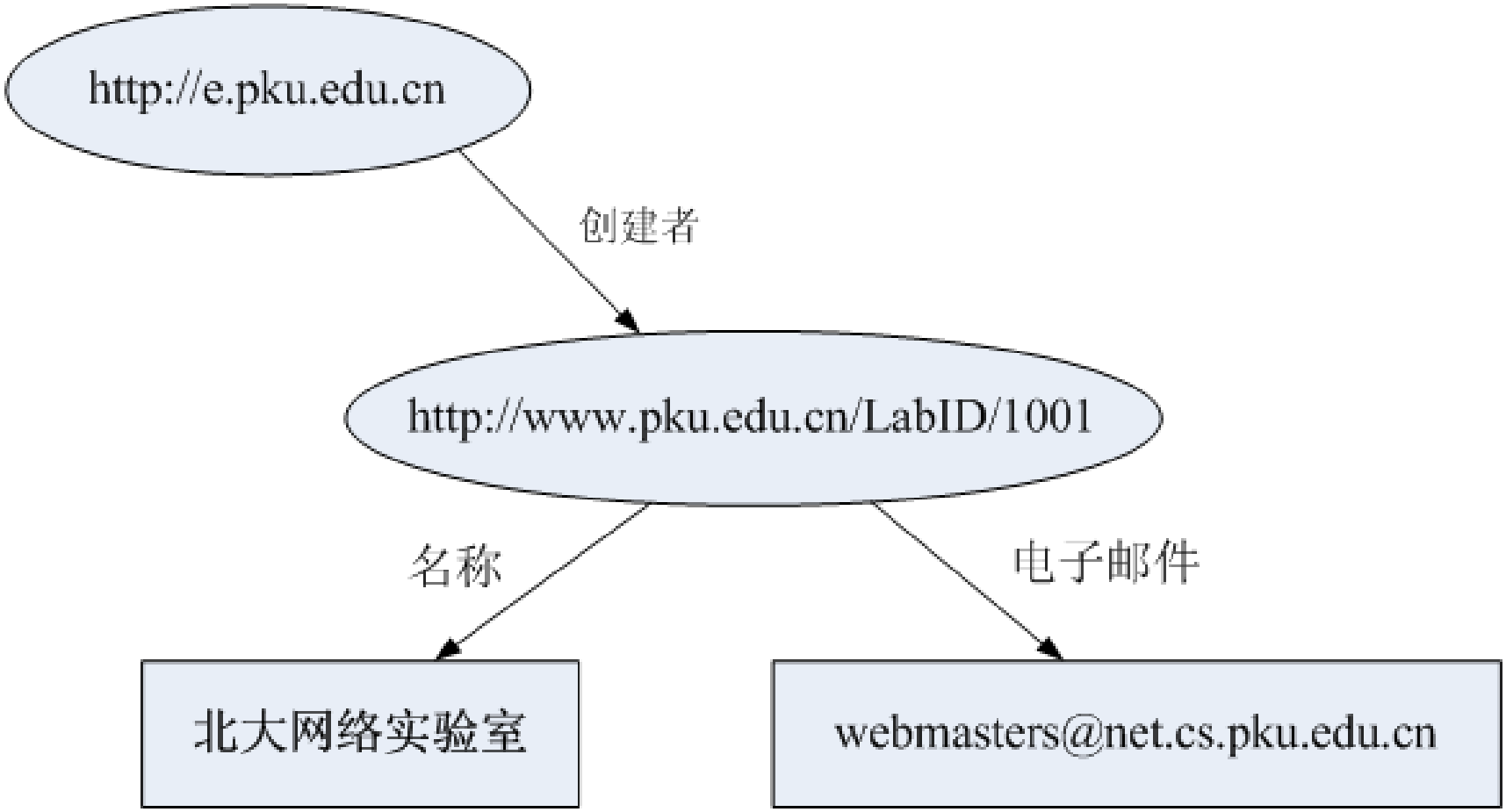
- 资源：资源就是一个打算谈论的“事物”，是一个由URI标识的对象，URI可以是URL或者其他形式的唯一标识符
- 属性：属性是一种特殊的资源，描述资源之间的关系，如“由……讲授”、“年龄”、“头衔”等等。RDF中属性仍然由URI标识
- 陈述：陈述用于描述资源所具有的属性
- 一个RDF声明或陈述由一个三元组表示（Subject，Predicate，Object）
  - 主体和谓词的类型是URI，表示一个Web 资源
  - 客体即可以是URI，也可以是文字（Literal）
  - 如：(Http://www.example.com/book.html， Creator，Raymond Lau)

RDF的图模型

- RDF的概念模型是图模型：可把RDF声明表示为有向、带标签的图
  - 椭圆节点表示用URI指代的主体和客体
  - 方框节点表示文字类型的客体
  - 有向弧标签则表示用URI指代的谓词



图模型例


http://e.pku.edu.cn
创建者
http://www.pku.edu.cn/LabID/1001
名称
电子邮件
北大网络实验室
webmasters@net.cs.pku.edu.cn

RDF /XML 语法

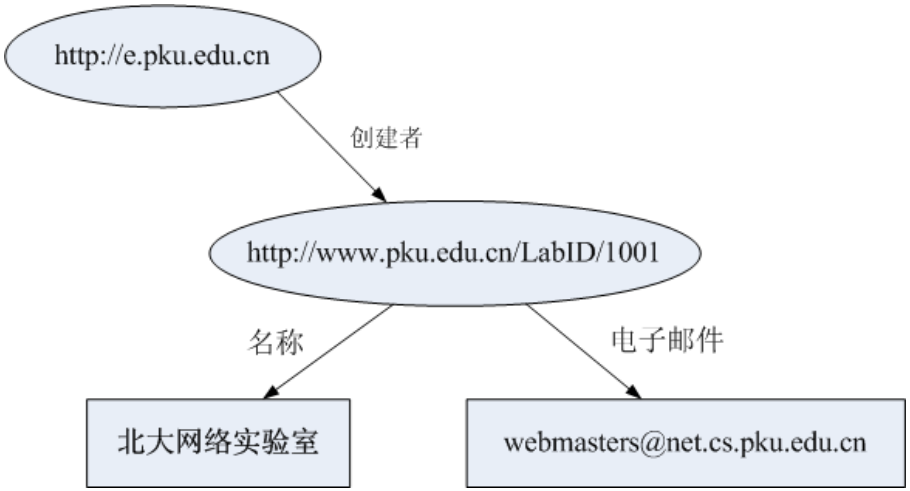
- RDF提供了一套XML语法，称为RDF/XML语法，用于RDF图的表示和交换。如
  - 对应于每个RDF三元组中的主体，生成一个rdf:Description元素
  - 同一个主体的多个声明可合并为一个rdf：Description元素

RDF 的XML 表示例

```
< rdf: RDF >
   < rdf: Description about = "http://e.pku.edu.cn/" >
      < s: Creator >
         < rdf: Description about = "http://www.pku.edu.cn/LabID/1001" >
            < v: Name > 北大网络实验室 < /v: Name >
            < v: Email > webmasters@net.cs.pku.edu.cn < /v: Email >
         < /rdf: Description >
      < /s: Creator >

      . . . . . .

< /rdf: RDF >
```



RDF Schema

- RDF只是定义了一种描述资源的框架，并没有定义可以使用哪些词汇对资源进行描述
- RDF Schema（简称RDFS）实现对RDF的扩展，可用于描述和定义与特定应用相关的类和属性，从而可创建自定义的词汇表

- 描述RDF类：类是指事物的类型或者分类
  - rdfs:Class、rdfs:Resouce、rdfs:type、rdfs:subClassOf
    - ➔<ex:Teacher, rdfs:type, rdfs:Class>
    - ➔<expersons:Paymond, rdfs:type, ex:Teacher>
- 描述RDF属性
  - rdfs:Property、 rdfs:range、rdfs:domain
    - ➔<exterms:name, rdfs:type, rdfs:Property>
    - ➔<exterms:name, rdfs:range, exterms:String>

RDFS 部分类构造符

表 1　RDF Schema 类构造符

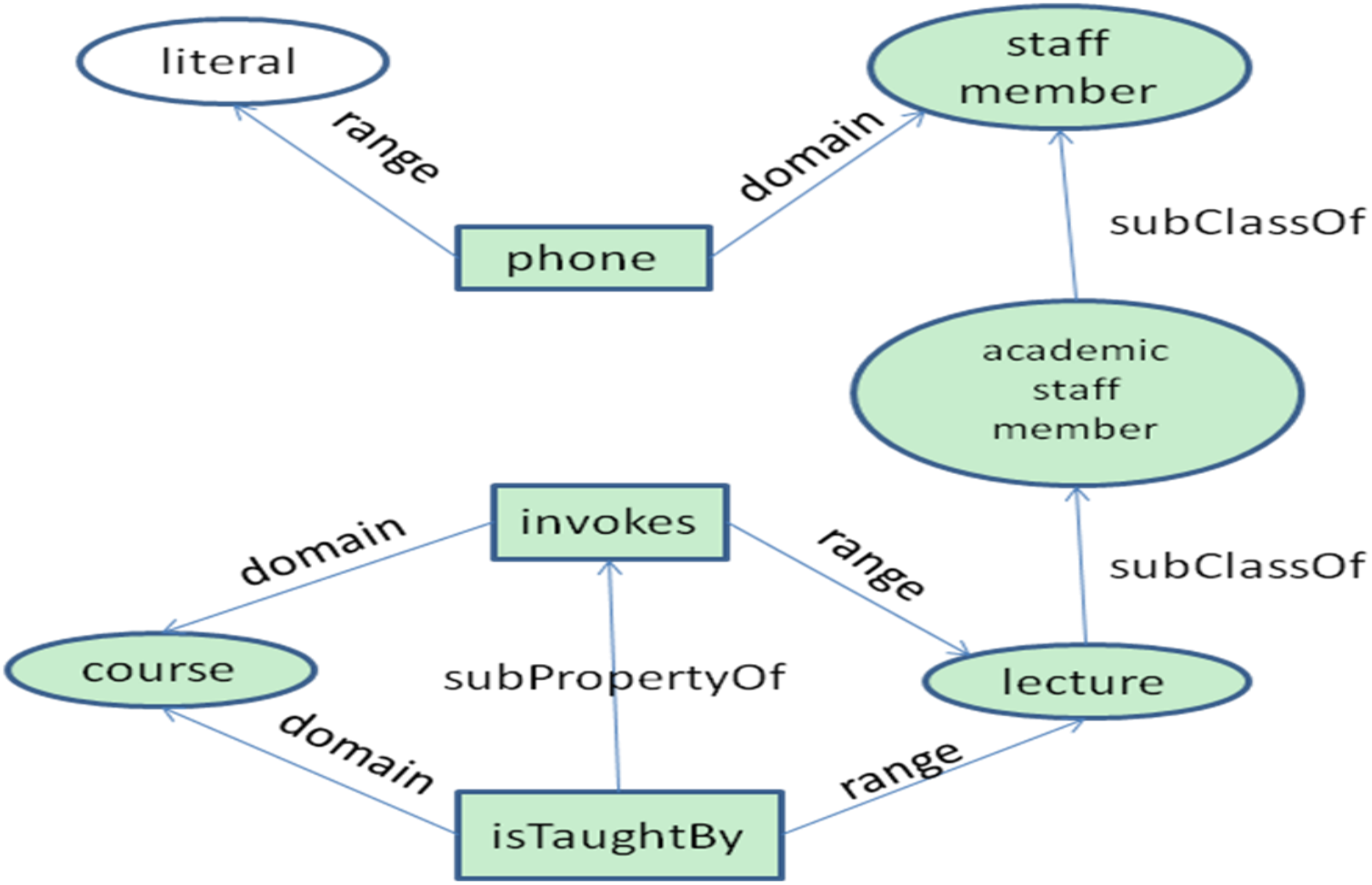
类 名	注 释
rdfs: Resource	所有的东西都称做资源
rdfs: Literal	作为属性的值,大多是数字和字符串
rdf: XMLLiteral	XML 值
rdfs: Class	定义类
rdf: Property	属性定义
rdfs: Datatype	自定义 RDF 中的数据类型
rdf: Statement	RDF 声明
rdf: Bag	无序的容器
rdf: Seq	有序的容器
rdf: Alt	可选取的容器
rdfs: Container	RDF 容器
rdfs: ContainerMembershipProperty	容器成员的属性
rdf: List	RDF 列表

RDFS 部分属性构造符

表 2　RDF Schema 属性构造符

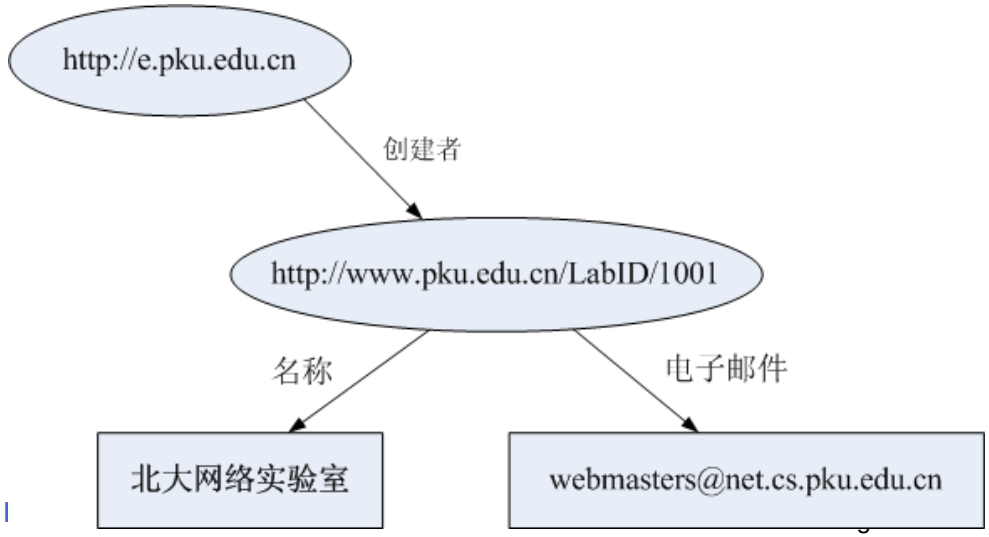
属性名	注 释
rdf: type	类的实例
rdfs: subClassOf	子类
rdfs: subPropertyOf	子属性
rdfs: domain	属性的域
rdfs: range	属性范围
rdfs: label	人类可理解的符号
rdfs: comment	资源描述说明
rdfs: member	资源的成员
rdf: firstRDF	列表的第一项
rdf: restRDF	列表第一项以后的所有剩余项
rdfs: seeAlso	关于资源的进一步信息
rdfs: isDefinedBy	资源的定义
rdf: value	属性值
rdf: subject	RDF 声明的主语
rdf: predicate	RDF 声明的谓语
rdf: object	RDF 声明的宾语

RDFS 实例


literal
range
staff member
domain
phone
subClassOf
academic staff member
domain
invokes
range
subClassOf
course
subPropertyOf
lecture
domain
range
isTaughtBy

RDF基本思想小结

- RDF的设计基于这一思想：
  - 被描述的资源具有一些属性，而这些属性各有其值
  - 对资源的描述就是对资源的属性及值进行声明
  - 属性的值既可以是文字（Literal）也可以是其他资源
  - 如果属性值是资源，该属性也可以看作是两个资源之间的关系



语义Web主要内容

- Why Semantic Web
- What's Semantic Web
- What's RDF
- What's Ontology
- 从语义网到知识图谱

RDF、RDFS的局限性

- 类的不相交性：有时候需要表示类的不相交性，比如 男性 和 女性 是不相交的。但RDFs只定义了类间的子类关系
- 类的布尔组合：有时希望通过对已有类的并、交或补等操作，组合产生新的类，如定义人类为男性和女性不相交的并
- 基数约束：有时需要对一个属性不同取值的个数加以约束。比如一个人只有一个亲生父亲和亲生母亲
- 属性的特殊性质：有时需要规定属性具有传递性、唯一性或定义属性的逆属性，比如“吃”与“被吃”

- 本体层就是要提供一个能明确地形式化地定义术语含义及术语间关系的语言

Ontology的发展历程

范畴	提出时间/提出人	定义
哲学		客观存在的一个系统的解释和说明，客观现实的一个抽象本质
计算机	1991/Neches等	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
	1993/Gruber	概念模型的明确的规范说明
	1997/Borst	共享概念模型的形式化规范说明
	1998/Studer	共享概念模型的明确的形式化规范说明

What is Ontology ...

- ***An ontology is a formal, explicit specification of a shared conceptualization*** ：共享概念模型的明确的形式化的规范说明
  - ‘Conceptualization’概念模型指通过抽象出客观世界中一些现象的相关概念而得到的模型
    - ➔**概念 (Concept):人们对事物本质的认识，逻辑思维的最基本单元和形式 。如，人们在生活实践中逐步抽象出力学相关的概念：重力、质量、加速度、摩擦力等**
    - ➔**这些概念相互间有关系，也有相应的公理、定理等**
  - ‘Shared’共享指Ontology中体现的是共同认可的知识，反映的是相关领域中公认的概念集
  - ‘Explicit’明确指所使用的概念及使用这些概念的约束都有明确的定义
  - ‘Formal’形式化指Ontology是计算机可读的

Ontology的建模元语

- 本体是领域知识的形式化说明，通常由概念、概念之间的关系、公理、规则组成，Five kinds of components:
  - 类（classes）或概念（concepts）
    - 指任何事务，如工作描述、功能、行为、策略和推理过程。从语义上讲，它表示的是对象的集合，其定义一般采用框架（frame）结构，包括概念的名称，与其他概念之间的关系的集合，以及用自然语言对概念的描述
    - e.g. In university-ontology: student and professor are two classes
  - 关系（relations）
    - 在领域中概念之间的交互作用，形式上定义为n维笛卡儿积的子集：$R:C_1 \times C_2 \times ... \times C_n$
    - Such as: subclass-of, is-a

Ontology的建模元语(Cont.)

- 函数（functions）
  - ➔一类特殊的关系。该关系的前n−1个元素可以唯一决定第n个元素。形式化的定义为$F:C_1 \times C_2 \times ... \times C_{n-1} \rightarrow C_n$
  - ➔如Mother-of就是一个函数，mother-of(x,y)表示y是x的母亲
- 公理（axioms）
  - ➔代表永真断言
  - ➔如概念乙属于概念甲的范围
- 实例（instances）
  - ➔代表元素。从语义上讲实例表示的就是对象
  - ➔E.g. Student called Peter is the instance of Student class

4种基本关系

关系名	关系描述
part-of	表达概念之间部分与整体的关系。
kind-of	表达概念之间的继承关系，类似于面向对象中的父类与子类之间的关系。
instance-of	表达概念的实例与概念之间的关系，类似于面向对象中的对象和类之间的关系。
attribute-of	表达某个概念是另一个概念的属性。如“价格”是桌子的一个属性。

简单推理例

- 定义基本谓词：

mother(X,Y)	X是Y的母亲
father(X,Y)	X是Y的父亲
male(X)	X是男性
female(X)	X是女性
parent(X,Y)	X是Y的父母
brother(X,Y)	X是Y的兄弟
notSame(X,Y)	X和Y是不同的个体

简单推理例(2)

- 定义规则：

male(X),male(Y),parent(P,X),parent(P,Y),notSame(X,Y)->brother(X,Y)

- 实例：

male(A1),

male(A2),

father(B，A1),

father(B,A2),

notSame(A1,A2)

得出 A1有个兄弟叫A2

构造Ontology的规则

最有影响的是Gruber在1995年提出的5条规则：

- 明确性和客观性：Ontology应该用自然语言对所定义的术语给出明确、客观的语义定义
- 完全性：所给出的定义是完整的，完全能表达所描述的术语的含义
- 一致性：由术语得出的推论与术语本身的含义是相容的，不会产生矛盾
- 最大单调可扩展性：向Ontology中添加通用或专用的术语时，不需要修改已有的内容
- 最小承诺：对待建模对象给出尽可能少的约束

目前大家公认在构造特定领域的Ontology的过程中需要领域专家的参与

本体例

名称	描述
WordNet	基于心理语言规则的英文词典，以synsets（在特定的上下文环境中可互换的同义词的集合）为单位组织信息。
FrameNet	英文词典，采用称为Frame Semantics的描述框架，提供很强的语义分析能力，目前发展为FramenetII。
GUM	面向自然语言处理，支持多语种处理，包括基本概念及独立于各种具体语言的概念组织方式。
SENSUS	面向自然语言处理，为机器翻译提供概念结构，包括7万多概念。
Mikrokmos	面向自然语言处理，支持多语种处理，采用一种语言中间的中间语言TMR表示知识。

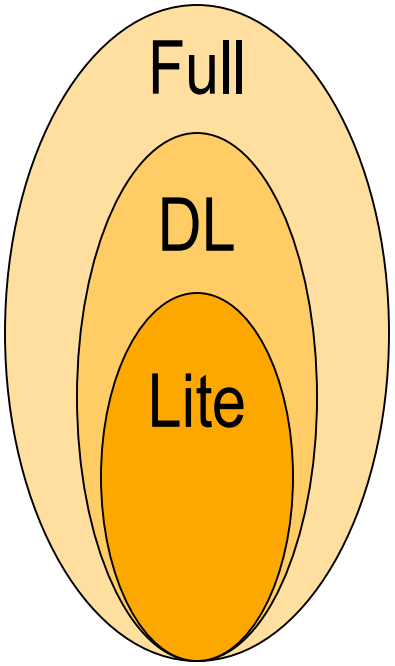
本体语言

- RDFS无法满足本体定义的要求，如：
  - 不能表示两个类的互不相交关系、类的基数和等价性等
  - 无法对属性的传递性、对称性、唯一性等进行定义
  - 没有定义推理和公理的机制
- 本体语言：
  - 能够描述（定义）本体并使得他们能够进行信息交换的语言标准
- 主要特点：
  - 语法和语义丰富
  - 描述的信息尽量涵盖半结构化的自然语言
  - 使用共有的、一致的字典以利于信息共享和交换
  - 提供一种领域理论

本体语言例：OWL

- W3C 提出的OWL Web 本体语言（OWL Web Ontology Language） 是一种在语义Web环境中表示本体的推荐标准。

- OWL的三个子语言
  - OWL Lite
  - OWL DL
  - OWL Full



OWL Example: animals

```
<?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>
```

OWL 支持的部分类构造符

表 5　OWL 支持的类构造符

构造符号	含义	描述逻辑式	例子
IntersectionOf	类交	$C_1 \cap, \cdots, \cap C_n$	Human ∩ Male
UnionOf	类并	$C_1 \cup, \cdots, \cup C_n$	Male ∪ Female
ComplementOf	类补	¬ C	¬ Male
OneOf	枚举	$\{x_1, \cdots, x_n\}$	{John, Mary}
AllValuesFrom	全称限定	∀P. C	∀hasChild. Lawyer
SomeValueFrom	存在限定	∃P. C	∃hasChild. Lawyer
HasValue	值限定	∃P. {x}	∃hasChild. {Mary}
MinCardinality	最小基数	≥nP	≥2hasChild
MaxCardinality	最大基数	≤nP	≤1hasChild
Cardinality	基数值	=nP	=1hasChild

OWL 支持的部分公理

表 6　OWL 中的公理

公理	含义	描述逻辑	例子
SubClassOf	子类	$C_1 \subseteq C_2$	Human⊆Animal∩ Biped
EquivalentClass	类等价	$C_1 \equiv C_2$	Man≡Human∩Male
SubPropertyOf	子属性	$P_1 \subseteq P_2$	hasDaughter⊆hasChile
EquivalentProperty	属性等价	$P_1 \equiv P_2$	cost≡price
DisjointWith	不相交	$C_1 \subseteq \neg C_2$	Male⊆¬ Female
SameAs	个体等价	$\{x_1\} \equiv \{x_2\}$	{President_Bush} ≡ {G_W_Bush}
DifferentFrom	个体不等	$\{x\} \subseteq \neg \{x\}$	{John} ⊆¬ {Peter}
InverseOf	互逆属性	$P_1 \equiv P_2^-$	hasChild≡hasParent¯
TransitiveProperty	传递属性	$P^- \subseteq P$	ancestor¯⊆ancestor
SymmetricProperty	对称属性	$P^- \equiv P$	equality¯≡equality
FunctionalProperty	惟一属性（宾语）	$T \subseteq \leqslant 1P$	T⊆⩽1hasMother
InverseFunctional-Property	惟一属性（主语）	$T \subseteq \leqslant 1P^-$	T⊆⩽1ismotherOf

Ontology语言栈

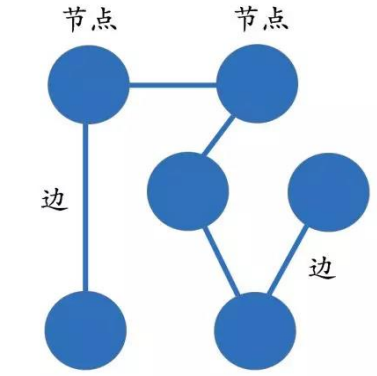
名称	描述
XML	结构化文档的表层语法
XML Schema	定义XML文档的结构约束的语言。
RDF	定义对象（或者资源）以及它们之间关系的数据模型，为数据模型提供了简单的语义，这些数据模型能够用XML语法进行表达。
RDF Schema	描述RDF资源的的属性和类型的词汇表，提供了对这些属性和类型的普遍层次的语义。
OWL	添加了更多的用于描述属性和类型的词汇，例如类型之间的不相交性（disjointness），基数（cardinality），等价性，属性的更丰富的类型，属性特征（例如对称性，symmetry），以及枚举类型（enumerated classes）等。

语义Web主要内容

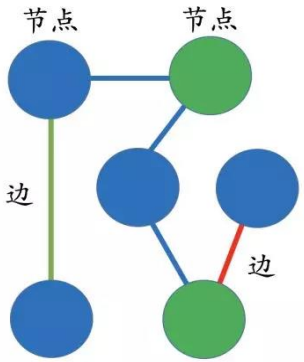
- Why Semantic Web
- What's Semantic Web
- What's RDF
- What's Ontology
- 从语义网到知识图谱

从语义网到知识图谱

- 知识图谱是由Google公司在2012年提出来的一个新概念
  - 某定义：A knowledge graph consists of a set of interconnected typed entities and their attributes
  - 可以简单地把知识图谱理解成多关系图（Multi-relational Graph）本质上就是语义网络（Semantic Network）的知识库
    - ➔Google的知识图谱是从Freebase导入的，Freebase是一个语义网项目，使用了RDF模型，用URI标识数据，还定义了自己的本体结构



包含一种类型的节点和边



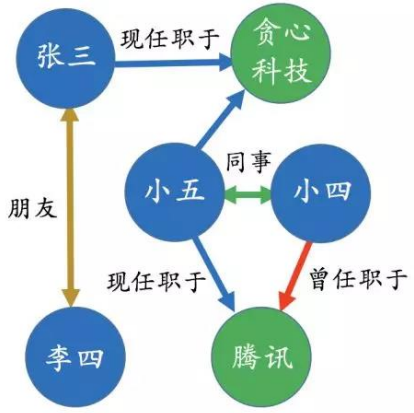
包含多种类型的节点和边

（不同形状和颜色代表不同种类的节点和边）

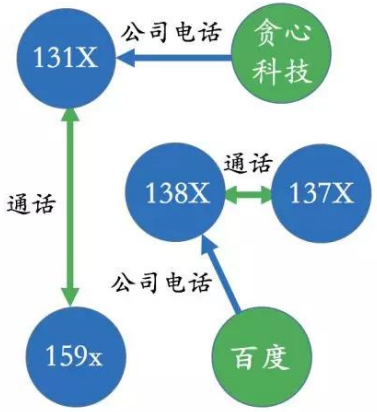
https://www.jiqizhixin.com/articles/2018-06-19

知识图谱中的实体和关系

- 在知识图谱里，通常用“实体（Entity）”来表达图里的节点、用“关系（Relation）”来表达图里的“边”。
  - 实体指的是现实世界中的事物比如人、地名、概念、药物、公司等
  - 关系则用来表达不同实体之间的某种联系
  - 根据是实体和实体之间的关系还是实体和数据值之间的关系分为对象属性（Object Property）和数据属性（Data Property）
- 知识图谱的构建是后续应用的基础，构建的前提是需要把数据从不同的数据源中抽取出来，抽取的难点在于处理非结构化数据



案例：社交网络



案例：风控知识图谱

实体命名识别

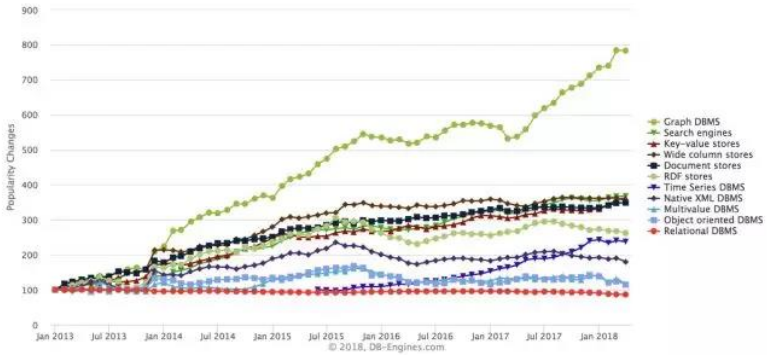
This hotel is my favorite Hilton property in NYC! It is located right on 42nd street near Times Square in New York, it is close to all subways, Broadways shows and next to great restaurants like Junior's Cheesecake, Virgil's BBQ

关系抽取

知识图谱的存储

- 知识图谱主要有两种存储方式：一种是基于RDF的存储，另一种是基于图数据库的存储

RDF	图数据库
存储三元组（Triple）	节点和关系可以带有属性
标准的推理引擎	没有标准的推理引擎
W3C标准	图的遍历效率高
易于发布数据	事务管理
多数为学术界场景	基本为工业界场景



数据库使用率增长

排名	数据库
22	Neo4j(图数据库)
38	MarkLogic(XML)
49	OrientDB(图，文档)
85	Jena(RDF)

部分图数据库排名

知识图谱系统的搭建

- 最重要的核心在于对业务的理解以及对知识图谱本身的设计，这种设计绝对离不开对业务的深入理解以及对未来业务场景变化的预估

1. 业务理解	10%
2. 知识图谱设计	10%
3. 算法	50%
4. 开发	30%

很多人以为…

1. 业务理解	30%
2. 知识图谱设计	30%
3. 算法	20%
4. 开发	20%

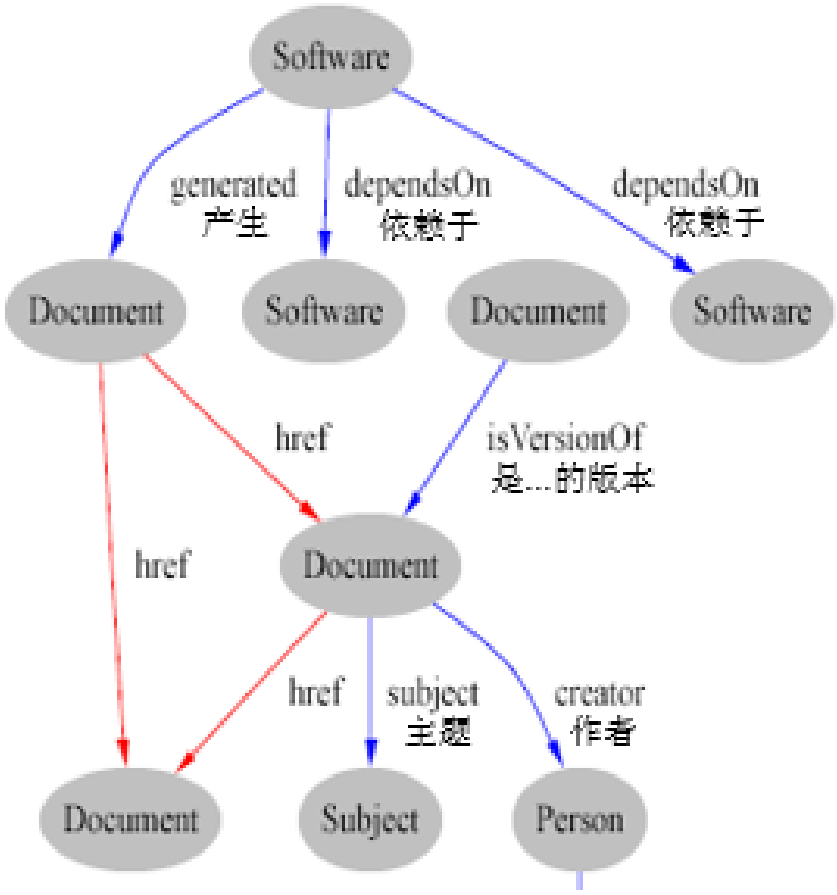
其实…

- 知识图谱目前的主要作用在于分析关系
  - 基于规则的知识推理
  - 基于深度学习、概率统计的知识推理

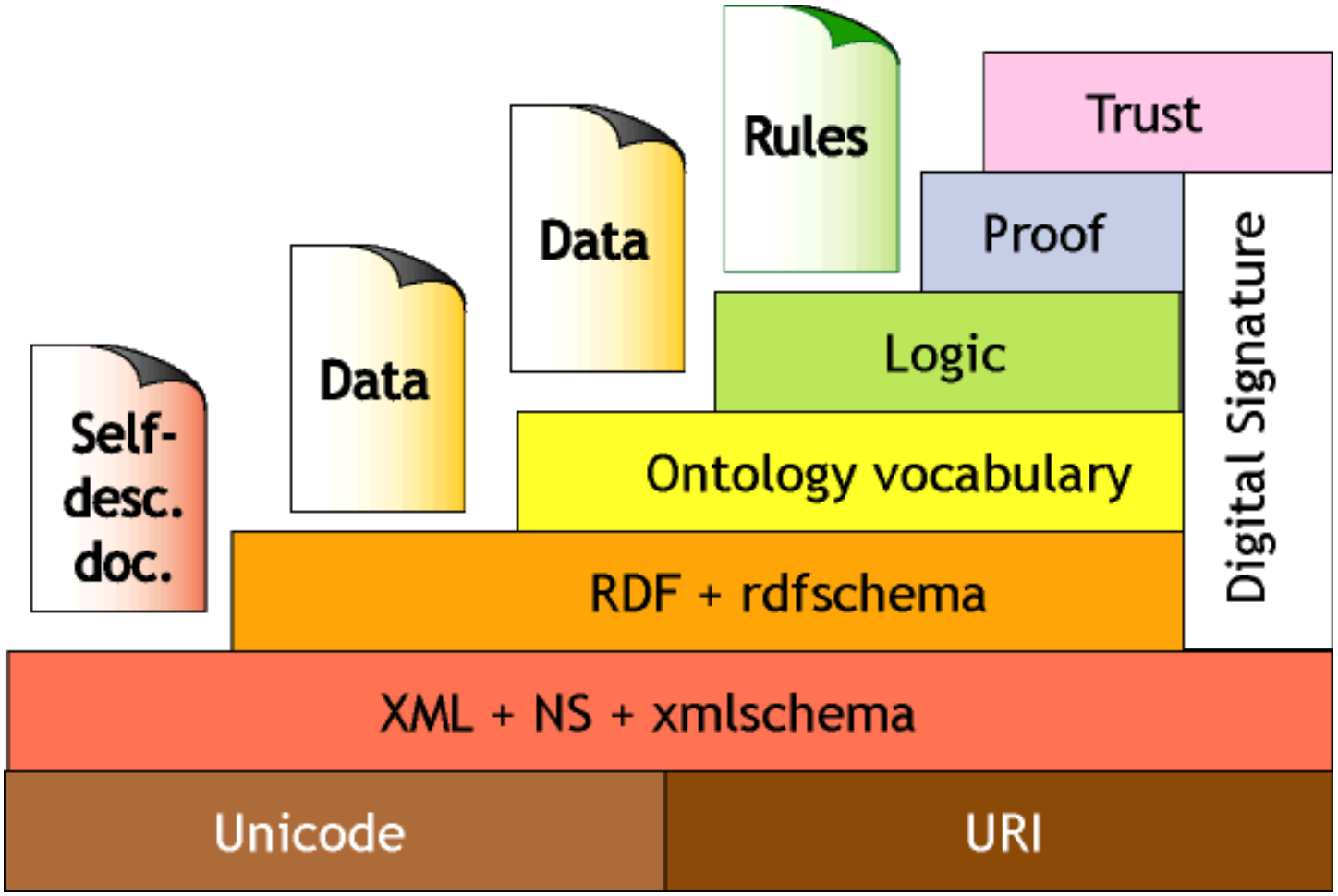
https://www.jiqizhixin.com/articles/2018-06-19

语义Web小结

- 传统的Web是用超链将文档连接起来，而语义Web是用语义链将Web上各种数据和资源连接起来，以发挥Web上数据和资源的最大潜力
- 语义Web的目标是改善传统的Web。它的主要思想是使语义信息成为计算机可处理的对象。它的主要技术主要包括元数据表示、本体、逻辑推理和智能主体技术等。语义Web的发展取决于其各层技术的发展。



语义Web小结(2)


Rules
Trust
Data
Proof
Data
Logic
Self-
desc.
doc.
Ontology vocabulary
Digital Signature
RDF + rdfschema
XML + NS + xmlschema
Unicode
URI

语义Web小结(3)

- 虽然语义Web有了良好的结构和框架，各层技术也有所发展，但每一点语义都是一条漫长的道路
  - 本体建立、本体匹配、逻辑推理等，无一不是很困难的问题
  - 语义Web很难一下子获得巨大的成功，它一点一点渗透到传统的Web中，最后在人们的不知不觉中，语义Web的时代就会到来
- 知识图谱技术其基础就是语义网
  - 知识图谱，是结构化的语义知识库，用于以符号形式描述物理世界中的概念及其相互关系。其基本组成单位是“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，构成网状的知识图谱