

如何用(X)HTML, XML 和RDF 对DC置标

*Tutorial at DC-2004, Shanghai
October 2004*

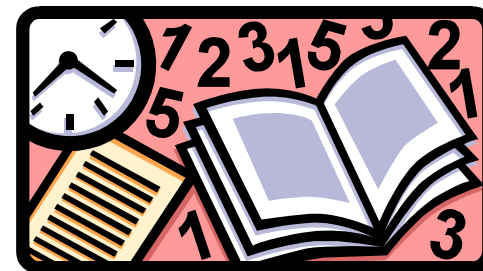
Andy Powell
a.powell@ukoln.ac.uk
UKOLN, University of Bath, UK
<http://www.ukoln.ac.uk/>

UKOLN由如下单位支持:



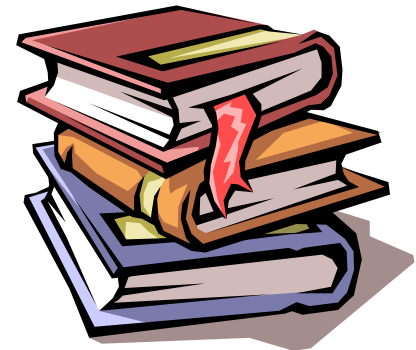
主要内容

- DC抽象模型 (30 mins)
- 用XHTML对DC置标 (15 mins)
- 用XML 对DC置标 (15 mins)
- 用RDF/XML对DC置标 (5 mins)
- 举例说明
 - OAI 协议和RSS (20 mins)



重要的 **DCMI** 文档...

- **DCMI Abstract Model – DRAFT**
<http://www.ukoln.ac.uk/metadata/dcmi/abstract-model/>
- **Expressing Dublin Core in HTML/XHTML meta and link elements**
<http://dublincore.org/documents/dcq-html/>
- **Guidelines for implementing Dublin Core in XML**
<http://dublincore.org/documents/dc-xml-guidelines/>
- **Expressing Simple Dublin Core in RDF/XML**
<http://dublincore.org/documents/dcmes-xml/>
- **Expressing Qualified Dublin Core in RDF/XML**
<http://dublincore.org/documents/dcq-rdf-xml/>
- **Namespace Policy for the DCMI**
<http://dublincore.org/documents/dcmi-namespace/>
- **DCMI Metadata Terms**
<http://dublincore.org/documents/dcmi-terms/>



DC的实现

- 本次培训主要涉及在HTML、XML及RDF中应用DC的机制；
- 本次培训并不考虑哪一种应用策略是最佳的！
- 问自己两个问题...
 - 我想达到什么目标？
 - 应用 HTML, XML 还是RDF 才能达到此目标？
- 有现成软件或服务来支持元数据的创建与使用吗？



DCMI抽象模型

(DCMI abstract model)



为什么要有DCMI抽象模型？

- 本次培训的第一部分并不介绍任何句法 (syntax)!
- 为什么？
- 因为在开始创建DCMI描述(DCMI descriptions)之前，我们需要理解哪些事物可以被称为“资源(resource)”
- 由DCMI抽象模型来解释(**DCMI abstract model**)
- 注意：这里只对此模型作简要介绍



资源的定义

- W3C/IETF 对资源的定义如下：



“.....可被标识的任何东西。常见的例子有：电子文档、图像和服务(如“洛杉矶今天的天气预报”)、资源集合。并不是所有的资源都是可通过网络获取的，如人、公司、图书馆流通的图书都可以被称作资源。

也就是说，资源可以是任何东西。

- 物理的东西(图书、汽车、人等)
- 数字的东西(网页，数字图像等)
- 概念性的东西 (颜色、时间点、主题等)

DC 和资源

- 但是.....看来我们想用DC来描述的东西太过宽泛了!
 - 我们真的能用 DC来描述人吗?
 - 人有名称(titles)和主题(subjects)吗?
- 不太合适..... 一般而言, 我们只用DC来描述所有资源中的一个子集。
- 比如任何出现在DCMI类型词表中的东西...
 - 资源集合, 数据集, 事件, 图象(静态或动态), 交互资源, 服务, 软件, 声音, 文本, 物理对象.....



DCMI抽象模型

- 一个描述(*description*)的组成如下:
 - 一个或多个声明(*statements*) (关于一个且仅一个资源) 及:
 - 可选的, 被描述资源的唯一标识符(*resource URI*)
- 一个声明(*statement*)的组成如下:
 - 一个*property URI* (属性的唯一标识符, 用来标识一个属性, 即*property*)
 - 一个*value URI* (值的唯一标识符, 用来标识这个值) 和/或一个或多个值的陈述 (*value representations*)



字符串值(Value strings)

- 每一个值的陈述(*value representation*)可以用“字符串值(*value string*)”,“多媒体值(*rich value*)”或“相关描述(*related description*)”的形式来表达
- 注意: 本次培训中, 我们不往下讨论 *rich values*和*related descriptions* 有什么区别
- 每个*value string*是简单的、人类可读的字符串, 用来表达资源所拥有属性的属性值
- 每一个*value string*可以有一个对应的*value string language*, 它是一个ISO 的语言标签, 如en-GB, 说明字符串值所用的语种



元素和限定

- 在DCMI,我们经常使用“元素 (element)”和“元素限定(element refinement)”两个术语
- 元素只不过是属性(*property*)的另一种说法
- 元素限定是一种特殊的属性(*property*)(子属性*sub-property*)，它含有另一个属性的部分语义，更为专指：
 - 举例： if “Ben is the **illustrator** of a Book” then it is also true to say that “Ben is a **contributor** to the Book”



编码体系(encoding schemes)

- 值(values)和字符串值(value strings)可以用编码体系(encoding schemes)来修饰(qualified)
- 词表编码体系(vocabulary encoding scheme)用于表示值的类别
 - 例如, 某个值来自于LCSH
- 句法编码体系(syntax encoding scheme)用于表示字符串值(value string)是怎样组成的
 - 例如, (“2004-10-12”)这个时间字符串值是根据W3CDTF 规则组成的



1:1原则

- 注意，该模型指出，每条描述(*description*)只描述，且仅能描述一个资源(*resource*)
- 这就是一般所指的1:1原则。
- 然而...



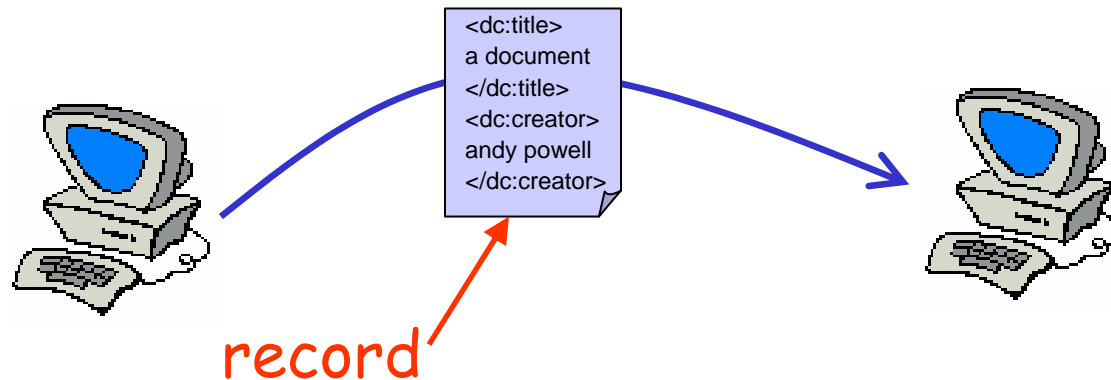
描述集(*Description sets*)

- 现实世界中的元数据应用倾向基于松散聚合的描述集(其中所描述的资源总是以某种方式相互关联)
- 在这儿称做描述集(*description sets*)
- 比如,一个描述集(*description set*)可能由对画的描述和对其画家的描述共同组成。



记录(Records)

- 为了在软件应用之间交换数据，描述集 (*description sets*)以元数据记录 (*records*) 的形式实例化
- 记录根据某个DCMI置标指南实例化 (如 XHTML meta tags, XML, RDF/XML, etc.)



简单DC还是限定DC?

- 在DCMI内部, 我们经常使用简单DC("simple DC")和限定DC("qualified DC")两个术语
- 简单DC("simple DC")仅使用DCMES的15个元素, 以字符串值(*value strings*)的形式支持一条描述
- 限定DC支持抽象模型中规定的所有特性, 允许使用所有的DCMI术语和其他非DCMI的术语

注意: 并不是所有的人都同意我的定义



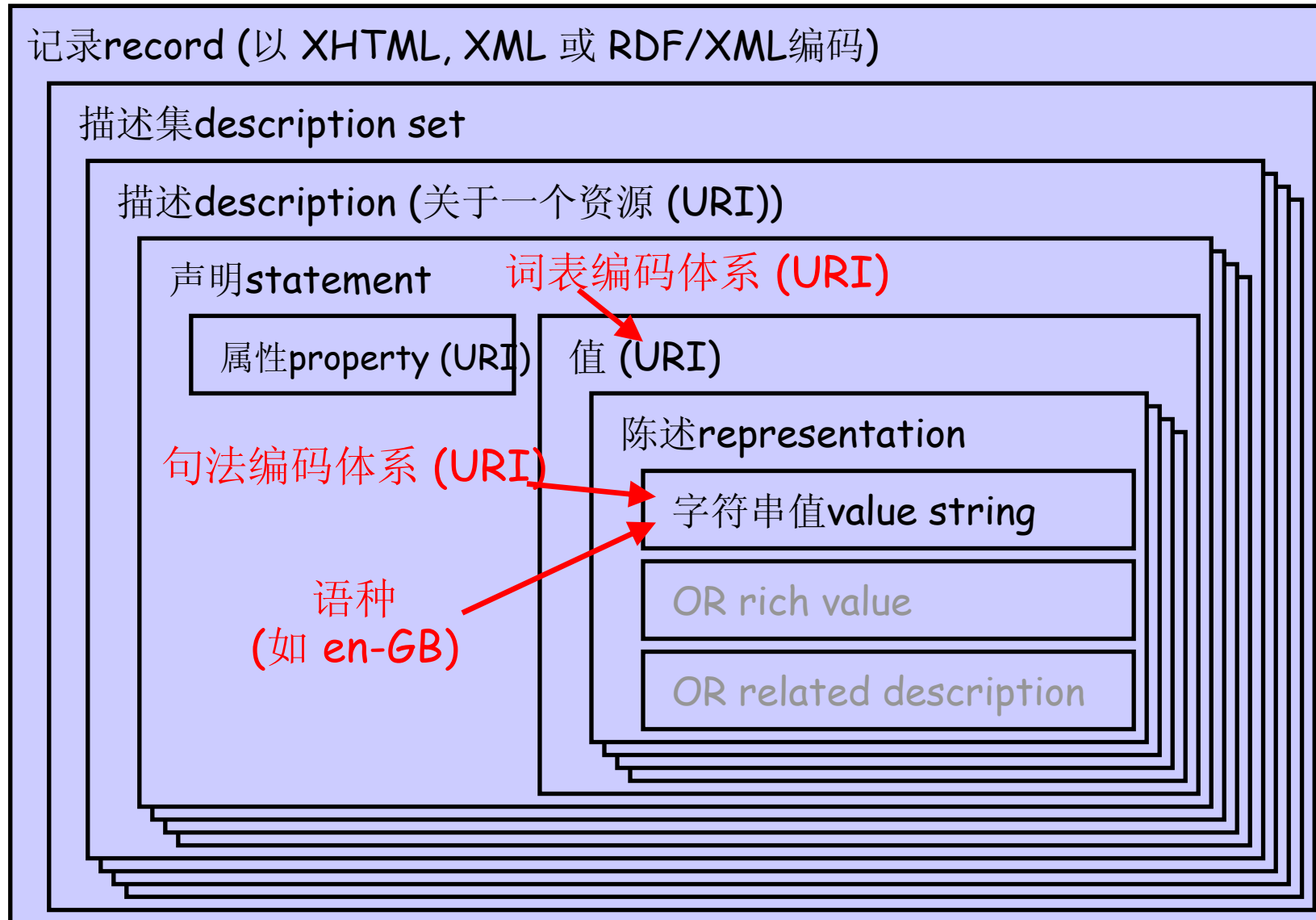
向上兼容原则(*Dumb-down*)

- 限定DC向简单DC转换的过程就称为“向上兼容”(dumbing-down)

	元素	值
uninformed	忽略任何不在DC元数据元素集(DCMES)的属性	使用URI值(<i>value URI</i> , 如果存在)或字符串值(<i>value string</i>)做为新的字符串值(<i>value string</i>)
informed	递归地解决子属性关系, 直到DC元数据元素集(DCMES)15个属性中的一个, 而不是忽略。	用 <i>rich values, related descriptions</i> 或 <i>value string</i> 和 <i>syntax encoding scheme</i> 来创建新的字符串值(<i>value string</i>)



Model summary



***Encoding DC in XHTML
(and HTML!)
用XHTML(和HTML)对DC编
码***



描述的是什么

- 一条DC描述嵌入一个(X)HTML文档中描述该文档
- 不要把关于其它东西的描述嵌入该文档。

...不是每个人都会同意这一点...



基础

- DC描述将嵌入到(X)HTML文档的 <head> 部分

```
<html >  
<head>  
...DC descri pti on goes here...  
</head>  
<body>  
...
```



DCMES的元素

- 使用XHTML中的<meta>元素的属性‘name’和‘content’来对DC15个元素和字符串值进行置标，使用如下形式：
`<meta name="DC.element" content="Value string" />`
- 例如：
`<meta name="DC.date" content="2001-07-18" />`



...15个DCMES元素的名称首字母一般小写...



字符串值(Value strings)

- 字符串值(*value strings*)在XHTML的 <meta> 元素的‘content’ 属性中...
- ‘content’ 属性中的字符串定义为CDATA, 即来自文档字符集中的一连串可能包括字符实体(**character entities**)的字符
...如果需要, 长的字符串值可以打包成多行...

...需要忽略某些特殊字符, 如 &, <, >,, etc...



字符串值(Value string)的语种

- 在说明字符串值(*value string*)语种的地方，应该用XHTML <meta>元素的‘xml:lang’属性，如：

```
<meta name="DC.subject" xml:lang="en"
content="seafood" />
<meta name="DC.subject" xml:lang="fr"
content="fruits de mer" />
```



置标时如何重复元素

- 多个属性值应通过重复 XHTML `<meta>` 元素来为这些属性置标, 例如:

```
<meta name="DC. title" content="First title" />
```

```
<meta name="DC. title" content="Second title" />
```



其他的DC元素

- DC还有元素不属于原来的DCMES15个元素中，如Audience
- 使用相同的形式置标，但改用‘DCTERMS’前缀：

```
<meta name="DCTERMS.element" content="Value" />
```

- 例如：

```
<meta name="DCTERMS.audience" content="software  
devel opers" />
```



...元素名称可以大小写混合，但首字母要小写...



元素限定(*element refinements*) 的置标

- 对元素限定使用相同的形式置标:

```
<meta name="DCTERMS. elementRefinement"  
      content="Value" />
```

- 例如:

```
<meta name="DCTERMS. modified"  
      content="2001-07-18" />
```



编码体系(encoding schemes)的 置标

- 编码体系使用XHTML <meta> 元素中的‘scheme’属性置标，使用如下形式：

```
<meta name="DC. element"  
      scheme="DCTERMS. Scheme"  
      content="Value" />
```

- 举例：

```
<meta name="DC. date"  
      scheme="DCTERMS. W3CDTF"  
      content="2001-07-18" />
```



名称的大小写问题

- 元素、元素限定、编码体系(*elements, element refinements and encoding schemes*) 应使用列入此文档中的名称:

DCMI Metadata Terms

<http://dublincore.org/documents/dcmi-terms/>



名称的大小写问题 (2)

- 元素和元素限定的名称可以是大小写混合，但首字母必须小写
- 编码体系的名称可以是大小写混合，但首字母必须大写

```
<meta name="DCTERMS. temporal "  
scheme="DCTERMS. Period" content="name=The  
Great Depression; start=1929; end=1939;" />
```



命名域的处理...

- ‘DC.’ 和 ‘DCTERMS.’两个前缀用来指明属性来自的命名域
- 把命名域的URI放在XHTML的<link>元素中:

```
<link rel="schema.DC"  
href="http://purl.org/dc/elements/1.1/" />  
<link rel="schema.DCTERMS"  
href="http://purl.org/dc/terms/" />
```

- 当然所有的字串都可以做为前缀的名称,现在使用的是 ‘DC.’ 和 ‘DCTERMS.’



值的URIs

- 当一个属性的值是另一资源的URI时 (如 DC.relation属性), 推荐使用另一种供选择的置标方式, 采用XHTML的<link> 元素, 使用如下形式:

```
<link rel="propertyName" href="valueURI" />
```

- 例如:

```
<link rel="DC.relation"  
href="http://www.example.org/" />
```

```
<link rel="DCTERMS.references"  
href="http://www.example.org/176459.pdf" />
```



DC元素和非DC元素混合时的置标

- 在XHTML <meta> 元素中，DC元数据可以与非DC元数据混合
- 下面的例子在同一个XHTML网页中内嵌了DC, AGLS和非标准的元数据：

```
<link rel="schema.DC"
href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.AGLS"
href="http://www.naa.gov.au/recordkeeping/gov
v_online/agls/1.2" />
<meta name="DC.title" content="Services to
Government" />
<meta name="keywords" content="archives,
information management, public
administration" />
<meta name="AGLS.Function" scheme="AGIFT"
content="recordkeeping standards" />
```



案例集锦

- Simple DC

[example 1](#)

- Qualified DC

[example 2](#)

- [ScreenCam](#) of using DC-dot

<http://www.ukoln.ac.uk/metadata/dcdot/>



Encoding DC in XML ***用XML对DC进行置标***



属性和值的置标

- 置标时属性(*properties*)作为XML的元素而字符串值(*value strings*)就是元素的内容
- 此属性的XML元素的名称必须是一个XML限定名 (QName)

```
<dc: title>Dublin Core in XML</dc: title>
```

- 切记：不要使用如下的结构：

```
<dc: title value="Dublin Core in XML" />
```



DCMES属性的名称

- DCMES中15个元素的属性名称应该全部小写:

```
<dc: title>Dubl in Core in XML</dc: title>
```

- 请不要这样使用:

```
<dc: Title>Dubl in Core in XML</dc: Title>
```



置标时如何重复元素

- 多个字符串值可以通过重复作为属性的XML元素来进行置标

```
<dc: title>First title</dc: title>  
<dc: title>Second title</dc: title>
```



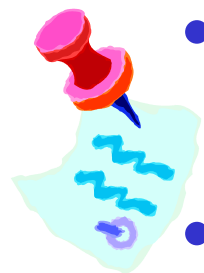
字符串值(Value string)的语种

- 在说明值(*value*)语种的地方，应该用‘xml:lang’属性，如：

```
<dc:subject xml:lang="en">  
  seafood  
</dc:subject>  
<dc:subject xml:lang="fr">  
  fruits de mer  
</dc:subject>
```



作为容器的元素



- 注意，记录(records)应该在一个或多个某种类型的XML容器元素中置标；
- 在这次培训中，不推荐任何容器元素的名称，也不推荐容器元素来源的命名域名称
- 下面的容器元素名称做为候选： <dc>， <dublinCore>， <resource>， <record> and <metadata>



简单DC的XML置标例子

- [example 3](#)



元素限定的XML置标

- 元素限定(*element refinements*)置标方式与其他属性相同
- 比如:

```
<dcterms: available>2002-06</dcterms: available>
```

- 请不要使用下面置标方式的任何一种:

```
<dc: date refinement="available" >  
2002-06</dc: date>
```

```
<dc: date type="available" >2002-06</dc: date>  
<dc: date>
```

```
    <dcterms: available>2002-06  
    </dcterms: available>  
</dc: date>
```



编码体系(*Encoding schemes*)

- 编码体系(*encoding schemes*) 应该使用 XML 元素的 'xsi : type' 属性来置标
- 编码体系的名称应该作为一个属性值, 采用 XML 限定名(QName)的形式:

```
<dc: i denti fi er xsi : type="dcterms: URI ">  
  http: //www. ukol n. ac. uk/  
</dc: i denti fi er>
```



名称的大小写问题

- 元素、元素限定、编码体系(*elements, element refinements and encoding schemes*) 应使用列入如下文档中的名称:

DCMI Metadata Terms

<http://dublincore.org/documents/dcmi-terms/>

... 注意: **15个DCMES**中的元素首字母必须小写...



下面是一些例子

- Qualified DC

[example 4](#)

- DC and IEEE LOM

[example 5](#)

- DC, IMS and ODRL

[example 6](#)

HEALTH WARNING
Examples 5 and 6 may
seriously damage your
interoperability!



Encoding DC in RDF

用RDF对DC置标



什么是RDF?

- 资源描述框架(Resource Description Framework)
- W3C应用元数据时的推荐标准
- 是一个模型，一种句法(syntax(es))
- 应用在Web上时，RDF 通常用XML来编码
- 是语义万维网 (semantic Web)的基础、支撑

W3C - Resource Description Framework (RDF)

<http://www.w3.org/RDF/>



为什么要使用 **RDF**?

- **RDF**提供共享元数据的模型(model)...
- ...共享语义(meaning)
- 元数据可以在相互了解不多或根本不了解的应用之间共享
- 例如一个基于**RDF**的书目应用能够吸收基于**RDF**的地理空间应用的元数据并对其意义有所理解。

...用(X)HTML和XML置标后，软件应用必须能够理解复杂的编码...



DC在RDF中的置标

- DC抽象模型(DC abstract model)可以很好地和RDF模型映射(因为RDF是抽象模型的基础!)
- 基于RDF/XML句法的DC是一种基于XML的RDF模型
- 简单DC在RDF中的置标与我们前面已经涉及的、不用RDF模型的XML置标类似...
- ...但是增加使用 `<rdf: RDF>` 和 `<rdf: Description>` 作为容器元素(container elements)
- [example 7](#)
- 限定DC在RDF中的置标太过复杂了，这儿不涉及



实践举例 – OAI 和 RSS



OAI-PMH

- 全称：OAI Protocol for Metadata Harvesting(用于元数据收割的OAI协议)
- 不同应用之间共享元数据记录的简单协议
- 最新的版本为 2.0
- 基于HTTP, XML, XML Schema 和 XML namespaces
- 允许一个收割器(harvester)远程访问一个资源库中部分或所有的元数据记录



OAI-PMH (2)

- 简单DC是默认(强制)的记录格式
- 支持任何以XML置标的元数据格式(如DC, IEEE LOM, MARC, ODRL, ...)

Open Archives Initiative

<http://www.openarchives.org/>



OAI-PMH的例子

- record from the American Memory repository at the Library of Congress

`http://memory.loc.gov/cgi-bin/oai2_0`

- [example 8](#)
- [ScreenCam](#) of using the '[repository explorer](#)'
- GetRecord for record identifier

`oai:loc1.loc.gov:loc.gmd/g3701p.rr003570`



RSS

- 全称：RDF站点概要(RDF Site Summary)或站点的详细摘要(Rich Site Summary)，甚至是(Really Simple Syndication)
- 至少有三个不同版本(0.91, 1.0 and 2.0)
- 都是基于XML的但不兼容
- 用于Web上共享新闻的简单格式
- RSS 频道(channels) – 一系列的条目(items)
- 通过更新XML文件来更新频道
- RSS客户端根据一定的规则有规律地搜集XML



RSS1.0和DC举例

- 基于RDF的RSS1.0
- 在RSS家族中最具灵活性和可扩展性，但并不是应用最广的
- 可以在频道描述和条目描述中使用DC
- [example 9](#)
- 全文位于:

RDF Site Summary 1.0 Modules: Qualified Dublin Core

<http://web.resource.org/rss/1.0/modules/dcterms/>



我们现在学了哪些？

- DC抽象模型
- 用XHTML对DC进行置标
- 用XML对DC进行置标
- 用RDF/XML对DC进行置标
- 两个实例
 - OAI Protocol for Metadata Harvesting
 - RSS



Questions?

