



Kleines Handbuch Metadaten

Metadatenprofile

Stefanie Rühle

1 Einleitung

Metadatenanwendungen sind notwendig, um Informationen zu finden, zu identifizieren und darauf zuzugreifen. Dies waren schon immer die zentralen Aufgaben von Metadaten und Metadatenanwendungen und daran hat sich auch im Zeitalter des Internet nichts geändert. Allerdings stellt die Veröffentlichung und Verlinkung von Metadaten im Semantic Web neue Anforderungen an die Datenprovider. Um Datenbestände unterschiedlicher Anbieter vernetzen zu können, muss transparent sein, was wozu womit beschrieben wird. Die Dokumentation solcher Informationen erfolgt i.d.R. in Form von Metadatenprofilen. Unter Berücksichtigung der Anforderungen von Nutzern, der Art der Ressourcen und der Systeme, in denen die Metadaten verwaltet werden, beschreiben sie die Metadatenterme, die für eine bestimmte Anwendung relevant sind. Was bei der Gestaltung eines solchen Metadatenprofils beachtet werden muss, wurde von der Dublin Core Metadata Initiative (DCMI) bereits 2008 im Rahmen des Singapore Framework for Dublin Core Application Profiles näher spezifiziert. Ziel des Singapore Framework ist es, die strukturelle Interoperabilität von Metadaten in vernetzten Systemen zu verbessern und zwar hinsichtlich ihrer inneren Konsistenz als auch hinsichtlich ihrer Konformität mit den Anforderungen von Interoperabilitätsmodellen wie dem Resource Description Framework (RDF) und dem DCMI Abstract Model (DCAM).

Der folgende Beitrag beschreibt was Metadatenprofile sind und wie sie aufgebaut sein müssen, um den Anforderungen des Singapore Framework zu entsprechen.

2 Metadatenprofile

2.1 Was ist ein Metadatenprofil?

“In the metadata community, the term application profile has been applied to describe the tailoring of standards for specific applications.”¹

Ein Metadatenprofil ist die detaillierte Beschreibung einer spezifischen Metadatenanwendung, die es ermöglicht, Metadaten auch außerhalb dieser Anwendung (z. B. in anwendungsübergreifenden Portalen) zu nutzen. Das Profil spezifiziert die Wiederverwendbarkeit vom Standpunkt

- des Datenproviders, der Metadaten unterschiedlichen Ursprungs in eine Metadatenanwendung einbinden möchte,
- des Entwicklers, der auf der Grundlage eines vorliegenden Metadatenprofils eine oder mehrere Anwendungen erstellt.

Um Metadatenanwendungen interoperabel zu gestalten, müssen Metadatenstandards verwendet werden – d.h. Anwendungen, die Metadaten untereinander austauschen, sollten mehr oder weniger gemeinsame Modelle, Elemente bzw. Syntax haben. Metadatenstandards sorgen für die Konsistenz der Metdaten, ermöglichen so den Datenaustausch und die anwendungsübergreifende Verknüpfung von Metadaten und Ressourcen, die durch diese Metadaten repräsentiert werden. Allerdings sind die in Metadatenstandards definierten Terme für konkrete Metadatenanwendungen nicht selten zu umfangreich – für die Anwendung werden daher nur die relevanten Metadatenterme ausgewählt – zu ungenau spezifiziert – für die Anwendungen gelten strikere Einschränkungen hinsichtlich des Verpflichtungsgrads, der Wiederholbarkeit usw. – und in manchen Fällen auch nicht ausreichend – für die Anwendung werden dann Terme aus anderen Standards übernommen oder neue Terme entwickelt, die sich mal mehr mal weniger als Subelemente einzelnen Termen aus dem ausgewählten Standard zuweisen lassen. Derartige Modifikationen des Standardformats sind i.d.R. nur über den Quellcode der Anwendung nachvollziehbar. Dies reicht zwar aus, um Metadaten „maschinenlesbar“ zu verarbeiten, erschwert aber das Wissen um das Format, das der Anwendung tatsächlich zugrunde liegt, ein Wissen, dass für die Nachnutzung der Daten und die häufig damit verbundenen Mappings sowie für die Nachhaltigkeit der Metadaten und der Metadatensysteme von entscheidender Bedeutung ist. Darum braucht es eine „menschenslesbare“ Dokumentation, die den Gebrauch eines Metadatenstandards und weiterer Metadatenterme innerhalb einer konkreten Anwendung übersichtlich beschreibt. Eine solche Dokumentation bzw. Spezifikation wird im Dublin Core Kontext Anwendungsprofil oder Metadatenprofil genannt.

¹ <http://dublincore.org/documents/singapore-framework/>

DCMI erkannte schon früh, dass eine nachvollziehbare und übersichtliche Dokumentation der verwendeten Metadaterme für die Interoperabilität von Metadatenanwendungen von großer Bedeutung ist. Die Möglichkeit, Metadaterme verschiedener Metadatenstandards miteinander zu kombinieren wenn die Metadaterme eines einzelnen Standards nicht ausreichen, hat zur Folge, dass identische Metadaterme nicht nur anwendungsübergreifend, sondern häufig auch fächerübergreifend Verwendung finden. Metadatenprofile sind in diesem Sinne

“... schemas which consist of data elements drawn from one or more namespaces, combined together by implementers and optimized for a particular local application.”²

Durch die Verwendung von Metadatenprofilen bietet sich somit die Möglichkeit, Metadaterme unterschiedlicher Standards zu kombinieren, allerdings unter der Voraussetzung, dass sie innerhalb eines bestimmten Namespace definiert und somit identifizierbar sind.³ Zudem sollte in einem Metadatenstandard der Verpflichtungsgrad, die Wiederholbarkeit und weitere Regeln, die für die Verwendung der Terme in dieser konkreten Anwendung gelten, definiert werden. Gilt z. B. für die Strukturierung der Datumsangaben ein Encoding wie ISO 8601 oder ist für die inhaltliche Beschreibung von Dokumenten ein kontrolliertes Vokabular wie die DDC oder die Schlagwortnormdatei verpflichtend, so muss dies im Metadatenprofil beschrieben werden.

2.2 Standards für die Gestaltung von Metadatenprofilen

Um Metadatenprofile möglichst einheitlich zu gestalten, hat DCMI Regeln für den Aufbau von Metadatenprofilen entwickelt.

2.2.1 CWA 14855:2003

Im Jahr 2003 veröffentlichte das European Committee for Standardization (CEN) die „Dublin Core Application Profile Guidelines“ (CWA 14855:2003)⁴. Der Standard bestimmt den strukturellen Aufbau von Metadatenprofilen und definiert Attribute, die verwendet werden müssen, um die für eine Anwendung relevanten Metadaterme im Rahmen eines Metadatenprofils zu identifizieren.⁵ Bereits zu diesem Zeitpunkt wurde neben identifizierenden Attributen wie der Bezeichnung des Elements und der Person oder Organisation, die für das Element verantwortlich ist auch ein URI gefordert, das das Element als solches identifizierbar macht. Insgesamt unterscheidet der Standard für die Beschreibung von Metadatermen vier Gruppen von beschreibenden Attributen:⁶

- identifizierende Attribute - Term URI, Name, Label, Defined By
- definierende Attribute - Definition, Comments, Type of Term
- Attribute zur Beschreibung von Relationen zwischen Metadatermen Relational Attributes – Refines, Refined By, Encoding Scheme For, Has Encoding Scheme, Similar To
- Attribute zur Beschreibung von Einschränkungen, die für die Metadaterme gelten – Obligation, Condition, Datatype, Occurrence

2.2.2 Singapore Framework

Unter Berücksichtigung der Anforderungen von Linked Data und Semantic Web entstand in den folgenden Jahren das „Singapore Framework for Dublin Core Application Profiles“⁷ ein Dublin Core Standard, der definiert, was ein Metadatenprofil in seiner Gesamtheit ausmacht und welcher Komponenten es bedarf, um die effiziente und nachhaltige Nutzung von Metadaten zu unterstützen. Dabei wurde vor allem auch der

² <http://www.ariadne.ac.uk/issue25/app-profiles>

³ Ein Term ist dann ausreichend dokumentiert und definiert, wenn er einen korrekten URI hat und einem der vier der in DCAM genannten Arten von Metadatermen zugewiesen ist, als da sind: class, property, data type und vocabulary encoding scheme, s.

<http://www.dublincore.org/documents/abstract-model/>

⁴ <ftp://ftp.cenorm.be/public/cwas/e-europe/mmi-dc/cwa14855-00-2003-nov.pdf>

⁵ “By providing a consistent presentation structure for such documents, these guidelines aim at making it easier for people to understand what others are doing in their metadata.” Ebd.: S. 5.

⁶ Ebd., S. 9 f.

⁷ <http://dublincore.org/documents/singapore-framework/>

Kontext berücksichtigt, in dem ein Metadatenprofil entsteht – steht ein Metadatenprofil doch in einem Spannungsfeld zwischen den Anforderungen der Zielgruppe, die die Metadaten-Anwendung nutzen soll, den Eigenschaften der Ressourcen, die mit den Metadaten beschrieben werden und den Möglichkeiten der Systeme, die für das Erfassen, Speichern und/oder die Bereitstellung der Daten verwendet werden.

Gemäß dem Singapore Framework muss ein Metadatenprofil festhalten, was wozu womit beschrieben wird. Entsprechend besteht ein Dublin Core Metadatenprofil aus den „Functional Requirements“, die die Frage beantworten sollen, wozu die Metadaten gebraucht werden, dem „Domain Model“, das die Frage beantwortet, was für Ressourcen beschrieben werden und welche Beziehungen zwischen diesen Ressourcen bestehen und dem „Description Set Profile“, das die Frage beantwortet, womit die Ressourcen beschrieben werden – Terme, Standards, Regeln. Ziel des Singapore Framework ist eine einheitliche Dokumentation der Verwendung von Metadaten. Diese Einheitlichkeit sorgt für Transparenz –in Bezug auf Funktion und Aufbau der Anwendung an sich, aber vor allem in Bezug auf die Metadaten, die in dieser Anwendung gespeichert und zur Verfügung gestellt werden. Das Metadatenprofil gibt somit Auskunft über den Grad der Interoperabilität der Anwendung, für die das Profil steht. Ziel des Singapore Framework ist aber vor allem auch die Konformität mit Linked Data geeigneten Interoperabilitätsmodellen wie dem Resource Description Framework (RDF)⁸ und dem DCMI Abstract Model (DCAM)⁹. DCAM Konformität der im „Description Set Profile“ definierten Terme ist darum ein wichtiges Qualitätskriterium für Dublin Core Metadatenprofile.¹⁰

DCAM Konformität

Das DCMI Abstract Model (DCAM) ist ein Meta-Modell, das die Informationsstrukturen abbildet, die bei der Beschreibung von Ressourcen eine Rolle spielen, wobei es sich vor allem an den im RDF Modell definierten Entitäten orientiert. Da DCAM syntaxunabhängig ist, ist es mit jeder im Linked Data Kontext verwendeten Syntax konform, kann aber auch als Grundlage für den Aufbau von traditionellen Datensätzen verwendet werden. DCAM liefert damit das Gerüst für die Beschreibung von Metadatermen im Rahmen des im Singapore Framework geforderten Description Set Profile und nennt die Anforderungen an DCAM-konforme Metadatenelemente.

Abbildung 1: Description Set Modell



⁸ <http://www.w3.org/RDF/>

⁹ <http://dublincore.org/documents/abstract-model/>

¹⁰ <http://dublincore.org/documents/2009/03/02/profile-review-criteria/>

Dabei unterscheidet DCAM unter Berücksichtigung der RDF-Entitäten verschiedene Arten von Metadatermenen:

- Classes – definieren die Art der Ressourcen, die beschrieben werden.
- Properties – definieren, was für Eigenschaften eine bestimmte Art von Ressourcen hat
- Syntax Encoding Scheme – definieren die Syntax des Wertes, der zu einer bestimmten Property gehört
- Vocabulary Encoding Scheme – definieren die möglichen Werte (z. B. Konzepte aus Thesauri, Klassifikation u.a. Normdateien), die bei der Verwendung einer bestimmten Property erlaubt sind.

Ein Informationsmodell wie DCAM unterstützt die Interoperabilität aber nicht nur durch die Festlegung von Anforderungen, vielmehr hilft es uns, Informationsstrukturen zu verstehen und erleichtert damit die Entwicklung von Mappings und syntaxübergreifender Datenkonvertierung.

3 In sieben Schritten zum Profil

Im Folgenden wird im Überblick gezeigt, was bei der Entwicklung eines Metadatenprofils beachtet werden muss. Grundlage sind die verpflichtenden Bestandteile des Singapore Framework: die „Functional Requirements“, das „Domain Model“ und das „Description Set Profile“. Außerdem verweisen wir auf verschiedene im Dublin Core Kontext entstandene Richtlinien für die syntaktische Kodierung von Metadatenprofilen.

3.1 Erster Schritt: Wozu werden die Metadaten benötigt?

Bevor die Entscheidung für oder gegen einen bestimmten Metadatenstandard oder einzelne Metadaterme fällt, muss zunächst deutlich werden, was mit den Metadaten in einer speziellen Anwendung überhaupt erreicht werden soll, welchem Zweck sie dienen, welche funktionellen Anforderungen an die Metadaten gestellt werden. Dabei muss unterschieden werden zwischen den Zielen des Diensteanbieters, der die Anwendung zur Verfügung stellt und den Anforderungen der Nutzer dieser Anwendung. Folgende Fragen müssen in diesem Zusammenhang zunächst beantwortet und dokumentiert werden:

- In welchem Kontext sollen die Metadaten genutzt werden, in welchem Kontext müssen sie interoperabel sein?
 - Sollen die Metadaten außerhalb der eigenen Anwendung zur Verfügung gestellt werden und wenn ja, in welchem Ausmaß (Austausch von Metadaten abteilungsübergreifend, communityübergreifend, interdisziplinär, ...)?
 - Werden die Metadaten in einen wissenschaftlichen Kontext eingebunden (z. B. zur Beschreibung von Forschungsdaten, wissenschaftlicher Literatur usw.) oder in einen geschäftlichen Kontext (z. B. zur Beschreibung von Waren, Logistikabläufen usw.) oder in einen anderen Kontext?
- Wer ist für das Metadatenprofil zuständig?
 - Wer hat das Metadatenprofil erarbeitet (einzelne Personen, Organisationen usw.)?
 - Wer wird das Profil in Zukunft pflegen und verfügbar machen?
- Wer ist die Zielgruppe der Metadatenanwendung?
 - StudentInnen, WissenschaftlerInnen, SchülerInnen, Fußballfans, SachbearbeiterInnen im Einwohnermeldeamt, usw. ?
 - Deutschsprachig, englischsprachig, mehrsprachig usw.?
- In welcher Form sollen die Metadaten diese Zielgruppe unterstützen? Sollen sie dazu dienen:
 - Informationen für eine wissenschaftliche Arbeit zu recherchieren?
 - den neuen Roman von Henning Mankell auszuleihen?
 - nach Programmen für Datenanalysen zu browsen?

- die Ergebnisse des letzten HSV-Spiels anzusehen?
- Wahlbenachrichtigungen an die BürgerInnen einer Gemeinde zu verschicken
- usw.
- Was für funktionelle Anforderungen ergeben sich daraus? Welche Anwendungsszenarien gibt es?
 - Stichwortsuche nach Namen, Titeln, Themen?
 - Browsing über Themen, Orte, Jahresangaben?
 - Auflistung/Anzeige von Namen, Titeln, Themen, Zugriffsvoraussetzungen?
 - usw.

3.2 Zweiter Schritt: Was wird beschrieben?

Im nächsten Schritt werden die Ressourcen, die über die Metadaten auffindbar gemacht werden sollen, analysiert und in Form eines Domain-Modells beschrieben. Dabei sind die folgenden Fragen zu beantworten:

- Was für Ressourcen werden beschrieben?
 - Bücher, Filme, Internetressourcen, Verfasser, Herausgeber, Computerspiele, Betriebssysteme oder Orte usw.
- Welchen Klassen sind diese Ressourcen zuzuordnen? Dabei ist zu beachten, dass die Zuordnung zu einer Klasse von der Granularität abhängt, mit der die Ressourcen beschrieben werden soll, und diese wiederum ergibt sich aus den Anforderungen, die die Zielgruppe an die Anwendung hat.
 - Bücher, Filme und Internetressourcen könnten zur Klasse „Medien“ gehören,
 - Verfasser und Herausgeber könnten zur Klasse „Personen“ oder zur Klasse „Organisationen“ gehören,
 - Computerspiele und Betriebssysteme könnten zur Klasse „Software“ gehören,
 - Orte könnten zur Klasse „Geographika“ gehören
 - usw.
- Was für Eigenschaften haben die Ressourcen?
 - Titel, Verfasser, Seitenzahl, Abspieldauer, Signatur, URL, Name, Standort, Speicherkapazitäten, Anbieter, Alter, Adresse, technische Voraussetzungen usw.
- Welche dieser Eigenschaften dienen dazu, Beziehungen zwischen Ressourcen zu beschreiben?
 - Verfasser: Buch *wird verfasst von* Person.
 - Anbieter: Software *wird zur Verfügung gestellt von* Organisation.
 - Technische Voraussetzungen: Programm *wird verwendet mit* Hardware.
 - Adresse: Person *ist wohnhaft in* Geographika.

3.3 Dritter Schritt: Welche Eigenschaften sind für die Anwendung relevant?

In Abhängigkeit von den im ersten Schritt definierten Anforderungen werden die im zweiten Schritt genannten Eigenschaften auf ihre Relevanz hin geprüft. Die ausgewählten Eigenschaften müssen die Metadatenanwendung dabei unterstützen, die Bedürfnisse der Zielgruppe zu befriedigen. Dabei müssen z. B. folgende Fragen geklärt werden:

- Welche Eigenschaften und Beziehungen sollen in der Metadatenanwendung suchbar gemacht werden?
- Welche Eigenschaften und Beziehungen sollen in der Metadatenanwendung angezeigt werden?

- Welche Eigenschaften und Beziehungen werden benötigt, damit die Nutzer auf die Ressourcen zugreifen können?

Für die Stichwortsuche nach Namen, Titeln und Themen müssten die Eigenschaften „Name“, „Titel“ und „Thema“ in der Anwendung suchbar sein. Durch die Verwendung von kontrollierten Vokabularen, die bestimmten Eigenschaften als mögliche Werte zugewiesen werden, können Listen aufgebaut werden, die die Suche nach relevanten Ressourcen unterstützen. Zudem gibt es Eigenschaften, die den Nutzer bei der Selektion der für ihn relevanten Ressourcen unterstützen, die für die Suche aber u. U. nicht benötigt werden – z. B. Seitenzahlen oder technische Voraussetzungen. Schließlich braucht es Informationen für den Zugriff auf die Ressourcen – z. B. den Identifier der Ressource oder den Namen des Anbieters, über den man das Objekt erhalten kann.

Die Eigenschaften, die in diesem Zusammenhang ausgewählt werden, müssen die Anforderungen der Zielgruppen widerspiegeln. Eigenschaften und Beziehungen, die im zweiten Schritt genannt wurden aber an dieser Stelle als nicht relevant für die funktionellen Anforderungen erachtet werden, sollten in den folgenden Schritten nicht mehr berücksichtigt werden.

3.4 Vierter Schritt: Welche Metadaterme werden verwendet?

Unter Berücksichtigung der im dritten Schritt ausgewählten Eigenschaften und des Kontext, in dem die Anwendung steht, wird geprüft, ob für diesen Kontext bereits ein Metadatenstandard existiert und wenn ja, inwieweit dieser für die Metadatenanwendung geeignet ist. Handelt es sich z. B. darum, traditionelle bibliothekarische Datenbestände zusammenzuführen, so bietet sich ein in diesem Kontext verbreiteter Standard wie MARC 21¹¹ an, da viele Bibliotheken ihre Daten in Form dieses Standards zur Verfügung stellen. Handelt es sich hingegen um die Beschreibung von Internetressourcen die als Linked Data veröffentlicht werden sollen, so empfiehlt sich eher die Verwendung der DCMI Metadata Terms¹². Geht es um die Beschreibung von Personen im Rahmen einer Expertendatenbank könnte FOAF¹³ verwendet werden.¹⁴

Die möglichen Metadatenstandards werden dahingehend geprüft, ob die Terme für die Metadatenanwendung geeignet sind und die im dritten Schritt ausgewählten Eigenschaften adäquat wiedergeben. Dabei erlaubt das Singapore Framework die Verwendung von Termen unterschiedlicher Standards solange diese Terme DCAM-konform sind – sich die Metadaterme den im Abstract Modell genannten Arten von Termen (Class, Property, Data Type, Vocabulary Encoding Scheme) zuweisen lassen und durch einen URI identifizierbar sind.

Abbildung 2: DCAM-konformer Term am Beispiel der Property alternative des Dublin Core Metadata Elements Sets¹⁵

Term Name:	alternative
URI:	http://purl.org/dc/terms/alternative
Label:	Alternative Title
Definition:	An alternative name for the resource.
Comment:	The distinction between titles and alternative titles is application-specific.
Type of Term:	Property
Refines:	http://purl.org/dc/elements/1.1/title
Refines:	http://purl.org/dc/terms/title
Version:	http://dublincore.org/usage/terms/history/#alternative-003
Has Range:	http://www.w3.org/2000/01/rdf-schema#Literal

¹¹ <http://www.loc.gov/marc/>

¹² <http://dublincore.org/documents/dcmi-terms/>

¹³ <http://xmlns.com/foaf/spec/>

¹⁴ Einen kurzen Überblick über mögliche Metadatenstandards aus dem Bereich des Kulturerbes gibt es in dem Beitrag „Metadaten“ in diesem Handbuch.

¹⁵ <http://dublincore.org/documents/dcmi-terms/#terms-alternative>

Benötigt eine Anwendung Metadaterme, die sich durch keinen vorhandenen Standard abdecken lassen, so kann ein anwendungsspezifischer Term definiert werden. Auch dieser Term kann nur dann mit den Termen der verschiedenen Standards kombiniert werden, wenn er DCAM-konform ist.

Abbildung 3: Anwendungsspezifischer Term am Beispiel der Property *hasTranslation* des Scholarly Works Application Profile¹⁶

Has Translation	
Term URI	http://purl.org/eprint/terms/hasTranslation
Definition	A translation of the described resource.
Domain	http://purl.org/eprint/entityType/Expression
Range	http://purl.org/eprint/entityType/Expression
Type of term	Property
Refines	http://purl.org/dc/elements/1.1/relation

3.5 Fünfter Schritt: Was für Regeln gelten für die ausgewählten Metadaterme

Die Dokumentation der verwendeten Terme erfolgt im Description Set Profile. Hier wird beschrieben, welche Metadaterme für die Anwendung zur Verfügung stehen und welche Regeln für die Metadaterme gelten. Dabei muss für jede Klasse von Ressourcen, die im Rahmen des Domain-Modells definiert wurden, ein eigenes Beschreibungsschema erstellt werden. Das Beschreibungsschema definiert die Klassen und deklariert, was für Aussagen über diese Klassen gemacht werden.

Abbildung 4: Definition der im DCMI Libraries Application Profile verwendeten Klassen¹⁷:

DCLib AP/DCLib Classes	
Classes of the DC-Lib AP for Bibliographic Text Resources	
BibliographicTextResource	
Name:	BibliographicTextResource
URI:	http://purl.org/dclib/terms/BibliographicTextResource
Definition:	The intellectual creation, realization and physical embodiment of a work in text.
Type of Term:	Class
DCLIB-Comment:	BibliographicTextResource aggregates the properties of the FRBR entities work, expression and manifestation in one class.
Narrower Than:	http://purl.org/dc/terms/BibliographicResource
Narrower Than:	http://purl.org/dc/dcmitype/Text
Item	
Name:	Item
URI:	http://filastandards.info/ns/fr/frbr/frbrer/C1004
Definition:	A single exemplar of a manifestation.
Type of Term:	Class
DCLIB-Comment:	An item is the single copy of a Bibliographic Text Resource.

Auch die Aussagen werden in Form eines Schemas beschrieben, des Aussagenschemas. Das Aussagenschema definiert die Metadaterme, indem es nicht nur die Eigenschaften nennt, sondern auch die für diese Metadatenanwendung geltenden Encoding Scheme sowie alle weiteren Regeln, die für diese Aussage gelten (Verpflichtungsgrad, Kardinalität, usw.)

¹⁶ http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_Terms#Has_Translation

¹⁷ http://wiki.dublincore.org/index.php/DCLib_AP/DCLib_Classes

Abbildung 5: Aussagenschema am Beispiel der DCMES Property Title, die im DCLib AP verwendet wird, um Instanzen der Klasse *Bibliographic Resource* zu beschreiben¹⁸

DCLib AP/Bibliographic Resource Properties

Properties used with Bibliographic Text Resource

Title

Name: title

URI: <http://purl.org/dc/elements/1.1/title>

Definition: A name given to the resource.

Type of Term: Property

DCLIB-Comment:

- A parallel/transliterated title is considered a main title, i.e. the title property is repeated.
- If no title is available, best practice is to give a constructed title, derive a title from the resource or supply [no title].
- Retain initial articles and use local sorting algorithms based on language. (For example, see: [Initial Definite and Indefinite Articles](#) for a list of article in various languages).
- A language qualifier may be used to indicate language of title if appropriate. (See [Guidelines for the Creation of Title Content](#))

Has Range:

- dc:title may be used with literal or non-literal values. For further information see [RDF Examples](#)

Obligation: M

Occurrence: 1-n

Für die Beschreibung eines Metadaterms sollten abhängig von der Art des Terms die folgenden Attribute verwendet werden:

- **URI:** dient der ausdrücklichen Identifikation des Terms
- **Name:** die Bezeichnung des Terms
- **Definition:** Die Beschreibung des Terms, die deutlich macht, inwieweit sich die Terme semantisch voneinander unterscheiden
- **Term-Typ:** gibt an, um welche Art von Metadaterm es sich handelt (Class, Property, Data Type, Vocabulary Encoding Scheme)
- **Occurrence:** gibt an, wie oft ein Term vom Typ „Property“ zur Beschreibung einer bestimmten Klasse von Ressourcen verwendet werden darf/muss
- **Refinement:** wird verwendet, wenn es sich bei dem Term um einen Subterm handelt. Ist dies der Fall, so muss der Term, dem der beschriebene Term als refinement zugewiesen wurde, hier genannt werden.
- **Type of Value:** gibt an, ob der Wert, der bei der Beschreibung der Ressource verwendet wird, ein einfacher Textstring ist (literal value) oder die URI einer Ressource (non-literal value).
- **Syntax Encoding Scheme:** wird verwendet, wenn der literale Wert ein literaler Wert einer bestimmten Syntax entsprechen soll
- **Vocabulary Encoding Scheme:** wird verwendet, wenn der Wert ein Term aus einem bestimmten kontrollierten Vokabular sein muss.

Die im Aussagenschema beschriebenen Regeln müssen den Anforderungen entsprechen, die im ersten Schritt definiert wurden. Zudem ist darauf zu achten, dass die im Domain Modell genannten Beziehungen zwischen Ressourcen adäquat wiedergegeben werden können.

Da das Singapore Framework die Kombination von Termen unterschiedlicher Standards und die Kombination von Termen aus Standards mit anwendungsspezifischen Termen erlaubt, kann ein Beschreibungsschema Terme aus unterschiedlichen Domänen umfassen. Voraussetzung hierfür ist jedoch weiterhin die DCAM-Konformität aller verwendeten Terme. Zudem muss die Definition der verwendeten Terme eindeutig sein: Metadaterme innerhalb einer Anwendung dürfen sich auf gar keinen Fall semantisch überschneiden. Um die Interoperabilität sicherzustellen, sollte aber auch darauf geachtet werden, dass es keine semantischen Schnittmengen mit

¹⁸ http://wiki.dublincore.org/index.php/DCLib_AP/Bibliographic_Resource_Properties

Termen gibt, die in der Community verbreitet und anerkannt sind. Und schließlich lässt sich die Semantik eines Terms anwendungsspezifisch einschränken. Auch diese Einschränkung muss eindeutig definiert werden und darf nicht in Widerspruch stehen zu der Definition, die dem Term im Metadatenstandard zugewiesen wurde.

3.6 Sechster Schritt: Was für Encoding Scheme gelten für die einzelnen Terme?

Ein Encoding Scheme nennt die Standards, die ggf. bei der Beschreibung einer Ressource mit einer bestimmten Property gelten. Was für Encoding Scheme verwendet werden, ist ebenfalls abhängig von den im ersten Schritt definierten funktionellen Anforderungen und den im zweiten Schritt definierten Beziehungen zwischen den Ressourcen. Ob und welches Encoding für einzelne Properties gilt, wird in dem die Property beschreibenden Aussagenschema spezifiziert. Dabei wird zwischen Syntax Encoding Schemes und Vocabulary Encoding Schemes unterschieden.

Ein Syntax Encoding Scheme kann nur dann verwendet werden, wenn es sich bei dem Wert, der für die Beschreibung einer Ressource verwendet wird, um einen Literal Value handelt. Ein Literal Value repräsentiert den Wert in Form einer einfachen Zeichenkette, ohne auf weitere Informationen zu verweisen. Das Syntax Encoding Scheme regelt die Struktur dieser Zeichenkette, indem es die gelten Standards benennt.

Abbildung 6: Beispiele der im DC Lib AP geltenden Syntax Encoding Scheme¹⁹

Syntax Encoding Schemes of the DC-Lib AP

DCMI Box

Name: Box

URI: <http://purl.org/dc/terms/Box>

Definition: The set of regions in space defined by their geographic coordinates according to the DCMI Box Encoding Scheme.

Type of Term: <http://www.w3.org/2000/01/rdf-schema#Datatype>

See: <http://dublincore.org/documents/dcmi-box/>

DCMI Period

Name: Period

URI: <http://purl.org/dc/terms/Period>

Definition: The set of time intervals defined by their limits according to the DCMI Period Encoding Scheme.

Type of Term: <http://www.w3.org/2000/01/rdf-schema#Datatype>

See: <http://dublincore.org/documents/dcmi-period/>

Ein Vocabulary Encoding Scheme schränkt die möglichen Werte ein, die mit einer bestimmten Property verwendet werden dürfen, indem es der Property ein oder mehrere kontrollierte Vokabulare zuweist. Dies hat zur Folge, dass bei der Beschreibung einer Ressource mit dieser bestimmten Property nur Werte aus den im Vocabulary Encoding Scheme genannten Vokabularen verwendet werden dürfen.

Abbildung 7: Beispiele der im DC Lib AP geltenden Vocabulary Encoding Scheme

Vocabulary Encoding Schemes of the DC-LIB AP

AAT

Name: AAT

URI: <http://purl.org/dclib/terms/AAT>

Type of Term: <http://purl.org/dc/dcam/VocabularyEncodingScheme>

Comment: Art and Architecture Thesaurus

See: http://www.getty.edu/research/conducting_research/vocabularies/aat/

ISO 639-3

Name: ISO639-3

URI: <http://purl.org/dc/terms/ISO639-3>

Definition: The set of three-letter codes listed in ISO 639-3 for the representation of names of languages.

Type of Term: <http://www.w3.org/2000/01/rdf-schema#Datatype>

See: <http://www.sil.org/iso639-3/>

In der Regel steht die Zeichenkette, die in diesem Zusammenhang verwendet wird, nicht für sich, sondern für eine weitere Ressource, die in dem kontrollierten Vokabular beschrieben wird. Ist dies der Fall, so sollte die Zeichenkette ein URI sein, der auf die Beschreibung des Terms im kontrollierten Vokabular verlinkt.

¹⁹ http://wiki.dublincore.org/index.php/DCLib_AP/Encoding_Schemes

Encodings Scheme nehmen somit Einfluss darauf, ob der mit einer Property verwendete Wert ein literal oder non-literal value ist. Wird eine Property durch ein Syntax Encoding Scheme eingeschränkt, so muss es sich bei dem Wert um einen literal value handeln. Wird sie hingegen durch ein Vocabulary Encoding Scheme eingeschränkt, handelt es sich bei dem Wert in der Regel um einen non-literal value. Ob ein Encodings Scheme verwendet wird, hängt jedoch auch hier vor allem von den im ersten Schritt genannten funktionellen Anforderungen ab.

3.7 Siebenter Schritt: Wie wird ein Description Set Profile maschinenlesbar?

Informationen über die Syntax bzw. das Datenformat, in dem die auf dem Metadatenprofil basierenden Datensätze in der Anwendung gespeichert oder für den Metadatenaustausch zur Verfügung gestellt werden, sollten im Rahmen des Metdatenprofils ebenfalls dokumentiert sein. Bei der Wahl der Syntax ist vor allem darauf zu achten, dass sie die im Description Set Profile genannten Regeln adäquat wiedergeben kann und die DCAM-Konformität der Metadatenterme nicht verletzt. Verschiedene Richtlinien für die Umsetzung eines Metadatenprofils in eine Syntax sind unter den Spezifikationen der Dublin Core Metadata Initiative veröffentlicht:

- Expressing Dublin Core metadata using HTML/XHTML meta and link elements
<http://dublincore.org/documents/2008/08/04/dc-html/>
- Expressing Dublin Core metadata using the Resource Description Framework (RDF)
<http://dublincore.org/documents/dc-rdf/>
- Expressing Dublin Core Description Sets using XML (DC-DS-XML)
<http://dublincore.org/documents/2008/09/01/dc-ds-xml/>
- A MoinMoin Wiki Syntax for Description Set Profiles
<http://dublincore.org/documents/2008/10/06/dsp-wiki-syntax/>