



Übersetzung des
Dublin Core Metadata Initiative Abstract Model (DCAM)

Identifizier: [urn:nbn:de:101-2017041901](https://nbn-resolving.org/urn:nbn:de:101-2017041901)
Title: Übersetzung des Dublin Core Metadata Initiative Abstract Model (DCAM)
Contributor: Thomas Baker, Stefanie Rühle
Date issued: 11.11.2008

Zu dieser Übersetzung

Diese Übersetzung wurde erstellt, um deutschsprachigen Lesern den Zugang zu dem DCMI Abstract Model zu erleichtern. Ziel war nicht, eine parallele Terminologie in deutscher Sprache aufzustellen. Aus diesem Grund wurden die im englischen Original verwendete Terminologie so weit als möglich beibehalten bzw. wurde auf diese hingewiesen (i. d. R. kursiv und in Klammern gesetzt).

Falls manche Formulierung etwas künstlich wirken, so sei bemerkt, dass die hier verwendeten Formulierungen auch im Englischen nicht üblich sind. Dies liegt wohl in der Natur der Aufgabe, die darin bestand, ein kleines aber in sich schlüssiges Gedankengebäude aufzustellen.

INHALT

1. EINFÜHRUNG	4
2. DCMI ABSTRACT MODEL	5
2.1. Das DCMI-Ressourcenmodell	5
2.2. Das DCMI-Beschreibungssatzmodell	6
2.3. Das DCMI-Vokabular-Modell	8
2.4. Anmerkungen	9
3. BESCHREIBUNGEN, BESCHREIBUNGSSÄTZE UND DATENSÄTZE	10
4. WERTE	10
5. SEMANTIK DES DCMI-ABSTRAKTMODELLS	11
6. CODIERUNGS-RICHTLINIEN	12
7. TERMINOLOGIE	12
8. BEZIEHUNG ZU DEN ÄLTEREN "DCMI GRAMMATISCHEN PRINZIPIEN"	16
QUELLENANGABEN	18

1. Einführung

Dieses Dokument beschreibt das Abstraktmodell für Dublin-Core-Metadaten. Ziel des Dokuments ist es vor allem, die Elemente und Strukturen, die in Dublin-Core-Metadaten verwendet werden, zu benennen. Das Dokument definiert die verwendeten Elemente und beschreibt, wie sie miteinander kombiniert werden, um Informationsstrukturen zu bilden. Es stellt ein von jeglicher besonderen Codierungssyntax unabhängiges Informationsmodell dar. Ein solches Informationsmodell macht es uns möglich, die Beschreibungen, die wir codieren wollen, besser zu verstehen und erleichtert die Entwicklung besserer Mappings und syntaxübergreifender Datenkonvertierungen.

Dieses Dokument richtet sich in erster Linie an Entwickler von Softwareanwendungen, die Dublin-Core-Metadaten unterstützen, an Personen, die neue syntaktische Codierungsrichtlinien für Dublin-Core-Metadaten entwickeln und an Personen, die Metadatenprofile entwickeln, die auf DCMI- oder anderen kompatibelen Vokabularen basieren.

Das DCMI-Abstraktmodell basiert auf der Arbeit des World Wide Web Consortium (W3C) am Resource Description Framework (RDF)^{[1][2]}. Die Verwendung von Konzepten aus RDF wird unten im Abschnitt 5 zusammengefasst.

Das DCMI-Abstraktmodell wird hier mit UML-Klassen-Diagrammen dargestellt^[3]. Für Leser, die solche UML-Klassen-Diagramme nicht kennen, eine kurze Anleitung: Linien, die in einem Maßpfeil enden, werden als 'ist' oder 'ist eine' gelesen (z.B. "*value* ist eine *resource*"). Linien, die mit einer Raute beginnen, werden als 'hat' oder 'hat eine' gelesen (z.B. "*statement* hat einen *property URI*"). Andere Beziehungen werden entsprechend gekennzeichnet. Die *kursiv geschriebenen* Wörter und Phrasen in diesem Dokument werden im Abschnitt 7 ("Terminologie") definiert.

Wir danken Dan Brickley, Rachel Heery, Alistair Miles, Sarah Pulis, den Mitgliedern des DCMI Usage Board und den Mitgliedern der DCMI Architecture Community für ihr Feedback zu den vorangegangenen Versionen dieses Dokuments.

Der Übersetzung liegt die folgende Ausgabe zugrunde:

Titel: DCMI Abstract Model

URL: <http://dublincore.org/documents/2007/06/04/abstract-model/>

Verfasser: Andy Powell, Mikael Nilsson, Ambjörn Naeve, Pete Johnston und Thomas Baker

veröffentlicht am: 04.06.2007

2. DCMI Abstract Model

2.1. Das DCMI-Ressourcenmodell

Das Abstraktmodell für Ressourcen (*resource*), die von Beschreibungen (*description*) beschrieben werden, sieht folgendermaßen aus:

- Jede beschriebene Ressource (*described resource*) wird mit einem oder mehreren Paaren aus Eigenschaften und Werten (*property-value pair*) beschrieben.
- Jedes dieser Paare besteht aus einer Eigenschaft (*property*) und einem Wert (*value*).
- Jeder Wert (*value*) ist eine Ressource (*resource*) - eine physische, digitale oder begriffliche Entität oder ein Literal (*literal*), das mit einer Eigenschaft (*property*) assoziiert wird, wenn ein Paar aus einer Eigenschaft und einem Wert (*property-value pair*) benutzt wird, um eine Ressource (*resource*) zu beschreiben. Demgemäß ist jeder Wert entweder ein literaler Wert (*literal value*) oder kein literaler Wert (*non-literal value*):
 - Ein literaler Wert (*literal value*) ist ein Wert, der ein Literal (*literal*) ist.
 - Ein nicht-literaler Wert (*non-literal value*) ist ein Wert, der eine physische, digitale oder begriffliche Entität ist.
- Ein nach RDF definiertes Literal (*literal*) ist eine Entität, die, um eine Ressource (*resource*) zu bezeichnen eine Unicode-Zeichenkette sowie optional die Sprache und/oder den Datentyp verwendet.

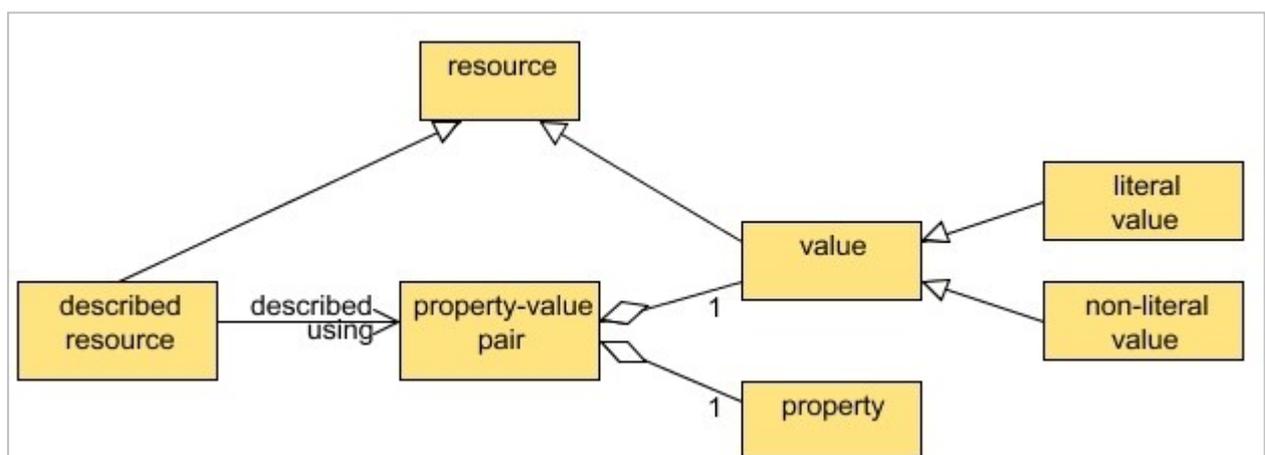


Abbildung 1: das DCMI-Ressourcenmodell

2.2. Das DCMI-Beschreibungssatzmodell

Das Abstraktmodell für einen DC-Metadaten-Beschreibungssatz (*description set*) sieht folgendermaßen aus:

- Ein Beschreibungssatz (*description set*) besteht aus einer oder mehreren Beschreibungen (*descriptions*), von denen jede eine einzige Ressource (*resource*) beschreibt.
- Eine Beschreibung (*description*) besteht aus einer oder mehreren Aussagen (*statements*) über eine, und nur eine Ressource (*resource*) und Null oder einem URI der beschriebenen Ressource (*described resource URI*), der die beschriebene Ressource (*described resource*) identifiziert.
- Jede Aussage (*statement*) realisiert ein Paar aus einer Eigenschaft und einem Wert (*property-value pair*) und besteht aus dem URI der Eigenschaft (*property URI*), der die Eigenschaft (*property*) identifiziert und einem Platzhalter für den Wert (*value surrogate*).
- Der Wert ist entweder durch einen literalen Platzhalter (*literal value surrogate*) oder einen nicht-literalen Platzhalter (*non-literal value surrogate*) vertreten.
 - Ein literaler Platzhalter (*literal value surrogate*) ist der Platzhalter für einen literalen Wert (*literal value*) und besteht aus genau einer Zeichenkette (*value string*). Die Zeichenkette (*value string*) ist ein Literal (*literal*), das den literalen Wert (*literal value*) codiert.
 - Ein nicht-literaler Platzhalter (*non-literal value surrogate*) steht für einen nicht-literalen Wert (*non-literal value*) und besteht aus Null oder dem URI des Werts (*value URI*), (einem URI, der den nicht-literalen Wert, der mit der Eigenschaft assoziiert ist, identifiziert), Null oder dem URI des Semantischen Codierungsschemas (*vocabulary encoding scheme URI*) (einem URI, der das Semantische Codierungsschema, dem der nicht-liternale Wert angehört, identifiziert), und Null oder mehreren Zeichenketten (*value strings*). Jede Zeichenkette (*value string*) ist ein Literal (*literal*), das den nicht-literalen Wert (*non-literal value*) repräsentiert.
- Eine Zeichenkette (*value string*) ist entweder eine einfache Zeichenkette (*plain value string*) oder eine typisierte Zeichenkette (*typed value string*).
 - Eine einfache Zeichenkette (*plain value string*) kann verwendet werden mit einer Angabe zur Sprache der Zeichenkette (*value string language*). Dabei handelt sich um eine ISO-normierte Sprachangabe (z. B. "en-GB"). Einfache Zeichenketten (*plain value string*) sollen von Menschen lesbar sein.
 - Eine typisierte Zeichenkette (*typed value string*) ist verbunden mit dem URI des syntaktischen Codierungsschemas (*syntax encodings scheme URI*), der das syntaktische Codierungsschema (*syntax encoding scheme*) identifiziert.

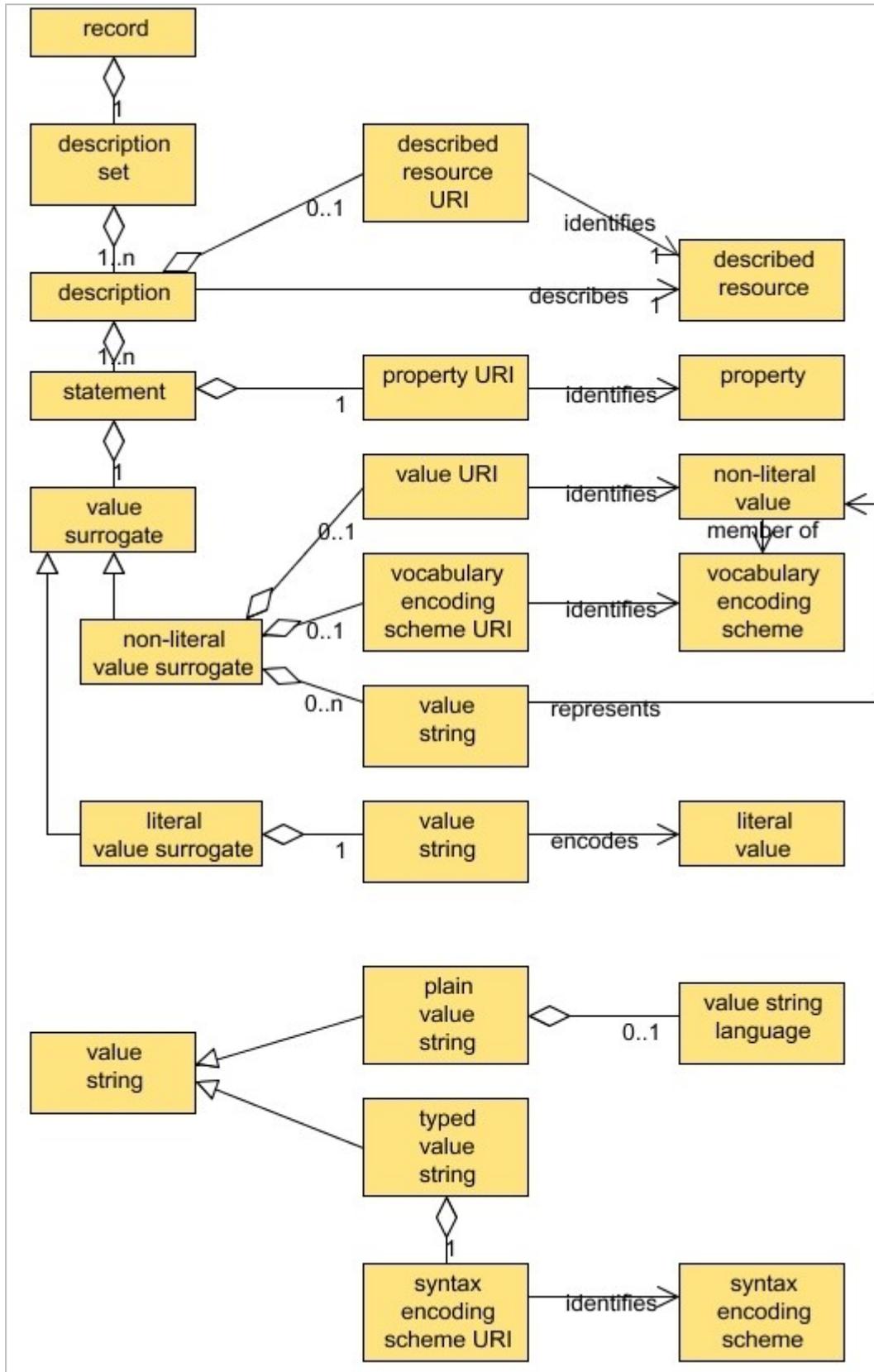


Abbildung 2: das DCMI-Beschreibungssatzmodell

2.3. Das DCMI-Vokabular-Modell

Das Abstraktmodell für Vokabulare (*vocabularies*), die in Beschreibungen (*descriptions*) mit Dublin-Core-Metadaten verwendet werden, sieht folgendermaßen aus:

- Ein Vokabular (*vocabulary*) besteht aus einem oder mehreren Termen (*terms*). Jeder Term (*term*) ist Teil eines oder mehrerer Vokabulare (*vocabularies*).
- Ein Term (*term*) ist entweder eine Eigenschaft (*property*) bzw. ein Element (*element*), eine Klasse (*class*), ein semantisches Codierungsschema (*vocabulary encoding scheme*) oder ein syntaktisches Codierungsschema (*syntax encodings scheme*).
- Jede Eigenschaft (*property*) kann zu der Domain einer oder mehrerer Klassen (*classes*) gehören (*has domain*). Wenn eine Eigenschaft (*property*) zu der Domain einer Klasse (*class*) gehört und die Eigenschaft (*property*) ist Teil eines Paares aus Eigenschaft und Wert (*property/value pair*), dann ist der Wert (*value*) ein Objekt dieser Klasse (*class*).
- Jede Eigenschaft (*property*) kann zu dem Range einer oder mehrerer Klassen (*classes*) gehören (*has range*). Wenn eine Eigenschaft (*property*) zu dem Range einer Klasse (*class*) gehört und die Eigenschaft (*property*) ist Teil eines Paares aus Eigenschaft und Wert (*property/value pair*), dann ist der Wert (*value*) ein Objekt dieser Klasse (*class*).
- Jede Ressource (*resource*) kann eine Instanz (*instance of*) einer oder mehrerer Klassen (*classes*) sein.
- Jede Ressource (*resource*) kann zu einem oder mehreren semantischen Codierungsschemas (*vocabulary encoding scheme*) gehören (*member of*).
- Jede Klasse (*class*) kann Subklasse einer oder mehrerer Klassen (*classes*) sein (*sub-class of*) (wobei die beiden Klassen derart definiert sind, dass jede Ressource, die Instanz der Subklasse ist auch eine Instanz der Klasse ist, der die Subklasse untergeordnet ist).
- Jede Eigenschaft (*property*) kann Sub-Eigenschaft einer oder mehrerer Eigenschaften (*properties*) sein (*sub-property of*). Ist dies der Fall, so werden die beiden Eigenschaften (*properties*) derart definiert, dass immer wenn die Sub-Eigenschaft (*sub-property*) Teil eines Paares aus Eigenschaft und Wert (*property/value pair*) ist, die Ressource (*resource*), die mit diesem Paar beschrieben wird, auch mit einem zweiten Paar aus Eigenschaft und Wert (*property/value pair*) beschrieben wird, das aus der Eigenschaft (*property*), der die Sub-Eigenschaft untergeordnet ist, und deren Wert (*value*) besteht.
- Jedes syntaktische Codierungsschema (*syntax encoding scheme*) ist eine Klasse (*class*), die aus Literalen (*literals*) besteht.

Beachte, dass der Begriff "Vokabular" (*vocabulary*) hier verwendet wird, um einen Satz aus einem oder mehreren Termen zu bezeichnen - einen Satz, der sich aus Eigenschaften (*properties*) bzw. Elementen (*elements*), Klassen (*classes*),

Semantischen Codierungsschemas (vocabulary encoding schemes) und/oder syntaktischen Codierungsschemas (syntax encoding schemes) zusammensetzt.

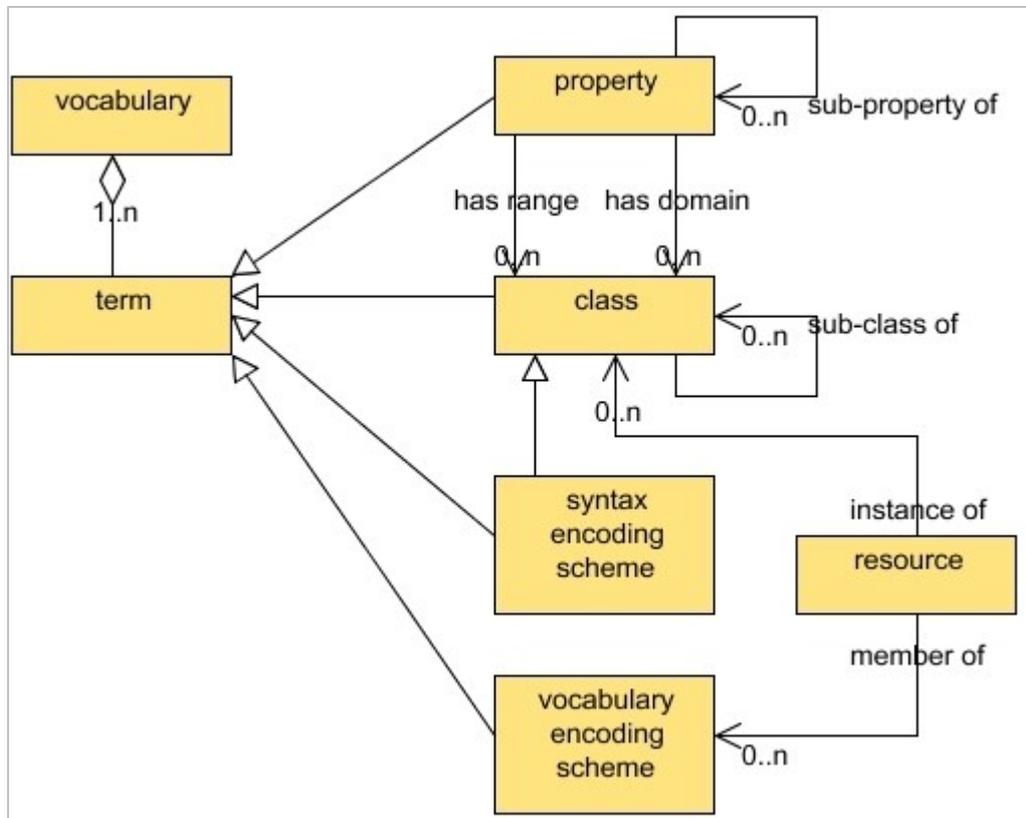


Abbildung 3: das DCMI-Vokabularmodell

2.4. Anmerkungen

Folgende Hinweise zu dem Modell sind zu beachten:

- Jeder nicht-literale Wert (*non-literal value*) kann eine Ressource (*resource*) sein, die in einer separaten Beschreibung (*description*) innerhalb ein- und desselben Beschreibungssatzes beschrieben wird – so kann z. B. eine separate Beschreibung die Metadaten zu einer Person beinhalten, die der Verfasser der beschriebenen Ressource ist. Ein literaler Wert (*literal value*) kann dagegen keine in einer separaten Beschreibung beschriebene Ressource sein.
- Das DCMI-Beschreibungssatzmodell verfügt über keinen expliziten Mechanismus, um die Klassen (*classes*) einer beschriebenen Ressource (*described resource*) kenntlich zu machen. Die Klassen (*classes*) einer beschriebenen Ressource (*described resource*) können entweder explizit durch eine oder mehrere Aussagen (*statements*) in der Beschreibung

(*description*) angegeben werden oder implizit von der Domain (*domain*) der Eigenschaften (*properties*), die in der Beschreibung (*description*) verwendet werden, abgeleitet werden.

- Das DCMI-Beschreibungssatzmodell macht die Unterscheidung zwischen literalen Werten (*literal values*) und nicht-literalen Werten (*non-literal values*) kenntlich, indem in den Aussagen (*statements*) literale Platzhalter (*literal value surrogates*) oder nicht-literale Platzhalter (*non-literal value surrogates*) verwendet werden. Bei nicht-literalen Werten (*non-literal values*) verfügt das DCMI-Beschreibungssatzmodell über keinen expliziten Mechanismus, um außerdem die Klassen (*classes*) der Werte (*value*) kenntlich zu machen. Die Klassen (*classes*) jedes möglichen nicht-literalen Wertes (*non-literal value*) können entweder explizit angegeben werden (durch eine oder mehrere Aussagen (*statements*) in einer separaten Beschreibung (*description*) dieses Werts) oder implizit von dem Range (*range*) der Eigenschaft (*property*) abgeleitet werden. Bei literalen Werten (*literal value*) können die Klassen (*classes*) eines Wertes (*value*) entweder explizit angegeben werden (indem für die Zeichenkette (*value string*) ein syntaktisches Codierungsschema (*syntax encoding scheme*) verwendet wird) oder implizit von dem Range (*range*) der Eigenschaft (*property*) abgeleitet werden.
- XML-Inhalte in einer Zeichenkette (*value string*) werden kenntlich gemacht, indem eine typisierte Zeichenkette (*typed value string*) zusammen mit der URI eines syntaktischen Codierungsschemas (*syntax encoding scheme URI*) aus <http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral> verwendet wird.

3. Beschreibungen, Beschreibungssätze und Datensätze

Das vorliegende Abstract Model schreibt vor, dass jede Beschreibung (*description*) mit Metadaten eine, und nur eine Ressource (*resource*) beschreibt. Dieses Prinzip wird das "Eins-zu-Eins-Prinzip" ("one-to-one principle") genannt.

Allerdings basieren Metadaten-Anwendungen i. d. R. auf lose gruppierten Beschreibungssätzen (*description sets*) (wobei die beschriebenen Ressourcen (*described resources*) normalerweise irgendwie miteinander in Beziehung stehen). Ein Beschreibungssatz (*description set*) kann beispielsweise sowohl die Beschreibung (*description*) eines Bildes als auch die des Künstlers enthalten. Außerdem kommt es häufig vor, dass ein Beschreibungssatz (*description set*) auch die Beschreibung (*description*) des Beschreibungssatzes (*description set*) selbst enthält (zuweilen auch als "administrative Metadaten" oder "Meta-Metadaten" bezeichnet).

Beschreibungssätze (*description sets*) dienen dem Austausch zwischen Software-Anwendungen und werden in Form von Meta-Datensätzen (*records*) gemäß den DCMI-Codierungs-Richtlinien (z.B. für XHTML meta tags, XML und RDF/XML) ^[4] realisiert.

4. Werte

Ein Wert (*value*) im Sinne der Dublin-Core-Metadaten ist eine physische, digitale oder begriffliche Entität oder ein Literal (*literal*), das mit einer Eigenschaft (*property*) assoziiert wird, wenn ein Paar aus einer Eigenschaft und einem Wert (*property value pair*) benutzt wird, um eine Ressource (*resource*) zu beschreiben. So ist zum Beispiel der Wert (*value*), der mit der Dublin Core-Eigenschaft (*property*) Creator assoziiert wird, eine Person, eine Organisation oder ein Dienst - also eine physische Entität. Der Wert (*value*), der mit der Dublin Core-Eigenschaft (*property*) Date assoziiert wird, ist ein Zeitpunkt (oder eine Zeitspanne) – also eine begriffliche Entität. Der Wert (*value*), der mit der Dublin Core-Eigenschaft (*property*) Coverage assoziiert wird, ist ein geographisches Gebiet oder ein Land – also eine physische Entität. Der Wert (*value*), der mit der Dublin Core-Eigenschaft (*property*) Subject assoziiert wird, ist ein Begriff (also eine begriffliche Entität) oder ein physisches Objekt oder eine Person (also eine physische Entität). Der Wert (*value*), der mit der FOAF-Eigenschaft (*property*) Name assoziiert wird, ist ein Literal (*literal*). Jede dieser Entitäten ist eine Ressource (*resource*).

5. Semantik des DCMI-Abstraktmodells

Beachte, dass dieses Dokument keine explizite Definition der formalen Semantik des DCMI Abstrakt-Modells vornimmt. Vielmehr wird die formale Semantik durch den Verweis auf RDF und das RDF Schema^[5] definiert. Analogien zwischen Begriffen des DCMI Abstract Model und den entsprechenden Begriffen bei RDF werden in der folgenden Tabelle angegeben:

DCAM	RDF/RDFS
<i>resource</i>	Klasse: http://www.w3.org/2000/01/rdfschema#Resource
<i>property</i> oder <i>element</i>	Klasse: http://www.w3.org/1999/02/22-rdfsyntaxns#Property
<i>class</i>	Klasse: http://www.w3.org/2000/01/rdf-schema#Class
<i>syntax encoding scheme</i>	Klasse: http://www.w3.org/2000/01/rdf-schema#Datatype
<i>has domain</i>	Eigenschaft: http://www.w3.org/2000/01/rdf-schema#domain
<i>has range</i>	Eigenschaft: http://www.w3.org/2000/01/rdf-schema#range
<i>sub-property of</i>	Eigenschaft: http://www.w3.org/2000/01/rdf-schema#subPropertyOf
<i>sub-class of</i>	Eigenschaft: http://www.w3.org/2000/01/rdf-schema#subClassOf
<i>plain value string</i>)	Plain literal. Siehe http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal
<i>typed value string</i>)	Typed literal. Siehe http://www.w3.org/TR/rdf-concepts/#dfn-typed-literal

Tabelle 1: Semantik des DCMI-Abstraktmodells

Zusammen mit dem DCMI-Dokument "Expressing Dublin Core metadata using the Resource Description Framework" (RDF)^[6] sind diese Analogien die Basis der formalen Semantik des DCMI Abstract Model. Auf diese Semantik tiefer einzugehen liegt allerdings außerhalb des Rahmens des vorliegenden Dokuments.

6. Codierungs-Richtlinien

Spezielle Codierungs-Richtlinien (z.B. für HTML meta tags, XML oder RDF/XML u. a. DCMI Encoding Guidelines^[4]) müssen nicht unbedingt alle der oben beschriebenen Aspekte des Abstraktmodells abbilden. Sie sollten jedoch auf das DCMI Abstrac Model verweisen und kennzeichnen, welche Teile des Modells abgebildet werden und welche nicht.

Codierungs-Richtlinien sollten angeben, wie ein nicht-literaler Wert (*non-literal value*) in einer separaten Beschreibung (*description*) als beschriebene Ressource (*described resource*) behandelt werden kann, wenn der nicht-literale Platzhalter des Werts (*non-literal value surrogate*) keine URI (*value URI*) zur Verfügung stellt.

7. Terminologie

Dieses Dokument verwendet die folgenden Begriffe:

class

Eine Gruppe, die aus Elementen besteht, die bestimmte Attribute, Verhaltensweisen, Beziehungen oder Semantik gemeinsam haben; eine Art Kategorie.

described resource

Eine Ressource (*resource*), die mittels einer Beschreibung (*description*) beschrieben wird.

described resource URI

Ein URI, der die beschriebene Ressource (*described resource*) identifiziert.

description

Eine oder mehrere Aussagen (*statements*) über eine, und nur eine, beschriebene Ressource (*described resource*).

description set

Ein Satz aus einer oder mehreren Beschreibungen (*descriptions*), von denen jede eine einzelne Ressource (*resource*) beschreibt.

element, <http://www.w3.org/1999/02/22-rdf-syntax-ns#Property>

Ein Synonym für *property*. Es ist anzumerken, dass "element" gewöhnlich auch für die strukturellen Markup-Bestandteile eines XML-Dokuments verwendet wird.

has domain, <http://www.w3.org/2000/01/rdf-schema#domain>

Die Beziehung zwischen einer Eigenschaft (*property*) und einer Klasse (*class*), die die Klasse (*class*) bezeichnet, deren Instanz eine beschriebene Ressource (*described resource*) ist, wenn die Eigenschaft Teil eines Paares aus Eigenschaft und Wert (*property value pair*) ist, mit dem die Ressource beschrieben wird.

has range, <http://www.w3.org/2000/01/rdf-schema#range>

Die Beziehung zwischen einer Eigenschaft (*property*) und einer Klasse (*class*), aus der folgt, dass dann, wenn die Eigenschaft (*property*) Teil eines Paares aus Eigenschaft und Wert (*property value pair*) ist, der Wert (*value*) eine Instanz jener Klasse (*class*) ist.

instance of

Die Beziehung zwischen einer Ressource (*resource*) und einer Klasse (*class*), die die Klasse (*class*) bezeichnet, deren Instanz eine Ressource (*resource*) ist.

literal

Ein nach RDF definiertes Literal (*literal*) ist eine Entität, die, um eine Ressource (*resource*) zu bezeichnen eine Unicode-Zeichenkette sowie optional die Sprache und/oder den Datentyp verwendet.

literal value

Ein Wert (*value*), der ein Literal (*literal*) ist.

literal value surrogate

Ein Platzhalter (*value surrogate*) für einen literalen Wert (*literal value*), der aus genau einer Zeichenkette (*value string*) (einem Literal (*literal*), das den Wert (*value*) codiert) besteht

member of

Die Beziehung zwischen einer Ressource (*resource*) und einem Semantischen Codierungsschema (*vocabulary encoding scheme*), die den Satz bezeichnet, dem die Ressource (*resource*) angehört.

non-literal value

Ein Wert (*value*), der eine physische, digitale oder begriffliche Entität ist.

non-literal value surrogate

Ein Platzhalter (*value surrogate*) für einen nicht-literalen Wert (*non-literal value*), der aus dem URI der Eigenschaft (*property URI*) (einem URI, der die Eigenschaft identifiziert), Null oder einem URI des Werts (*value URI*) (einem URI, der den nicht-literalen Wert, der mit der Eigenschaft assoziiert ist,

identifiziert), Null oder einem URI des Semantischen Codierungsschemas (*vocabulary encoding scheme URI*) (einem URI, der das Semantische Codierungsschema, dem der nicht-literale Wert angehört, identifiziert) und Null oder mehreren Zeichenketten (*value strings*) (Literalen, die den Wert repräsentieren).

plain value string

Ein Zeichenkette (*value string*), die nicht durch den URI eines syntaktischen Codierungsschemas (*syntax encoding scheme URI*) festgelegt ist.

property, <http://www.w3.org/1999/02/22-rdf-syntax-ns#Property>

Ein besonderer Aspekt, eine Eigenart, ein Attribut oder eine Beziehung, die verwendet werden, um Ressourcen (*resources*) zu beschreiben.

property URI

Ein URI, der eine Eigenschaft (*property*) identifiziert.

property/value pair

Die Verbindung zwischen einer Eigenschaft (*property*) mit einem Wert (*value*), die verwendet wird, um die Art einer Ressource (*resource*) zu beschreiben.

record

Die Instantiierung eines Beschreibungssatzes (*description set*), die den DCMI-Codierungs-Richtlinien (z.B. für XHTML meta tags, XML oder RDF/XML) entsprechend erstellt wird.

resource, <http://www.w3.org/2000/01/rdf-schema#Resource>

Alles, was identifiziert werden kann. Gängige Beispiele sind elektronische Dokumente, Abbildungen, Dienste (z.B. "der heutige Wetterbericht für Los Angeles") oder eine Sammlung anderer Ressourcen (*resources*). Nicht alle Ressourcen (*resources*) sind im Netz "abrufbar"; so können zum Beispiel Menschen, Firmen, Ideen und gebundene Bücher in einer Bibliothek ebenfalls als Ressourcen (*resources*) angesehen werden.

statement

Die Instantiierung eines Paares aus Eigenschaft und Wert (*property-value pair*), die aus dem URI der Eigenschaft (*property URI*) (dem URI, der die Eigenschaft identifiziert) und einem Platzhalter (*value surrogate*) besteht.

sub-class of, <http://www.w3.org/2000/01/rdf-schema#subClassOf>

Eine Beziehung zwischen zwei Klassen (*classes*), in der die beiden Klassen (*classes*) dergestalt definiert sind, dass alle Ressourcen (*resources*), die Instanzen der Subklasse sind, auch Instanzen der anderen Klasse (*class*) sind.

sub-property of, <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>

Eine Beziehung zwischen zwei Eigenschaften (*properties*), in der die beiden Eigenschaften (*properties*) dergestalt definiert sind, dass dann, wenn die Sub-

Eigenschaft (*sub-property*) Teil eines Paares aus Eigenschaft und Wert (*property-value pair*) ist, das eine Ressource (*resource*) beschreibt, die Ressource (*resource*) gleichzeitig mit einem zweiten Paare aus Eigenschaft und Wert (*property-value pair*) beschrieben wird, das aus der Eigenschaft (*property*) und dem Wert (*value*) besteht.

encoding scheme, <http://www.w3.org/2000/01/rdf-schema#Datatype>

Ein Satz bestehend aus Zeichenketten und ein dazugehöriger Satz aus Regeln, der das Mapping zwischen dem Satz aus Zeichenketten und dem einer Gruppe von Ressourcen (*resources*) beschreibt. Die Regeln für das Mapping können Struktur der Zeichenkette festlegen (zum Beispiel DCMI Box) oder sie können einfach alle Zeichenketten und die entsprechenden Ressourcen aufzählen (zum Beispiel ISO 3166).

syntax encoding scheme URI

Ein URI, der ein syntaktisches Codierungsschema (*syntax encoding scheme*) identifiziert.

term

Eine Eigenschaft (*property*) (ein Element (*element*)), eine Klasse (*class*), ein semantisches Codierungsschema (*vocabulary encoding scheme*) oder ein syntaktisches Codierungsschema (*syntax encoding scheme*).

typed value string

Ein Zeichenkette (*value string*), die durch den URI eines syntaktischen Codierungsschemas (*syntax encoding scheme URI*) festgelegt ist.

URI

Ein Uniform Resource Identifier^[7] oder Internationalized Resource Identifier^[8]
Für die Zwecke des DCMI Abstract Model wird die Äquivalenz von URIs durch RDF definiert.

value

Die physische Entität, die begriffliche Entität oder das Literal (*literal*), die zu einer Eigenschaft (*property*) gehören, wenn ein Paar aus Eigenschaft und Wert (*property-value pair*) verwendet wird, um eine Ressource (*resource*) zu beschreiben.

value URI

Die URI, die den Wert (*value*) identifiziert.

value string

Ein Literal (*literal*), das entweder zu dem URI eines syntaktischen Codierungsschemas (*syntax encoding scheme*) gehören oder mit einer Sprachangabe (*value string language*) verbunden sein kann. Im Falle eine literalen Platzhalters (*literal value surrogate*) codiert die Zeichenkette (*value string*) den Wert (*value*); im Falle eines nicht-literalen Platzhalters (*non-literal value surrogate*) repräsentiert die Zeichenkette (*value string*) den Wert (*value*).

value string language

Eine ISO-normierte Sprachangabe, die die Sprache einer Werte-Zeichenkette (*value string*) anzeigt.

value surrogate

Ein literaler Platzhalter (*literal value surrogate*) oder ein nicht-literaler Platzhalter (*non-literal surrogate*).

vocabulary

Ein Satz aus einem oder mehreren Termen (*terms*).

vocabulary encoding scheme, <http://purl.org/dc/dcam/VocabularyEncodingScheme>

Die Auflistung der Ressourcen (*resources*), die zu einem Gruppe gehören.

vocabulary encoding scheme URI

Ein URI, der ein semantisches Codierungsschema (*vocabulary encoding scheme*) identifiziert.

8. Beziehung zu den älteren "DCMI grammatischen Prinzipien"

Das Modell, das den Dublin-Core-Metadaten zugrundeliegt, hat sich seit den späten 1990er Jahren, als die ersten Formalismen aufgestellt wurden, weiterentwickelt. Die folgende Tabelle ist ein grober Überblick über die Analogien zwischen der Terminologie früherer Versionen der "DCMI Grammatical Principles"^[9] und dem vorliegenden DCMI Abstract Model.

DCMI Grammatical Principles (2003)	DCMI Abstract Model (2007)
vocabulary term	<i>resource</i>
element	<i>property</i> oder <i>element</i>
element refinement	<i>property</i> mit einer <i>sub-property-of</i> -Beziehung
encoding scheme	<i>syntax encoding scheme</i> oder <i>vocabulary encoding scheme</i>
syntax encoding scheme	<i>syntax encoding scheme</i>
qualifier	<i>property</i> mit einer <i>sub-property-of</i> -Beziehung, <i>syntax encoding scheme</i> oder <i>vocabulary encoding scheme</i>
vocabulary encoding scheme	<i>vocabulary encoding scheme</i>

Tabelle 2: Grammatische Prinzipien der DCMI und das DCMI Abstract Model

Quellenangaben

1. Klyne, Graham and Jeremy Carroll, editors. Resource Description Framework: Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/rdf-concepts/>.
2. Brickley, Dan and R.V. Guha, editors. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. 10 February 2004. <http://www.w3.org/TR/rdf-schema/>.
3. Booch, Grady, James Rumbaugh und Ivar Jacobson. The Unified Modeling Language User Guide. Addison-Wesley, 1998.
4. DCMI Encoding Guidelines. <http://dublincore.org/resources/expressions/>.
5. Hayes, Patrick, editor. RDF Semantics. W3C Recommendation. 10 February 2004. <http://www.w3.org/TR/rdf-mt/>.
6. Nilsson, Mikael, Andy Powell, Pete Johnston, Ambjörn Naeve. Expressing Dublin Core metadata using the Resource Description Framework (RDF). DCMI Recommendation. January 2008. <http://dublincore.org/documents/dc-rdf/>.
7. Berners-Lee, T., R. Fielding, L. Masinter. RFC 2986: Uniform Resource Identifier (URI): Generic Syntax. Internet Engineering Task Force (IETF). January 2005. <http://www.ietf.org/rfc/rfc2986.txt>.
8. Duerst, M., M. Suignard. RFC 3987: Internationalized Resource Identifiers (IRIs). Internet Engineering Task Force (IETF). January 2005. <http://www.ietf.org/rfc/rfc3987.txt>.
9. DCMI Usage Board. DCMI Grammatical Principles. November 2003. <http://dublincore.org/usage/documents/principles/>.