

# 都柏林核心元数据抽象模型

资源名:	都柏林核心元数据抽象模型
创建者:	<u>Andy Powell</u> UKOLN, University of Bath, UK <u>Mikael Nilsson</u> KMR Group, CID, NADA, KTH (Royal Institute of Technology), Sweden <u>Ambjörn Naeve</u> KMR Group, CID, NADA, KTH (Royal Institute of Technology), Sweden <u>Pete Johnston</u> UKOLN, University of Bath, UK
翻译者:	张春景(上海图书馆数字图书馆研究所), 夏翠娟(华东师范大学信息学系研究生)
发布日期:	2004-12-08
标识符:	<a href="http://www.ukoln.ac.uk/metadata/dcmi/abstract-model/">http://www.ukoln.ac.uk/metadata/dcmi/abstract-model/</a>
替代:	<a href="http://www.ukoln.ac.uk/metadata/dcmi/abstract-model/2004-11-24/">http://www.ukoln.ac.uk/metadata/dcmi/abstract-model/2004-11-24/</a>
被替代:	无
最新版本:	<a href="http://dublincore.org/documents/abstract-model/">http://dublincore.org/documents/abstract-model/</a>
文档状态:	DCMI 工作草案 <sup>1</sup> 。
文档描述:	此文档描述了都柏林核心元数据记录的抽象模型。

## 目录

1. 引言
  2. DCMI 抽象模型
  3. 描述、描述集及记录
  4. 值
  5. 向上兼容原则
  6. 置标指南
  7. 术语
- 参考文献  
致谢
- 附录 A-关于结构值的提示  
附录 B-抽象模型与 RDF  
附录 C-抽象模型与 XML  
附录 D-抽象模型与 XHTML

---

<sup>1</sup> <http://dublincore.org/documents/#workingdrafts>

# 1. 引言

本文档详细说明了DCMI元数据描述的抽象模型[DCMI]，主要目的是提供一个参考模型以便对各类专门DC编码规则进行比较。一个好的参考模型应该独立于任何特定的编码语法，并能对需编码的对象的属性描述有更深入的理解，从而有助于不同编码语法之间更好地映射和翻译。

## 2. DCMI 抽象模型

被 DCMI 元数据所描述的 *资源* 如下：

- 每个 *资源* (*resource*) 具有零个或多个 *属性/值对* (*property/value pairs*) ；
- 每个 *属性/值对* (*property/value pairs*) 由一个 *属性* (*property*) 和一个 *值* (*value*) 组成；
- 每个 *值* 本身是一个 *资源* (即：用来描述 *资源* (*resource*)，与属性相关的物理或概念实体。)；
- 每个 *资源* (*resource*) 可以是一个或多个 *类* (*classes*) 中的成员； (注：作为 *属性* (*property*) *值* (*value*) 的资源所在的类 (*class*) 常被称为 *编码体系词表* (*vocabulary encoding scheme*) ) ；
- 每个 *属性* (*property*) 和 *类* (*class*) 均具有其被声明的语义；
- 每个 *类* (*class*) 通过限定 (子类) 关系与一个或多个其它 *类* (*class*) 相关 (当两个类共享部分 *语义* (*semantics*) 时，所有属于 *子类* (*sub-class*) 的 *资源* (*resource*) 同时也是另一个相关 *类* (*class*) 的成员；
- 每个 *属性* (*property*) 只能与一个其它 *属性* (*property*) 通过限定关系 (子属性) 相关 (当两种属性共享部分 *语义* (*semantics*) 时，*子属性* (*sub-property*) 的有效 *值* (*value*) 也是相关 *属性* (*property*) 的有效 *值* (*value*) ) 。

DCMI 元数据描述的抽象模型如下：

- 一条 *描述* (*description*) 是由一个或多个 *陈述* (*statements*) (该陈述仅与一个且唯一一个 *资源* (*resource*) 有关)，以及零个或一个 *资源的 URI* (*resource URI*) 组成 (URI 用来标识所描述的 *资源* (*resource*) ) ；
- 每个 *陈述* (*statements*) 由一个 *属性 URI* (*property URI*) (这里的 URI 用于标识一个 *属性* (*property*) )，零个或一个 *值 URI* (*value URI*) (这里的 URI 用于标识 *属性* (*property*) 的 *值* (*value*) )，零个或一个 *编码体系 URI* (*encoding scheme URI*) (这里的 URI 标识 *值* (*value*) 的 *类* (*class*) )，零个或多个 *值的表述* (*value representations*) 组成；
- 每个 *属性* (*property*) 都是被描述 *资源* (*resource*) 的一项特性；
- 每个 *属性 URI* (*property URI*) 可以在多个 *陈述* (*statements*) 中重复；
- *值的表述* (*value representation*) 可以是 *字符串值* (*value string*)、*复合值* (*rich value*) 或 *相关描述* (*related description*) 等形式；
- 每个 *字符串值* (*value string*) 都是一个简单的、人类可读的字符串，用以表示 *属性的值* (*value of the property*) ；

- 每个字符串值(value string)可以有相应的编码体系 URI( encoding scheme URI), 用来标识一个语法编码体系(syntax encoding scheme);
- 每个字符串值(value string)可以有相应的字符串语种(value string language), 它是一个 ISO 语种标记(例如, en-GB);
- 每个复合值(rich value)是一些标记文本、图像、视频、音频等, 或者它们的组合, 表示作为属性(property)值(value)的资源(resource);
- 每条相关描述(related description)都是一个用来描述属性(property)值(value)的资源(resource)。

上文中用斜体显示的术语, 会在下面的“术语”一节中定义。关于模型的许多方面需要注意:

- 一条“相关描述”描述一个相关的资源, 因此并非是“描述”的一部分, 例如, 当一个人是所描述资源的创建者时, 一条相关描述可以提供关于这个“人”的元数据。
- 在某些语境中, 语法编码体系也可以认为是某种“数据类型”
- 在 DCMI 元数据描述中, 所描述资源的类通常由 DC 类型属性的值来描述。

上述 DCMI 资源和描述的抽象模型可以由 UML (统一建模语言) 表示如下:

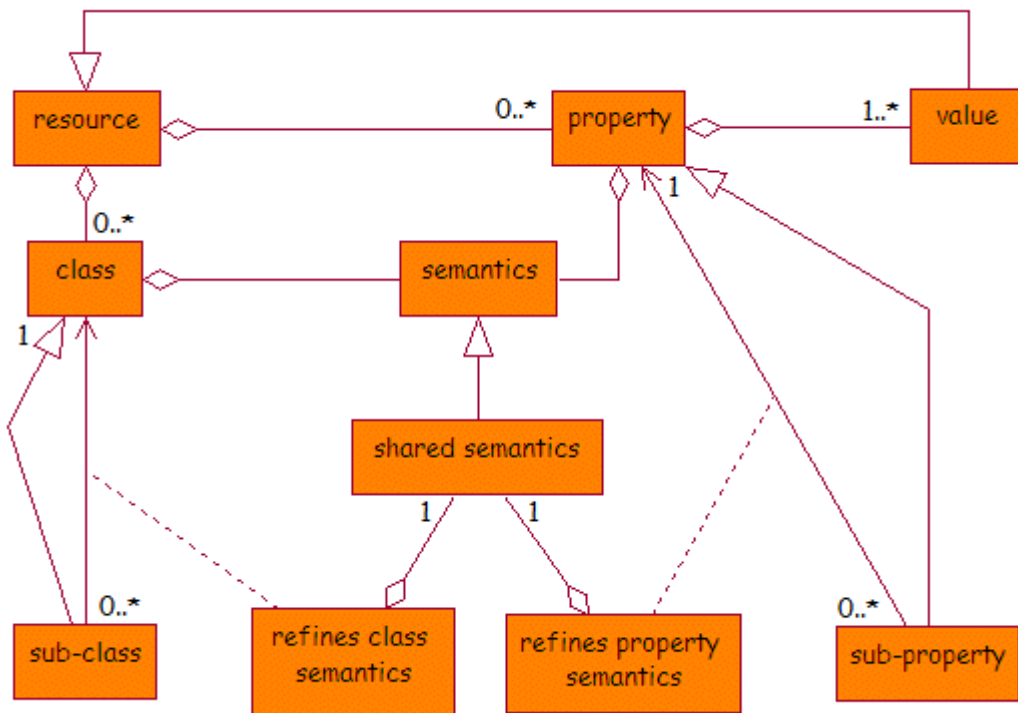


图 1 DCMI 资源模型

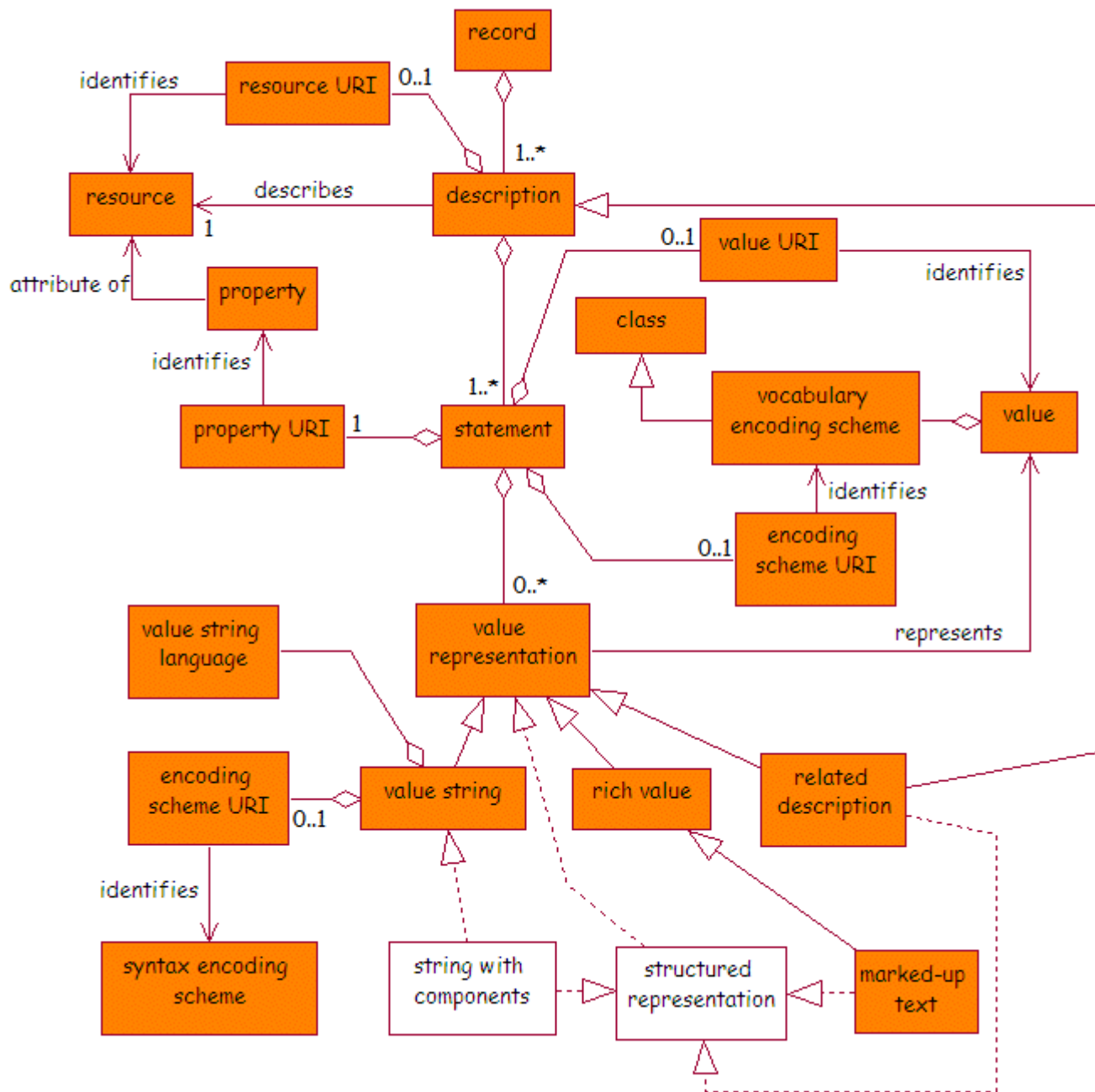


图 2 DCMI 描述模型

不熟悉 UML 类图的读者应该注意：尾部带箭头的直线意思是“是”（例如，“词表编码体系是一个类”），以钻石形状开始的线段的意思是“包含”或“有”（例如，“一个陈述包含一个属性 URI”）。其他关系也有适当的标记。在抽象模型的文本描述中，白色方框中的类没有明确地被提到，但在附录 A 中有相应讨论。注意这里采用 UML 进行建模，并不意味着可以依靠它来进行 DCMI 元数据应用软件的开发。

### 3. 记录和描述

上文所述的抽象模型表明：一条描述中的每个属性必须是所描述资源的一个特性。这就是通常所说的“1:1 原则”，这个原则就是：一条元数据描述仅描述一个资源。

然而，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，这样的描述集往往指的是元数据记录。例如：一条元数据记录可以由关于一幅画的描述和关于其画家的描述共同组成。此外，一条记录也将包含关于该记录自身的描述（有时被称为“管理性元数据”或者“元元数据”），此类情况是非常多见的。

本文档将一条 DCMI 元数据记录作如下定义：

- 一条 DCMI 元数据记录是一条或多条描述的集合，这些描述是关于一个或多个相关资源的，这些资源根据某个 DCMI 置标指南实例化（这些置标指南有：XHTML meta tags, XML, RDF/XML, 等等） [DCMI-ENCODINGS]。

## 4. 值

一个 DCMI 元数据值是物理的或者概念的实体，当描述一个资源时，这个实体就成为该资源的属性。例如：DC. Creator 属性的值是一个“人”、组织、或服务——一个物理实体。DC. Date 属性的值是一个时间点——一个概念实体。DC. Coverage 属性的值可以是一个地区或国家的地理区域——一个物理实体。DC. Subject 属性的值可以是一个概念——一个概念实体，或者一个物理对象或人——一个物理实体。每个这样的实体都是一个资源。

值可以用 URI 值来标识；可以用一个或者多个字符串值和/或复合值来表示；可以有一些“相关描述”——但值是一个资源。

## 5. 向上兼容原则

将一条修饰 DC 记录解析为一条简单 DC 记录的过程通常指的就是“向上兼容”。向上兼容的处理可以分成两类：属性向上兼容和值向上兼容。而且，每种方式都可以用“预设（informed）”和“非预设（uninformed）”两种方法之一来实现。“预设”方式指软件在运行向上兼容的算法时，有关特定应用的 DCMI 元素属性关系和值的规则已经嵌入此软件，“非预设”方式指系统没有预先建立有关属性关系和值的规则。

如上所述，向上兼容的具体做法可以大致由下表所示：

	元素（属性）向上兼容	值向上兼容
<b>Uninformed</b>	忽略任何不属于 DC 元数据元素集的属性。	使用 URI 值（如果存在）或字符串值作为新字符串值。
<b>Informed</b>	递归地解决子属性关系，直到 DC 元数据元素集 15 个属性中的一个。否则就忽略。	用相关描述或字符串值的知识来构造新的字符串值。

并且在所有情况下，向上兼容算法都应该：

- 忽略任何相关描述和复合值
- 忽略任何编码体系 URI.

注意软件应该使用以 RDF schema [DC-RDFS] 和 DC XML 命名空间 [DC-NAMESPACES] 表示的 DCMI 术语声明，以便自动解析子属性。

## 6. 置标指南

具体的置标指南（HTML meta tags, XML, RDF/XML 等）无需包括上述抽象模型的所有内容。然而，DCMI 的置标指南应该包括 DCMI 抽象模型并且说明该模型的哪些部分置标哪些部分不置标。置标指南尤其应该做出规定：一条有复合值和相关描述的描述是嵌入记录中，还是在另一条记录中单独进行置标并用 URI 链接。

文末的附录 B、C、D 比较了抽象模型采用 RDF/XML, XML 和 XHTML 置标不同之处（译文略）。

## 7. 术语

本文档用到以下术语：

### 资源 (*resource*)

资源是任何可以标识的东西。常见的例子有电子文档，图像，服务（例如，“洛杉矶今天的天气预报”），还有其他资源的集合。并非所有的资源都是网上可检索的；例如，人，机构，还有图书馆里装订成册的书都可以被认为是资源。

### 资源 URI (*resource URI*)

一个资源 URI 是用于标识单个资源的 URI

### 属性 (*property*)

属性是用来描述资源某个特定的方面、特征、特性或者关系。

### 属性 URI (*property URI*)

一个属性 URI 是用于标识单个属性的 URI。

### 元素 (*element*)

在 DCMI 中，元素通常被用作“属性”的同义字。然而，应该注意的是元素这个词也往往用于指 XML 文档中的一个结构化的标记组件。

### 元素限定 (*element refinement*)

元素限定是指资源的一个属性，与一个特定 DCMI “属性”有同样的意思，但是相对来说具有更窄的语义。由于元素限定和元素一样也是资源的属性，所以元素限定可以独立于它所限定的属性用在元数据记录中。在 DCMI 的实际操作中，元素限定只能有一个 DCMI 父属性。

### 描述 (*description*)

一条描述由一条或多条陈述组成，这些描述是关于一个资源的，且只能是关于一个资源的。

### 陈述 (*statement*)

一条陈述由一个属性 URI（这里的 URI 用于标识一个属性），零个或一个值 URI（这里的 URI 用于标识属性的值），零个或一个编码体系 URI（这里的 URI 标识值的类），零个或多个值的表述组成。

### 记录 (*record*)

一条记录是一条或多条描述的集合，这些描述是关于一个或多个相关资源的，根据某种 DCMI 的置标指南（XHTML meta tags, XML, RDF/XML 等）[DCMI-ENCODINGS]实例化。

### 值 (*value*)

一个值是物理的或概念的实体，当它用于描述一个资源时，就成了某个属性的值。

### **值 URI (*value URI*)**

一个值 URI 是用于标识属性值的 URI。

### **值的表示 (*value representation*)**

一个值的表示代表该值，例如是该值的表示形式。

### **字符串值 (*value string*)**

一个字符串值是用来表示元素或元素修饰值的简单字符串。一般来说，字符串值不包含任何标记文本。

### **字符串值语种 (*value string language*)**

字符串值语种说明了字符串值所用的语种。

### **编码体系词表 (*vocabulary encoding scheme*)**

一个编码体系词表是一个“类”，表示属性值取自于某个受控词表（或概念空间），如国会图书馆主题词表

### **编码体系语法 (*syntax encoding scheme*)**

一个编码体系语法表示字符串值使用某种格式符号的格式，如“2000-01-01”是日期的标准表示。

### **编码体系 (*encoding scheme*)**

编码体系是编码体系词表和编码体系语法的通称。

### **编码体系 URI (*encoding scheme URI*)**

编码体系 URI 是指标识一个编码体系的 URI。对于 DCMI 推荐的所有编码体系，URI 由编码体系的名称和 DCTerms 的命名空间 URI 连接而成。

### **标记文本 (*marked-up text*)**

标记文本是一个字符串。该字符串包括 HTML, XML 或其他标记文本（例如 TeX），还有那些表示元素或者元素修饰词的值的字符串。

### **复合值 (*rich value*)**

一些标记文本，图像，视频和一些音频等（或者这些的组合）。它们都表示属性的值。

### **相关描述 (*related description*)**

一条相关描述是一条资源的描述，用于描述所描述资源的相关资源。

### **修饰词 (*qualifier*)**

修饰词目前具体指元素修饰词和编码体系的通称。

### **结构化值 (*structured value*)**

结构化值可以是下列其中之一：

- 一个包含机器可解析的组件的字符串值，并且该字符串值有一个相关的语法编码体系，该编码体系说明了这些组件是怎样在该字符串内置标的。
- 一些标记文本。
- 一条相关描述。

## **参考文献**

### **DCMI**

Dublin Core Metadata Initiative

[<http://dublincore.org/>](http://dublincore.org/)

## **UML**

The Unified Modeling Language User Guide

Grady Booch, James Rumbaugh and Ivar Jacobson, Addison-Wesley, 1998

## **DCTERMS**

DCMI Metadata Terms

[<http://dublincore.org/documents/dcmi-terms/>](http://dublincore.org/documents/dcmi-terms/)

## **DCMES**

Dublin Core Metadata Element Set, Version 1.1: Reference Description

[<http://dublincore.org/documents/dces/>](http://dublincore.org/documents/dces/)

## **DCMI-ENCODINGS**

DCMI Encoding Guidelines

[<http://dublincore.org/resources/expressions/>](http://dublincore.org/resources/expressions/)

## **DCAP**

DCMI Usage Board Review of Application Profiles

[<http://dublincore.org/usage/documents/profiles/>](http://dublincore.org/usage/documents/profiles/)

## **DC-RDFS**

DCMI term declarations represented in RDF schema language

[<http://dublincore.org/schemas/rdfs/>](http://dublincore.org/schemas/rdfs/)

## **DC-NAMESPACES**

Namespace Policy for the Dublin Core Metadata Initiative (DCMI)

[<http://dublincore.org/documents/dcmi-namespace/>](http://dublincore.org/documents/dcmi-namespace/)

## **致谢**

感谢 Tom Baker，感谢 DC 应用委员会的成员们，感谢 DC 体系结构工作组的成员们，感谢他们为本文档的最近版本所提供的意见。