

DEUTSCHE INITIATIVE FÜR NETZWERKINFORMATION E. V.



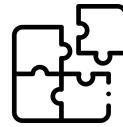
Einführung in die Gestaltung von Metadatenprofilen



Einführungen



Übungen




Lösungen

Zielgruppe:

Anwender und Entwickler von Metadatenprofilen mit Basis-Vorkenntnissen im Bereich Metadaten und XML.

Inhalt:

Die funktionellen Anforderungen von Informationsdiensten sind das Finden, Identifizieren und Selektieren von, sowie der Zugriff auf die für einen bestimmten Sachverhalt benötigten Informationen. Diese Aufgaben waren schon immer zentraler Bestandteil von Informationsdiensten und haben sich auch mit dem Internet nicht geändert. Geändert hat sich allerdings die technische Infrastruktur, die von den Informationssuchenden und Informationsanbietern genutzt wird. Nutzer fordern immer vehementer den "One-Stop-Shop", der es ihnen ermöglicht, die unterschiedlichsten Dienste über eine gemeinsame Oberfläche zu nutzen.



Dies erfordert jedoch die Vernetzung heterogener Informationsangebote und die Interoperabilität der verwendeten Daten und Metadatenformate. Nur wenn die Datenformate interoperabel sind, ist ein Mash-up heterogener Datenbestände möglich.

Dieses E-Tutorial stellt die im Dublin-Core-Kontext entwickelten Lösungsansätze zur interoperablen Gestaltung von Metadatenformaten vor. Ausgangspunkt ist das "Singapore Framework for Dublin Core Application Profiles".

Dieses Tutorial orientiert sich inhaltlich an der Schulung "Interoperable Gestaltung von Metadatenformaten".

Inhaltsangabe

Thema 1

Grundlagen

Metadaten und Interoperabilität
Metadatenprofile

Thema 2

Singapore Framework

Singapore Framework

Thema 3

In sieben Schritten zum Profil

Was will ich mit meinen Metadaten erreichen?

Was will ich beschreiben?

Welche Eigenschaften sind für meine Anwendung relevant?

Übung: Klassen und ihre Eigenschaften

Welche Metadatensterme brauche ich?

Übung: Metadatenstandards

Was für Regeln gelten für meine Metadatensterme?

Übung: Properties

Was für Kodierungsschemata gelten für die einzelnen Terme?

Übung: Kodierungsschemata

Übung: Literale und Non-Literale Inhalte

Wie wird mein Beschreibungssatzprofil maschinenlesbar?

Übung: In sieben Schritten zum Profil



Thema 1

Grundlagen

Metadaten und Interoperabilität

Was sind Metadaten?

Metadaten sind einheitlich strukturierte Daten, vor allem notwendig für die Beschreibung von Objekten in maschinenlesbarer Form. Im Grunde genommen kann man alles mit Metadaten beschreiben: Gegenstände, Konzepte, Dokumente, Forschungsdaten, Orte, Personen usw. Es gibt verschiedene Arten von Metadaten und mindestens so viele Typologien wie Standards. Die Definition von Metadatatypen ist i.d.R. abhängig von dem Kontext, in dem sie verwendet werden und von den Funktionen, die sie erfüllen sollen.



Wozu brauche ich Metadaten?

Metadaten dienen vor allem der maschinenlesbaren Beschreibung. Wir sehen uns heute häufig einer nicht mehr überschaubaren Menge an Objekten/Ressourcen gegenüber, aus der wir die für uns relevanten aussuchen müssen – wir suchen also zunehmend nach der Nadel im Heuhaufen. Um dies möglichst effizient zu gestalten, verwenden wir Metadaten. Sie helfen uns, die relevanten Objekte zu suchen, zu finden und schließlich auch benutzen zu können. Gleichzeitig vereinfachen sie den Austausch und die Nachnutzbarkeit der Objekte zwischen Menschen, Gruppen, Organisationen. Und schließlich ist das Suchen und Finden von Objekten im World Wide Web ohne Metadaten gar nicht möglich.

Die Metadaten, die in diesem Kontext verwendet werden, stehen dabei in einem ununterbrochenen Spannungsfeld zwischen den Eigenschaften der Objekte, den Anforderungen, die die Nutzer haben und den Möglichkeiten, die die Systeme bieten.

Die Eigenschaften der verschiedenen Objekte weichen mehr oder weniger voneinander ab und erfordern somit mehr oder weniger unterschiedliche Metadaten.

Verschiedene Nutzer haben unterschiedliche Anforderungen (stellenweise an ein- und dasselbe Objekt), die ebenfalls, meist kontextabhängig, unterschiedliche Metadaten erfordern.

Die Systeme, die Metadaten generieren, verwalten und speichern sollen, müssen in der Lage sein, Metadaten zu interpretieren und zur Verfügung zu stellen, was stellenweise weitere Metadaten erfordert.

Diese Vielzahl unterschiedlichster Metadaten lässt sich nur dann miteinander kombinieren, wenn die zugrunde liegenden Metadatenformate interoperabel sind.



Was ist Interoperabilität?

Interoperabilität ist die Fähigkeit, Daten zwischen verschiedenen Systemen ohne größeren Informationsverlust auszutauschen. Im Bereich der Metadaten wird dabei zwischen struktureller, syntaktischer und semantischer Interoperabilität unterschieden. Strukturelle Interoperabilität liegt dann vor, wenn den verschiedenen Metadatenformaten ein gemeinsames Datenmodell zugrundeliegt. Als Beispiel sei hier das Dublin Core Abstract Model genannt. Die syntaktische Interoperabilität erfordert hingegen eine gemeinsame Kodierungssyntax (z. B. XML).

Semantische Interoperabilität schließlich ist nur möglich, wenn ein gemeinsames Vokabular (z. B. Das Dublin Core Metadata Element Set) verwendet wird. Letzteres wird i.d.R. durch die Verwendung von Metadatenstandards ermöglicht.



Wozu brauche ich einen Metadatenstandard?

Ein Metadatenstandard beschreibt die Eigenschaften von Objekten, die in einer bestimmten Community oder auch communityübergreifend für die Beschreibung der Objekte verwendet werden sollen. Er sorgt damit für die Konsistenz der Metadaten in den verschiedenen Anwendungen einer Community und erleichtert den Austausch und die Verlinkung von Metadaten zwischen diesen Anwendungen. Metadatenstandards gibt es viele, aber nur wenige passen genau zu meinen Anforderungen. Vielleicht möchte ich einige der im Standard vorgeschlagenen Eigenschaften gar nicht nutzen, vielleicht einiges aus einem anderen Standard hinzufügen. Ist dies der Fall, so braucht es eine umfassende und nachvollziehbare Dokumentation, damit jeder, der meine Metadaten in einem anderen Kontext nutzen möchte, schnell erkennen kann, wie meine Metadatenanwendung aufgebaut ist. Eine solche Dokumentation erfolgt in Form eines Metadatenprofils.



Metadatenprofile

Definitionen

Es gibt verschiedene Definitionen, die in einem Satz beschreiben, was ein Metadatenprofil ist. Sie sind hierbei mal mehr und mal weniger aussagekräftig:

"... a declaration specifying which metadata term an organization, information provider, or user community uses in its metadata ... identifies the source of metadata terms..."

Quelle: <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/MMI-DC/cwa14855-00-2003-Nov.pdf>

"... a type of metadata schema ..."

(Quelle: <http://www.ariadne.ac.uk/issue25/app-profiles/>)

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

(Quelle: <http://dublincore.org/documents/singapore-framework/>)

Oder ein bisschen ausführlicher:

"Ein Metadatenprofil ist eine Spezifikation, die die in einer bestimmten Anwendung benötigten Metadaterme detailliert beschreibt unter Berücksichtigung der Anforderungen, die an die Anwendung gestellt werden und der Objekte, die mit den Metadaten beschrieben werden. Welche Komponenten für die Gestaltung eines Metadatenprofils notwendig sind, beschreibt das "Singapore Framework for Dublin Core Application Profiles"

(Quelle: http://www.kim-forum.org/Subsites/kim/DE/Materialien/Glossar/glossar_node.html)



Dokumentieren für Menschen

Ein Metadatenprofil ist vor allem eine Dokumentation, die die folgenden Fragen beantworten soll:

Für wen? - Das Metadatenprofil nennt die Zielgruppe der Metadatenanwendung, für die das Profil gilt.

Wozu? - Das Metadatenprofil nennt Aufgaben und Nutzungsszenarien, die die Metadaten in der Anwendung erfüllen müssen.

Was? - Das Metadatenprofil definiert die Objekt/Ressourcen, die mit den Metadaten beschrieben werden sollen.

Womit? - Das Metadatenprofil nennt die Metadaterme, die für die Beschreibung der Objekte/Ressourcen verwendet werden, die Standards, die diesen Termen zugrundeliegen und die Regeln, die für die Verwendung dieser Terme gelten.



Menschenlesbar:

Label	LiLa AP name of the term
URI	Specifies the URI of the given term
Term name	(Dublin Core) name of the term; abbreviation for the term URI
Defined by	Name and URI of the metadata vocabulary of the given term
Type of term	Specifies the type of the term according to the Dublin Core Metadata Initiative Abstract Model (DCAM) as class, property, syntax encoding scheme, vocabulary encoding scheme)
Refines	Specifies the property which is refined by the given term
Definition in LiLa AP	Specifies the definition of the given term in the LiLa AP context which will be based on the DC definitions in most cases
Comments and best practice examples	Recommendations on how to use this term
Encoding schemes	Name and URI of the syntax encoding scheme and/or the vocabulary encoding scheme, if applicable
Obligation	Specifies whether filling in this term is mandatory (m), recommended (r), or optional (o)
Repeatable	Specifies whether the term can be repeated
<i>Pica field for mapping</i>	<i>Specifies the Pica library catalogue field which corresponds to the LiLa AP field</i>
<i>LOM field for mapping</i>	<i>Specifies the LOM field which corresponds to the LiLa AP field</i>

Label	Subject
URI	http://purl.org/dc/elements/1.1/subject
Term name	dc:subject
Defined by	Dublin Core Metadata Element Set, Version 1.1, http://purl.org/dc/elements/1.1/
Type of term	Property
Refines	-
Definition in LiLa AP	Topic of the described LiLa content.
Comments and best practice examples	The topics of a LiLa content are specified using key words of the Library of Congress Subjects Headings (LCSH) and classifications of the Dewey Decimal Classification System (DDC). It is recommended to use a maximum of five key words. A separate statement should be used for each key word or each subject descriptor.
Encoding schemes	dcterms:LCSH (http://purl.org/dc/terms/LCSH) dcterms:DDC (http://purl.org/dc/terms/DDC)
Obligation	Mandatory
Repeatable	Yes
<i>Pica field for mapping</i>	
<i>LOM field for mapping</i>	<i>general.keywords. For those wishing more specificity of Subject, a category of classification can be used with a purpose of "Subject". classification has elements for description, keywords, and taxonpath(s) that are specific for the purpose.</i>



Metadatenprofile ... und Maschinen

Gleichzeitig muss ein Metadatenprofil die verwendeten Standards maschinenlesbar benennen. Das Profil sollte daher auch in Form eines maschinenlesbaren Schemas (z. B. eines xml-Schemas) vorliegen, das es einem Computer ermöglicht, Daten, die in diesem Format verarbeitet werden, zu erkennen und zu prüfen.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dcam="http://purl.org/dc/dcam/" >
  <rdf:Description rdf:about="http://dublincore.org/pages/home">
    <dcterms:title>DCMI Home Page</dcterms:title>
    <dcterms:publisher> <rdf:Description rdf:about="http://example.org/agents/DCMI"> <rdf:value>Dublin Core Metadata
  Initiative</rdf:value>
  </rdf:Description>
  </dcterms:publisher>
  <dcterms:subject rdf:parseType="Resource">
    <dcam:memberOf rdf:resource="http://purl.org/dc/terms/LCSH" />
    <rdf:value>Metadata</rdf:value>
  </dcterms:subject>
  </rdf:Description>
</rdf:RDF>

<?xml version="1.0" encoding="UTF-8" ?>
<dcds:descriptionSet xmlns:dcds="http://purl.org/dc/xmlns/2008/09/01/dc-ds-xml/">
  <dcds:description dcds:resourceURI="http://dublincore.org/pages/home">
    <dcds:statement dcds:propertyURI="http://purl.org/dc/terms/title">
      <dcds:literalValueString>DCMI Home Page</dcds:literalValueString>
    </dcds:statement>
    <dcds:statement dcds:propertyURI="http://purl.org/dc/terms/publisher" dcds:valueURI="http://example.org/agents/DCMI">
      <dcds:valueString>Dublin Core Metadata Initiative</dcds:valueString>
    </dcds:statement> <!-- vocabulary encoding scheme URI -->
    <dcds:statement dcds:propertyURI="http://purl.org/dc/terms/subject" dcds:vesURI="http://purl.org/dc/terms/LCSH">
      <dcds:valueString>Metadata</dcds:valueString>
    </dcds:statement>
  </dcds:description>
</dcds:descriptionSet>
```

Bild von einem XML-Schema (s. z. B. <http://www.loc.gov/standards/mods/mods.xsd>)



Mensch - Maschine

Es geht darum für Menschen und für Maschinen gleichermaßen Dokumente zu beschreiben. Die Informationen für den Menschen müssen in eine Form umgesetzt werden, mit der die Maschine umgehen kann.

Für Menschen geht es darum Objekte zu beschreiben, das kann z.B. eine Homepage sein. Hierbei hat jedes Objekt eine Vielzahl an Eigenschaften, die für die Beschreibung herangezogen werden können wie der Titel der Homepage oder das Erstellungsdatum. Man beschreibt das Objekt (hier die Homepage) indem man den verschiedenen Eigenschaften, die ein Objekt hat, einen Wert zuweist. Beispielsweise, dass die Homepage eine DCMI Home Page ist. Jedes Objekt gehört außerdem zu einer Klasse ähnlicher Objekte. So gehört ein Buch zur Klasse der Bücher, eine Person zur Klasse der Personen usw.

Für eine Maschine sieht die gleiche Beschreibung einer Homepage dagegen folgendermaßen aus:

```
<dods:description dods:resourceURI="http://dublincore.org/pages/home">
<dods:statement dods:propertyURI="http://purl.org/dc/terms/title">
<dods:literalValueString>DCMI Home Page</dods:literalValueString>
</dods:statement>
<dods:statement dods:propertyURI="http://purl.org/dc/terms/publisher" dods:valueURI="http://example.org/agents/DCMI">
<dods:valueString>Dublin Core Metadata Initiative</dods:valueString>
</dods:statement>
</dods:description>
```



Für die Verwendung der Werte, kann ich Regeln aufstellen, z. Bsp.

In dem Feld "subject" dürfen nur Begriffe aus dem LCSH verwendet werden.

Datumsangaben in dem Feld "date" müssen den W3C Vorgaben entsprechen.

Indem ich ein Objekt mit einer Eigenschaft und einem Wert beschreibe, mache ich eine Aussage über das Objekt

Vocabulary Encoding Scheme:

```
<dcds:statement dcds:propertyURI="http://purl.org/dc/terms/subject" dcds:vesURI="http://purl.org/dc/terms/LCSH">  
<dcds:valueString>Metadata</dcds:valueString>  
</dcds:statement>
```

Syntax Encoding Scheme:

```
<dcds:statement dcds:propertyURI="http://purl.org/dc/terms/date">  
<dcds:literalValueString dcds:sesURI="http://www.w3.org/2001/XMLSchema#date">2005-05-05</dcds:literalValueString>  
</dcds:statement>
```



Thema 2

Singapore Framework

„ ...a framework for designing metadata applications for maximum interoperability and for documenting such applications for maximum reusability. “

Das Singapore Framework ist ein Standard für die Dokumentation von Metadatenprofilen. Es beschreibt, wie ein Metadatenprofil aussehen muss, um Dublin-Core konform zu sein, indem es die Komponenten definiert, die für die Beschreibung eines Anwendungsprofils notwendig sind.

Verpflichtende Bestandteile eines Metadatenprofils sind laut dem Singapore Framework die "funktionellen Anforderungen", das "Domain Modell" und das "Beschreibungssatzprofil".

Unter den funktionellen Anforderungen (functional requirements) wird aufgelistet, welche Funktionalitäten durch die Metadaten erfüllt werden sollen.

Das Domain Modell (domain model) definiert die Objekte, die mit den Metadaten beschrieben werden und die Beziehungen, die zwischen diesen Objekten bestehen.

Im Beschreibungssatzprofil (description set profile) werden die Metadaterme beschrieben, die notwendig sind, um die funktionellen Anforderungen umzusetzen zu können.



Neben diesen verpflichtenden Bestandteilen empfiehlt das Singapore Framework als weitere Bestandteile eines Metadatenprofils:

Regeln für die Verwendung der Terme

Die syntaktische Kodierung der Terme

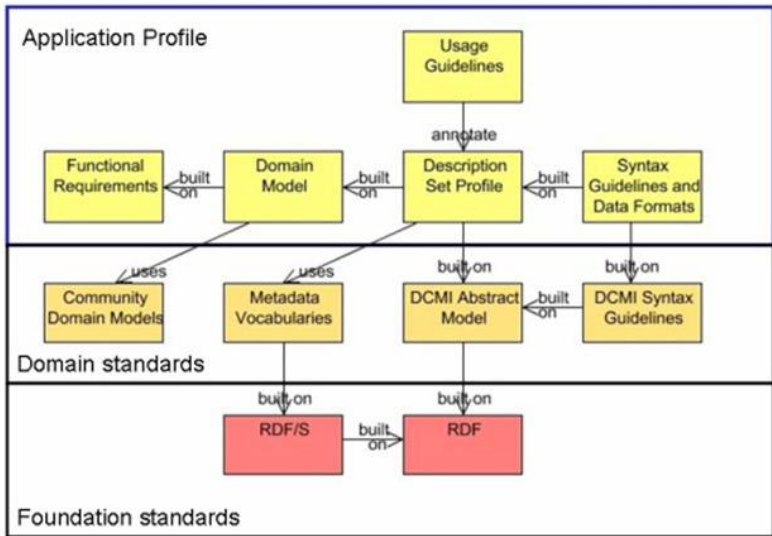
Das Singapore Framework ermöglicht die Kombination von verschiedenen Metadatenstandards (z. B. können Elemente der DCMES mit Elementen aus SKOS kombiniert werden) und mit domainspezifischen Metadaten termen. Voraussetzung dafür ist allerdings, dass alle verwendeten Terme über einen URI identifizierbar sind. Damit unterstützt das Singapore Framework Interoperabilität von Metadatenanwendungen

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

Informationen zu Metadatenstandards finden Sie in dem Tutorial "Einführung in Metadaten und Metadatenformate".



Das Singapore Framework sieht grafisch dargestellt folgendermaßen aus:



(Quelle: <http://dublincore.org/documents/singapore-framework/>)



Was bedeutet DCAM-konform?

Das DCMI Abstract Model (DCAM) ist ein syntaxunabhängiges Meta-Modell, das auf dem RDF Modell basiert und die syntaxübergreifende Konvertierung von Metadaten ermöglicht. DCAM definiert die Begriffe, die benötigt werden, um Informationsstrukturen und insbesondere die Beziehungen innerhalb dieser Strukturen auf der Metaebene zu beschreiben. Dabei unterscheidet DCAM verschiedene Typen von Metadatermen, als da sind Eigenschaften (Properties), Klassen (Classes), syntaktische Kodierungsschemas (Syntax Encoding Scheme) sowie semantische Kodierungsschemas (Vocabulary Encoding Scheme). Ein Metadaterm ist dann DCAM-konform, wenn er explizit einem dieser Term-Typen zugewiesen wurde sowie über einen den Term identifizierenden URI verfügt.



Thema 3

1. Schritt: Was will ich mit meinen Metadaten erreichen?

Zielsetzung und Geltungsbereich

Bevor ich darüber nachdenke, was für Metadaterme ich benötige, muss ich zunächst genau beschreiben, was mit den Metadaten in meiner Anwendung eigentlich erreicht werden soll, welche funktionellen Anforderungen ich an die Metadaten habe. Dabei ist zu unterscheiden zwischen meinen Zielen als Dienstanbieter und den Anforderungen derer, die meine Anwendung nutzen sollen. Folgende Fragen muss ich daher erst einmal beantworten:

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, community-übergreifend, interdisziplinär, ...)?

Werden die Metadaten in einen wissenschaftlichen Kontext eingebunden (z. B. zur Beschreibung von Forschungsdaten, wissenschaftlicher Literatur usw.) oder in einen ökonomischen 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 sich in Zukunft um das Profil kümmern, es verfügbar machen, es pflegen?

Wie sieht meine Zielgruppe aus?

StudentInnen, WissenschaftlerInnen, SchülerInnen, Fußballfans, SachbearbeiterInnen im Einwohnermeldeamt, usw.?

deutschsprachig, englischsprachig, mehrsprachig usw.?



Beispiel zu Zielsetzung und Geltungsbereich

Zielsetzung und Geltungsbereich der DC-Lib AP für Bibliographische Text-Ressourcen

Angelehnt an das Singapore Framework ist das Dublin Core Application Profile ein Dokument oder ein Set aus Dokumenten, die eine Metadatenanwendung beschreiben. Es werden folgende Punkte beschrieben:

welche Terme aus welchen Namensräumen werden benutzt (Description Set Profile)

welche Entitäten mit welchen Elementen beschrieben werden (Domain Model)

und welche Funktionalitäten die Applikation benötigt (Functional Requirements)

Geltungsbereich dieses Profils ist traditionell bibliothekarisches Textmaterial - veröffentlicht oder unveröffentlicht - wie Monographien, fortlaufende Publikationen (Zeitschriften, Jahrbücher, usw.), Artikel, Preprints und Postprints, aber auch digitalisierte oder digitale Versionen dieser Materialien. Diese Materialien werden Bibliographische Textressourcen genannt.

Zielsetzung dieses Application Profile ist ein Dublin Core Application Profile, das die Bedürfnisse von Bibliotheken aufnimmt und es ermöglicht die Bibliographischen Textressourcen zu beschreiben. Außerdem soll das AP die Publikation dieser Metadaten als Linked Data unterstützen.

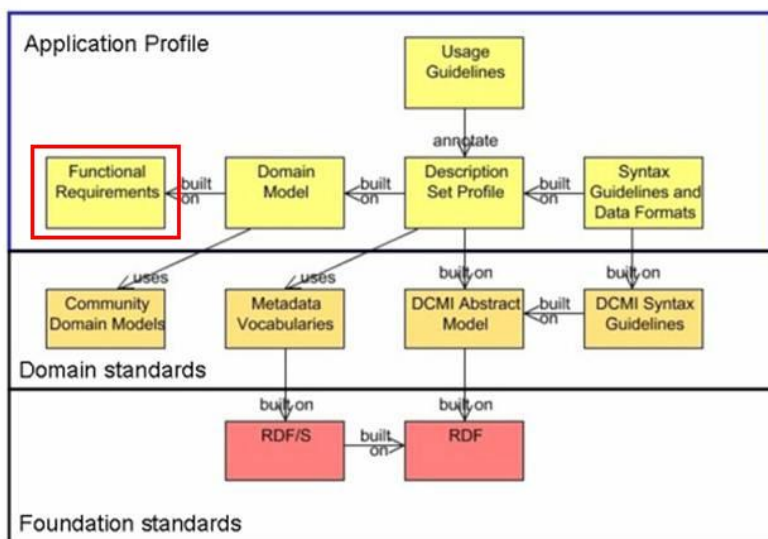
(Quelle: http://wiki.dublincore.org/index.php/DCLib_AP)



Funktionale Anforderungen

Die Funktionale Anforderungen gehören zu den verpflichtenden Komponenten des Singapore Framework. Sie bilden die Grundlage des Metadatenprofils. Alle anderen Komponenten sind von dieser Komponente abhängig.

Bei den funktionalen Anforderungen handelt es sich um einen Baustein des "Description Set Profile" (DSP) bzw. das DSP baut auf den "Functional Requirements" auf.



(Quelle: http://www.kim-forum.org/material/pdf/uebersetzung_singapore_20090213.pdf)

Um die funktionalen Anforderungen näher zu spezifizieren, muss ich mir überlegen, in welcher Form die Metadaten die Zielgruppe der Anwendung unterstützen sollen, welche Nutzungsszenarien für meine Anwendung relevant sind.



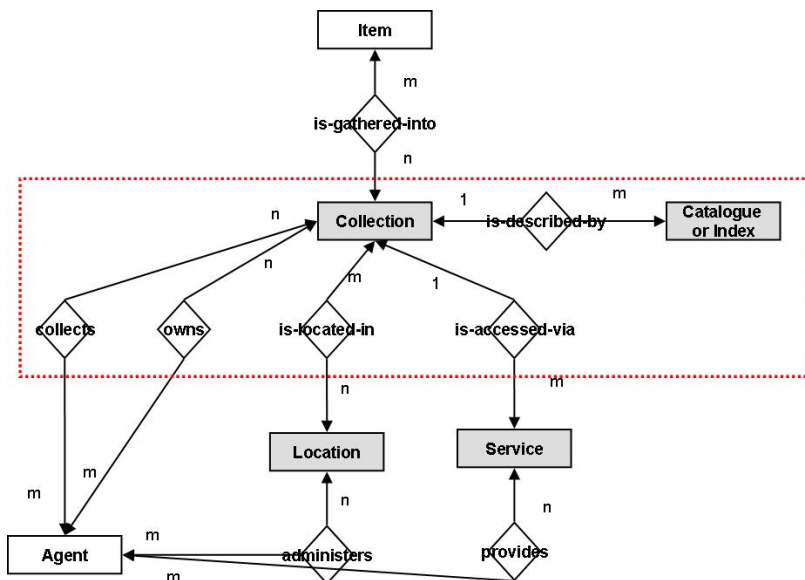
Sollen sie dazu dienen

- Informationen für eine Diplomarbeit zu recherchieren?
- den neuen Roman des Lieblingsautors auszuleihen?
- nach Programmen für Datenanalysen zu browsen?
- die Ergebnisse des letzten HSV-Spiels anzusehen?
- Wahlbenachrichtigungen an die Bürgerinnen und Bürger einer Gemeinde zu verschicken?
- usw.

Auf der Grundlage dieser Nutzungsszenarien spezifiziere ich funktionelle Anforderungen, die ich nach Wichtigkeit für meine Anwendung priorisiere

Beispiel: Dublin Core Collections Application Profile

<http://dublincore.org/groups/collections/collection-application-profile/>





Beispiele zu Funktionalen Anforderungen

Funktionale Anforderungen des DC Collections AP

Das DC Collection AP ist dazu gedacht, einfache Beschreibungen von Sammlungen und Katalogen zu unterstützen oder für eine breite Spanne von Sammlungen genutzt zu werden. Es ist in erster Linie gestaltet um das Auffinden und die Auswahl von Sammlungen zu unterstützen. Es ist nicht dazu da jede Art von Sammlung in all ihren Charakteristika zu beschreiben.

Das Ziel ist, dass das DC Collection AP folgende Funktionen unterstützen soll:

Die Entdeckung von Sammlungen und Katalogen durch die Möglichkeit der Suche nach verschiedenen Attributen wie z.B.

den Namen oder Titel einer Sammlung oder eines Katalogs oder Index

das Thema einer Sammlung

den Media Type oder das Format eines Mediums

Beziehungen zwischen Sammlungen, zwischen Katalogen, zwischen Sammlungen und Katalogen usw.

die Identifikation einer bekannten Sammlung oder eines Katalogs

die Auswahl einer oder mehrerer Sammlungen oder Kataloge aus einer größeren Anzahl aus gefundenen Sammlungen oder Katalogen

das Identifizieren des Ortes, an dem sich der Katalog oder die Sammlung befindet

das Identifizieren des Services, der den Katalog oder die Sammlung anbietet

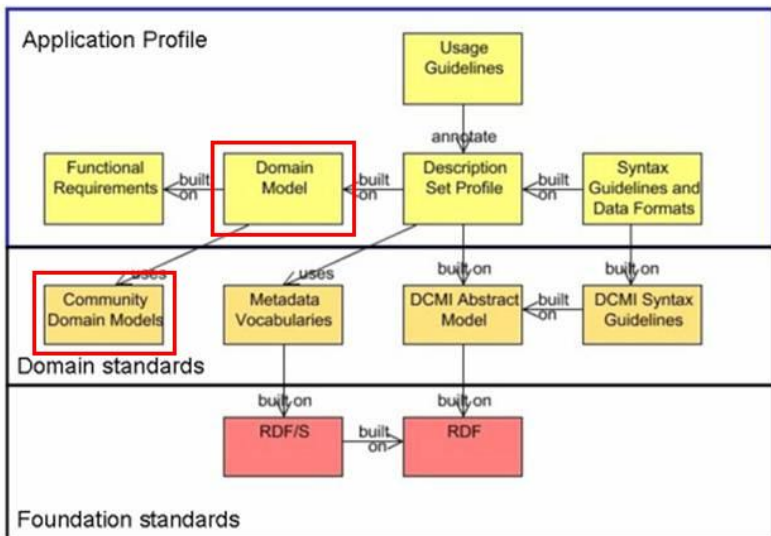
(Quelle: <http://dublincore.org/groups/collections/collection-application-profile/index.shtml>)



2. Schritt: Was will ich beschreiben?

Objekt Klassen

Im nächsten Schritt sehe ich mir die Objekte, die ich mit meiner Anwendung verfügbar machen will, einmal genau an und beschreibe sie in Form eines Domain-Modells.





Was für Objekte will ich beschreiben?

Bücher, Filme, Internetressourcen, Verfasser, Herausgeber, Software, Hardware, Orte usw.?

Welchen Klassen kann ich diese Objekte zuordnen?

Bücher, Filme und Internetressourcen können zur Klasse „Medien“ gehören,

Verfasser und Herausgeber können zur Klasse „Personen“ oder zur Klasse „Organisationen“ gehören,

Software und Hardware können zur Klasse „technische Voraussetzungen“ gehören,

Orte können zur Klasse „geographische Angaben“ gehören usw.

Die Zuordnung zu verschiedenen Klassen ist abhängig von der Granularität, mit der ich meine Objekte beschreiben will und diese wiederum ist bedingt durch die Anforderungen, die meine Zielgruppe an die Anwendung hat.



Beispiel zu Objekt Klassen

Wenn man ein Domain Modell erstellt muss man sich über zwei Dinge Gedanken machen bzw. zwei Dinge definieren:

die Objekte, die in dem Metadatenprofil beschrieben werden sollen

die Beziehungen, die zwischen diesen Objekten bestehen sollen

Im Folgenden ein Beispiel für ein einfaches Domain Modell.

Beispiele zu Objekt Klassen II

Die selbstdefinierten Objekte sind gleichbedeutend mit Klassen:



Figuren



Bücher



Organisationen



Personen



Das dazugehörige Template sieht dann so aus:

Beschreibungssatzprofil
Beschreibungstemplate Label: Buch URI: http://example.org/myExample/entityType/Buch Definition: Type of Term: Class
Beschreibungstemplate Label: Organisation URI: http://example.org/myExample/entityType/Organisation Definition: Type of Term: Class
Beschreibungstemplate Label: Person URI: http://example.org/myExample/entityType/Person Definition: Type of Term: Class
Beschreibungstemplate Label: Figur URI: http://example.org/myExample/entityType/Protagonist Definition: Eine fiktive Persönlichkeit Type of Term: Class



Objekt Eigenschaften und Relationen

Was für Eigenschaften haben die Objekte?

Titel, Verfasser, Seitenzahlen, Abspieldauer, Signatur, URL, Name, Standort, Speicherkapazitäten, Anbieter, Alter, Adresse, technische Voraussetzungen usw.

Welche dieser Eigenschaften können dazu dienen, Beziehungen zwischen Objekten zu beschreiben?

Verfasser: Buch wird verfasst von Autor

Anbieter: Hardware wird zur Verfügung gestellt von Organisation

Technische Voraussetzungen: Programm wird verwendet mit Hardware

Adresse: Person ist wohnhaft in geographische Angabe

usw.



Beispiele zu Objekt Eigenschaften und Relationen

Hier ein Beispiel für ein Domain Modell.

Die Objekte, um die es in diesem Beispiel gehen soll, sind:



Figuren



Bücher



Organisationen



Personen

(Quellen: http://upload.wikimedia.org/wikipedia/de/thumb/8/8b/The_WaltDisney_Company-Logo.svg/656px-The_WaltDisney_Company-Logo.svg.png, http://www.welt.de/multimedia/archive/00692/Mickey_grosse_DW_Ve_692442a.jpg, <http://images.wikia.com/locopedia/images/d/d6/Minnie7.gif>, <http://www.vintagebooks.de/delphin/Micky-Maus-Buch-04.jpg>, <http://www.leadership-with-you.com/images/waltdisney.jpg>)

Die Beziehungen sind:



(Quellen: http://upload.wikimedia.org/wikipedia/de/thumb/8/8b/The_WaltDisney_Company-Logo.svg/656px-The_WaltDisney_Company-Logo.svg.png, http://www.welt.de/multimedia/archive/00692/Mickey_grosse_DW_Ve_692442a.jpg, <http://images.wikia.com/locopedia/images/d/d6/Minnie7.gif>, <http://www.vintagebooks.de/delphin/Micky-Maus-Buch-04.jpg>, <http://www.leadership-with-you.com/images/waltdisney.jpg>)



3. Schritt: Welche Eigenschaften sind für meine Anwendung relevant?

Eigenschaften und funktionale Anforderungen

Nun überlege ich mir, welche der im 2. Schritt genannten Eigenschaften und Beziehungen ich für die Beschreibung meiner Objekte benötige, um den Anforderungen meiner Nutzer gerecht zu werden.

Welche Eigenschaften und Beziehungen sollen in meiner Anwendung „suchbar“ sein?

Welche Eigenschaften und Beziehungen sollen in meiner Anwendung „angezeigt“ werden?

Welche Eigenschaften und Beziehungen werden benötigt, damit der Nutzer auf das Objekt „zugreifen“ kann?

Für die Stichwortsuche nach Namen, Titeln und Themen müssten die Eigenschaften „Name“, „Titel“ und „Thema“ in meiner Anwendung suchbar sein. Will ich weitere Eigenschaften über ein Browsing nutzbar machen, so bedarf es eines kontrollierten Vokabulars, über das ich entsprechende Listen aufbauen kann. Zudem gibt es Eigenschaften, die den Nutzer bei der Selektion der für ihn relevanten Objekte unterstützen, die für die Suche aber vielleicht gar nicht benötigt werden. Dies können z. B. Seitenzahlen oder technische Voraussetzungen sein. Schließlich braucht es Informationen für den Zugriff auf die Objekte wie z. B. der Identifier des Objekts und/oder der Name des Anbieters, über den man das Objekt erhalten kann.

Auch hier gilt, dass die Wahl der benötigten Eigenschaften in erster Linie die Anforderungen der Zielgruppe widerspiegeln muss. Außerdem ist zu beachten, dass Eigenschaften und Beziehungen, die im 2. Schritt genannt wurden, aber hier nicht als notwendig erachtet werden, in den folgenden Schritten nicht mehr berücksichtigt werden.



Eigenschaften und funktionale Anforderungen - Beispiel

Beispiel:



hat einen Namen
hat eine Adresse
hat Freunde
hat einen Schöpfer
hat ein Geschlecht



hat einen Nachnamen
hat einen Vornamen
hat ein Geburtsdatum
hat ein Geburtsland



hat einen Namen
hat einen Gründer
hat ein Gründungsdatum



hat einen Titel
hat eine Sprache
hat einen Verleger
hat Figuren



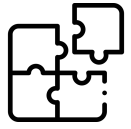
Übung: Klassen und ihre Eigenschaften

Quiz

Die Lösungen finden Sie auf der nächsten Seite

Welche Basis-Eigenschaften haben Mickey und Minnie Mouse?

- a) Hat einen Namen
- b) Hat einen Titel
- c) Hat eine Adresse
- d) Hat Freunde
- e) Hat einen Verleger
- f) Hat einen Schöpfer
- g) Hat Figuren
- h) Hat ein Geschlecht



Lösung:

Welche Basis-Eigenschaften haben Mickey und Minnie Mouse?

- a) Hat einen Namen
- b) Hat einen Titel
- c) Hat eine Adresse
- d) Hat Freunde
- e) Hat einen Verleger
- f) Hat einen Schöpfer
- g) Hat Figuren
- h) Hat ein Geschlecht



Quiz

Welche Basis-Eigenschaften hat das Mickey-Mouse-Buch?

Die Lösungen finden Sie auf der nächsten Seite

- a) Hat einen Titel
- b) Hat einen Namen
- c) Hat eine Sprache
- d) Hat einen Verleger
- e) Hat ein Geburtsdatum
- f) Hat Freunde
- g) Hat Figuren



Lösung

Welche Basis-Eigenschaften hat das Mickey-Mouse-Buch?

- a) Hat einen Titel
- b) Hat einen Namen
- c) Hat eine Sprache
- d) Hat einen Verleger
- e) Hat ein Geburtsdatum
- f) Hat Freunde
- g) Hat Figuren

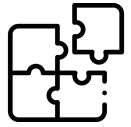


Quiz

Welche Basis-Eigenschaften hat Walt Disney?

Die Lösungen finden Sie auf der nächsten Seite

- a) Hat einen Nachnamen
- b) Hat einen Verleger
- c) Hat einen Vornamen
- d) Hat einen Schöpfer
- e) Hat ein Geburtsdatum
- f) Hat ein Geburtsland
- g) Hat einen Gründer



Lösung

Welche Basis-Eigenschaften hat Walt Disney?

- a) Hat einen Nachnamen
- b) Hat einen Verleger
- c) Hat einen Vornamen
- d) Hat einen Schöpfer
- e) Hat ein Geburtsdatum
- f) Hat ein Geburtsland
- g) Hat einen Gründer



Quiz

Welche Basis-Eigenschaften hat The Walt Disney Company?

Die Lösungen finden Sie auf der nächsten Seite

- a) Hat einen Nachnamen
- b) Hat einen Namen
- c) Hat einen Gründer
- d) Hat einen Verleger
- e) Hat Freunde
- f) Hat ein Gründungsdatum
- g) Hat ein Geburtsdatum



Lösung

Welche Basis-Eigenschaften hat The Walt Disney Company?

- a) Hat einen Nachnamen
- b) **Hat einen Namen**
- c) **Hat einen Gründer**
- d) Hat einen Verleger
- e) Hat Freunde
- f) **Hat ein Gründungsdatum**
- g) Hat ein Geburtsdatum



4. Schritt

Welche Metadaterme brauche ich?

Eigenschaften und Metadatenstandards

Unter Berücksichtigung der im 3. Schritt ausgewählten Eigenschaften und des Kontexts, in dem meine Anwendung steht, prüfe ich zunächst, welcher bereits bestehende Metadatenstandard für meine Anwendung geeignet ist.

Beispiele:

Ich will Bücher beschreiben:

DC Lib AP (<http://dublincore.org/documents/library-application-profile/index.shtml>)

TEL (http://www.theeuropeanlibrary.org/portal/organisation/cooperation_old/archive/telproject_archive/tel_ap_v1.3.html)

Ich will Internetressourcen beschreiben:

DCMI Metadata Terms (<http://dublincore.org/documents/dcmi-terms/>)

Ich will Computer-Ressourcen und/oder Programme beschreiben

CWM (http://www.omg.org/technology/documents/modeling_spec_catalog.htm)

GLUE (<http://glueschema.forge.cnaf.infn.it/Main/HomePage>)

Ich will Personen beschreiben

FOAF (<http://xmlns.com/foaf/spec/>)



Auswahl der Properties

Dann prüfe ich, welche Terme ich aus den einzelnen Standards für meine konkrete Anwendung benötige, d.h. welche Terme die von mir benötigten Eigenschaften adäquat wiedergeben. Dabei kann ich bei Bedarf auf Terme unterschiedlicher Standards zugreifen, solange diese DCAM-konform sind. DCAM-Konformität ist dann gegeben, wenn sich die Metadaterme den im Abstract Modell genannten Term-Typen (Property, Class, Syntax Encoding Scheme, Vocabulary Encoding Scheme) zuweisen lassen und außerdem durch einen URI identifizierbar sind.

Beispiel

DCMI Metadata Terms (<http://dublincore.org/documents/dcmi-terms/>)

Term Name: accessRights

Term Name: accessRights	
URI:	http://purl.org/dc/terms/accessRights
Label:	Access Rights
Definition:	Information about who can access the resource or an indication of its security status.
Comment:	Access Rights may include information regarding access or restrictions based on privacy, security, or other policies.
Type of Term:	Property
Refines:	http://purl.org/dc/elements/1.1/rights
Refines:	http://purl.org/dc/terms/rights
Has Range:	http://purl.org/dc/terms/RightsStatement
Version:	http://dublincore.org/usage/terms/history/#accessRights-002



Beispiel

FOAF (<http://xmlns.com/foaf/spec/>)

http://xmlns.com/foaf/spec/#term_family_name

Property: foaf:family_name

Property: foaf:familyName

familyName - The family name of some person.

Status: testing

Domain: having this property implies being a [Person](#)

The [familyName](#) property is provided (alongside [givenName](#)) for use when describing parts of people's names. Although these concepts do not capture the full range of personal naming styles found world-wide, they are commonly used and have some value.

There is also a simple [name](#) property.

Support is also provided for the more archaic and culturally varying terminology of [firstName](#) and [lastName](#).

See the [issue tracker](#) for design discussions, status and ongoing work on rationalising the FOAF naming machinery.



Benötige ich für meine Anwendung Eigenschaften, die sich durch keinen Standard abdecken lassen, so muss ich eigene Terme entwickeln. Aber auch diese Terme kann ich nur dann mit den Termen der verschiedenen Standards kombinieren, wenn sie DCAM-konform sind.

Beispiel

SWAP (<http://dublincore.org/scholarwiki/SWAPDSP>)

Affiliated Institution

Term URI: <http://purl.org/eprint/terms/affiliatedInstitution>

Type of Term: Property

Affiliated Institution	
Property	http://purl.org/eprint/terms/affiliatedInstitution
Literal?	No
Definition	An organisation to which a creator of the eprint is affiliated.
Eprint-specific recommendation	Use this <i>property</i> to provide the affiliated organisation's name and/or the URI of the affiliated institution and/or to link to a <i>related description</i> (with the <i>description set</i>) about the affiliated institution. Where a name of the affiliated organisation is provided, see A note about the form of personal and organisational names used in value strings .
Value (Non-Literal)	Description: agent Vocabulary Encoding Scheme Constraint Occurrence: optional Value String Constraint: Max occurrence: 1 Syntax Encoding Syntax Constraint: Occurrence: disallowed Language Constraint: Occurrence: optional

Die Dokumentation der verwendeten Terme erfolgt im Beschreibungsprofil.

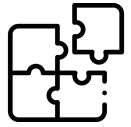


Quiz

Die Lösungen finden Sie auf der nächsten Seite

Welche Metadatenstandards für Bücher wurden genannt?

- a) DC Lip AP
- b) DCMI Metadata Terms
- c) FOAF
- d) CWM
- e) TEL
- f) GLUE



Lösung

Welche Metadatenstandards für Bücher wurden genannt?

- a) **DC Lip AP**
- b) DCMI Metadata Terms
- c) FOAF
- d) CWM
- e) **TEL**
- f) GLUE



Quiz

Die Lösungen finden Sie auf der nächsten Seite

Welche Metadatenstandards für Personen wurden genannt?

- a) DC Lip AP
- b) DCMI Metadata Terms
- c) FOAF
- d) CWM
- e) TEL
- f) GLUE



Lösung

Welche Metadatenstandards für Bücher wurden genannt?

- a) DC Lip AP
- b) DCMI Metadata Terms
- c) **FOAF**
- d) CWM
- e) TEL
- f) GLUE



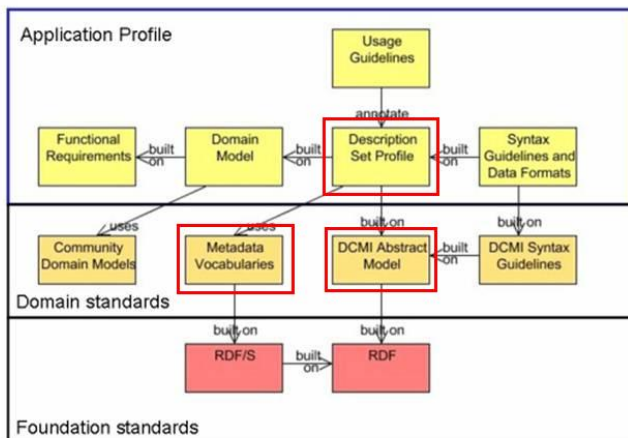
Schritt 5

Was für Regeln gelten für meine Metadaterme?

Definition der Klassen

Jetzt beschreibe ich die von mir ausgewählten Terme im Beschreibungssatzprofil. Das Beschreibungssatzprofil (BSP) dokumentiert, welche Metadaterme innerhalb meiner Anwendung Verwendung finden mit allen dazugehörigen Regeln. Dabei muss für jede Objekt-Klasse, die ich im Rahmen des Domain-Modells definiert habe, ein eigenes Beschreibungsschema erstellt werden. Das Beschreibungsschema definiert die Klasse und deklariert, was für Aussagen über diese Klasse gemacht werden.

Das Beschreibungssatzprofil (Description Set Profile) baut auf den Funktionalen Anforderungen und dem Domain Modell auf. Außerdem auch auf dem DCMI Abstract Modell und es nutzt verschiedene Metadaten Vokabularien.





Definition einer Klasse am Beispiel FOAF

(<http://xmlns.com/foaf/spec/>)

http://xmlns.com/foaf/spec/#term_Person

Class: foaf:Person

Person - A person.

Status: stable

Properties include: [myersBriggs](#) [familyName](#) [publications](#) [lastName](#) [family_name](#) [firstName](#) [currentProject](#) [surname](#) [knows](#) [workInfoHomepage](#) [pastProject](#) [geekcode](#) [schoolHomepage](#) [workplaceHomepage](#) [img](#) [plan](#)

Used with: [knows](#)

Subclass Of [Agent](#) [Spatial Thing](#) [Person](#)

Disjoint With: [Organization](#) [Project](#)

The [Person](#) class represents people. Something is a [Person](#) if it is a person. We don't nitpic about whether they're alive, dead, real, or imaginary. The [Person](#) class is a sub-class of the [Agent](#) class, since all people are considered 'agents' in FOAF.



Definition der Properties

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 in diesem Kontext notwendigen Kodierungsschemas sowie alle weiteren Regeln, die für eine Aussage in diesem bestimmten Kontext gelten.

Definition einer Aussage am Beispiel von FOAF

`http://xmlns.com/foaf/spec/#term_family_name`

Property: foaf:family_name

family_name - The family name of some person.

Status: archaic

Domain: having this property implies being a [Person](#)

This property is considered an *archaic* spelling of [familyName](#).

A number of naming constructs are under development to provide naming substructure; draft properties include foaf:firstName, foaf:givenname and foaf:surname. These are not currently stable or consistent; see the issue tracker for design discussions, status and ongoing work on rationalising the FOAF naming machinery.

There is also a simple foaf:name property.



Attribute zur Beschreibung von Metadatermen

Folgende Attribute sollten verwendet werden, um einen Metadaterm, der für eine Eigenschaft steht, zu beschreiben:

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, dass es sich um eine Eigenschaft (Property) handelt.

Occurence: gibt an, ob und wie oft der Term verwendet werden darf/muss

Refinement: gibt an, ob es sich ggf. um eine Sub-Eigenschaft (Sub-Property) handelt

Type of Value: gibt an, ob der Eintrag, der bei der Beschreibung des Objekts erfolgt, für sich steht (ein sogenannter literaler Wert) oder ob dieser Eintrag auf die Beschreibung eines weiteren Objekts verweist (ein sogenannter nicht-literaler Wert).

has Syntax Encoding Scheme: gibt an, welche Regeln für einen literalen Wert gelten

has Vocabulary Encoding Scheme: gibt an, welches kontrollierte Vokabular für einen nicht-literalen Wert verwendet wird.

Es ist zu beachten, dass die für die Aussagen geltenden Regeln und Kodierungsschemas den in den funktionellen Anforderungen genannten Systemfunktionen entsprechen und die im Domain-Modell genannten Beziehungen zwischen den Objekten adäquat wiedergeben.



Definition domainspezifischer Metadaterme

Da das Singapore Framework die Kombination von Termen aus verschiedenen Standards sowie die Kombination mit selbstentwickelten Termen erlaubt, kann ein Beschreibungsschema Metadaterme aus unterschiedlichsten Domänen beinhalten. Voraussetzung dafür ist allerdings die DCAM-Konformität der Terme, in diesem Fall also die Kennzeichnung als Property und die Identifizierung durch einen URI.

Die Kombination von Metadatenstandards und eigenen Termen am Beispiel von AGLS (<http://www.agls.gov.au/documents/aglsterms/>)

Verwendung eines DCMI Metadata Term

Term Name: accessRights

Term Name: accessRights	
URI:	http://purl.org/dc/terms/accessRights
Label:	Access Rights
Definition:	Information about who can access the resource or an indication of its security status.
Comment:	Access Rights may include information regarding access or restrictions based on privacy, security, or other policies.
Type of Term:	<u>Property</u>
Refines:	http://purl.org/dc/elements/1.1/rights
Refines:	http://purl.org/dc/terms/rights
Has Range:	http://purl.org/dc/terms/RightsStatement
Version:	http://dublincore.org/usage/terms/history/#accessRights-002



Verwendung eines AGLS Metadata Term

Term Name: accessMode

URI: <https://agls.gov.au/documents/aglsterms/>

Term Name:	accessMode
URI:	/agls/terms/accessMode
Label:	Access Mode
Definition:	Perceptual mode for the resource.
Comment:	There may be more than one such mode.
Type of Term:	Property
Refines:	/agls/terms/accessibility
Version:	/history/#accessMode-001

Label: Access Mode

Definition: Perceptual mode for the resource.

Comment: There may be more than one such mode.

Type of Term: Property

Refines: <http://www.agls.gov.au/agls/terms/accessibility>

Version: <http://www.agls.gov.au/history/#accessMode-001>

Die Definition der verwendeten Terme muss eindeutig sein, Metadataerme 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 Termen gibt, die in der Community verbreitet und anerkannt sind.



Standardterme einschränken

Schließlich kann ich die Verwendung von Termen, die ich aus Metadatenstandards übernehme, für den eigenen Gebrauch einschränken. Auch diese Einschränkung muss eindeutig definiert werden. Und ich muss darauf achten, dass meine Definition nicht der Semantik der Definition widerspricht, die dem Term im Rahmen des Standards zugewiesen wurde. Ich darf die Bedeutung eines Terms zwar jederzeit verfeinern, aber auf gar keinen Fall erweitern.

Beispiel für einen domainspezifischen Term:

Beschreibungssatzprofil

Beschreibungstemplate

Label: Buch

URI: <http://example.org/myExample/entityType/Buch>

Definition:

Type of Term: Class

Aussagentemplate

Label: Protagonist

URI: <http://example.org/myExample/protagonist>

Definition: Eine fiktive Persönlichkeit, die in einem literarischen Werk eine Rolle spielt

Type of Term: property

Occurence: 1/n

Literal / Non-Literal: Non-Literal

Refines: <http://purl.org/dc/terms/subject>

Has Vocabulary Encoding Scheme: <http://example.org/myExample/entityType/Figur>



Übung:

Properties

Die Lösungen finden Sie auf der nächsten Seite

Es gibt verschiedene Properties, die man dafür verwenden kann, das Erscheinungsdatum anzugeben. Man kann verwenden:

- oder
- oder
-

Um den Namen einer Ressource anzugeben gibt es ebenfalls mehrere Möglichkeiten:

- oder
- oder
-

Hilfe gibt es auf:

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



Lösung

Properties

Es gibt verschiedene Properties, die man dafür verwenden kann, das Erscheinungsdatum anzugeben. Man kann verwenden:

- dc:date oder
- dcterms:date oder
- dcterms:issued

Um den Namen einer Ressource anzugeben gibt es ebenfalls mehrere Möglichkeiten

- dc:title oder
- dcterms:title oder
- dc:alternative



Schritt 6

Was für Kodierungsschemas gelten für die einzelnen Terme?

Kodierungsschemas

Abhängig von den funktionellen Anforderungen und den im Domainmodell definierten Beziehungen weisen verschiedene Metadatermen sogenannte Kodierungsschemas zu. Ein Kodierungsschema (Encoding Scheme) nennt die Standards, die ggf. bei der Beschreibung der Eigenschaft eines Objekts berücksichtigt werden müssen. Ob und welche Kodierungsschemas für einzelne Terme gelten, wird in dem Beschreibungssatzprofil im jeweiligen Aussagenschema dokumentiert. Dabei wird zwischen syntaktischen Kodierungsschemas (Syntax Encoding Scheme) und semantischen Kodierungsschemas (Vocabulary Encoding Scheme) unterschieden.



Syntaktische Kodierungsschema

Ein syntaktisches Kodierungsschema (Syntax Encoding Scheme) kann nur dann verwendet werden, wenn es sich bei dem Wert, der für die Beschreibung der Eigenschaft eines Objekts verwendet wird, um einen literalen Wert, ein sogenanntes „Literal“ handelt. Ein „Literal“ repräsentiert den Wert in Form einer einfachen Zeichenkette, ohne auf weitere Informationen zu verweisen. Das syntaktische Kodierungsschema regelt die Struktur dieser Zeichenkette, indem es entsprechende Standards benennt.

Die Verwendung eines „Literal“ mit syntaktischem Kodierungsschema am Beispiel von SWAP (<http://dublincore.org/scholarwiki/SWAPDSP>)

Date Available	
Property	http://purl.org/dc/terms/available
Max occurrence	1
Literal?	Yes
Definition	Date (often a range) that the resource will become or did become available. The date that the described expression of the eprint was first made public.
Eprint-specific recommendation	Use a <i>value string</i> to provide the date, formatted according to the W3C Date Time Format (W3CDTF) specification.
Value (Literal)	Syntax Encoding Scheme:
	Occurrence mandatory
	Choose from http://purl.org/dc/terms/W3CDTF



Semantische Kodierungsschema

Ein semantisches Kodierungsschema (Vocabulary Encoding Scheme) schränkt die möglichen Werte ein, die ich zur Beschreibung einer Eigenschaft meines Objekts verwenden darf, indem es ein Vokabular (z. B. einen Thesaurus oder eine Klassifikation) benennt, dessen Terme ich benutzen muss.

Die Verwendung eines semantischen Kodierungsschemas am Beispiel des Dublin Core Collections Application Profile

(<http://dublincore.org/groups/collections/collection-application-profile/index.shtml>)

Subject [dc:subject]

Property URI	http://purl.org/dc/elements/1.1/subject
Qualified Name for Property	dc:subject
Defined By	The Dublin Core Metadata Element Set, v1.1 http://purl.org/dc/elements/1.1/
Type of Term	Property
Subproperty Of	[n/a]
Source Label	Subject and Keywords
Label in this DCAP	Subject
Source Definition	The topic of the resource.
Usage in this DCAP	A subject or topic associated with the items in the collection.
Comments for this DCAP	Keywords or subject descriptors associated with items in the collection. The terms used indicate the subject matter of the collection. The vocabulary encoding schemes below are those recommended by DCMI. Other appropriate vocabulary encoding schemes may be used, and the use of a scheme must be indicated. A value string must be provided; a value URI may also be provided. Where multiple keywords or subject descriptors are provided, a separate statement should be used for each keyword or descriptor.

Uses Vocabulary Encoding Scheme	dcterms:LCSH, Dublin Core Terms http://purl.org/dc/terms/LCSH			
	Value URI	Value String	Syntax Encoding Scheme(s)	Rich Represent
	Optional	Mandatory		Not permitted
Uses Vocabulary Encoding Scheme	dcterms:LCC, Dublin Core Terms http://purl.org/dc/terms/LCC			
	Value URI	Value String	Syntax Encoding Scheme(s)	Rich Represent
	Optional	Mandatory		Not permitted
Uses Vocabulary Encoding Scheme	dcterms:MESH, Dublin Core Terms http://purl.org/dc/terms/MESH			
	Value URI	Value String	Syntax Encoding Scheme(s)	Rich Represent
	Optional	Mandatory		Not permitted



Uses Vocabulary Encoding Scheme	dcterms:DDC, Dublin Core Terms http://purl.org/dc/terms/DDC			
	Value URI	Value String	Syntax Encoding Scheme(s)	Rich Represent
	Optional	Mandatory		Not permitted
Uses Vocabulary Encoding Scheme	dcterms:UDC, Dublin Core Terms http://purl.org/dc/terms/UDC			
	Value URI	Value String	Syntax Encoding Scheme(s)	Rich Represent
	Optional	Mandatory		Not permitted
Obligation	Optional			
Condition	[n/a]			
Minimum Occurrences	0			
Maximum Occurrences	unbounded			

Die Zeichenkette, die in diesem Zusammenhang für die Beschreibung der Eigenschaft verwendet wird, steht nicht für sich. Sie steht für ein weiteres Objekt, das im Rahmen des semantischen Kodierungsschemas beschrieben wird. Der Wert, den diese Zeichenkette repräsentiert, ist ein nicht-literaler Wert ein sogenanntes „Non-Literal“. Ein „Non-Literal“ ist ein Wert, der mittels eines URI auf ein anderes Objekt verweist, unabhängig davon, ob es sich um ein Objekt handelt, das im Rahmen eines semantischen Kodierungsschemas beschrieben wird oder nicht.



Literale und nicht-literale Werte

Die Entscheidung, einem Metadaten-term einen literalen oder einen nicht-literalen Wertebereich zuzuweisen, steht mir frei, solange ich keine Kodierungsschemas verwende. Entscheide ich mich jedoch für ein syntaktisches Kodierungsschema, so muss der Wertebereich als „literal“ deklariert werden. Entscheide ich mich hingegen für ein semantisches Kodierungsschema, so muss der Wertebereich als „nicht-literal“ deklariert werden. Ob ich überhaupt ein Kodierungsschema brauche und wenn ja, welches, ist letztlich abhängig von meinen funktionellen Anforderungen.



Übung

Kodierungsschemas

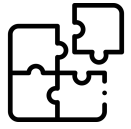
Bitte zuordnen, die Lösung finden Sie auf der nächsten Seite

Kodierungsschema, das Datumsangaben
näher spezifiziert

TGN

Kodierungsschema, das Werte für
geographische Angaben vorgibt

W3C DTF, Period



Lösung

Kodierungsschemas

Kodierungsschema, das Datumsangaben
näher spezifiziert = **W3C DTF, Period**

Kodierungsschema, das Werte für
geographische Angaben vorgibt = **TGN**



Übung:

Literale und nicht-Literale Inhalte

Die Lösungen finden Sie auf der nächsten Seite

Handelt es sich beim Inhalt von W3CDTF um ein Literal oder ein Non-Literal?

Handelt es sich beim Inhalt von Period um ein Literal oder ein Non-Literal?

Handelt es sich beim Inhalt von TGN um ein Literal oder ein Non-Literal?



Lösung

Literale und nicht-Literale Inhalte

Handelt es sich beim Inhalt von W3CDTF um ein Literal oder ein Non-Literal? = **Literal**

Handelt es sich beim Inhalt von Period um ein Literal oder ein Non-Literal? = **Literal**

Handelt es sich beim Inhalt von TGN um ein Literal oder ein Non-Literal? = **Non-Literal**



Schritt 7

Wie wird mein Beschreibungssatzprofil maschinenlesbar?

Syntax

Hinweise auf die Kodierungssyntax und damit auf das Datenformat, in dem die Metadatensätze in der Anwendung gespeichert oder für den Metadatenaustausch zur Verfügung gestellt werden, sollten ebenfalls im Rahmen eines Metadatenprofils dokumentiert sein. Dabei muss beachtet werden, dass die Syntax die im Beschreibungssatzprofil genannten Regeln adäquat wiedergeben kann und die DCAM-Konformität der Metadaterme nicht verletzt. Verschiedene Richtlinien für die syntaktische Kodierung von Metadatenprofilen finden sich auf den Seiten der Dublin Core Metadata Initiative:

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/>



Übung

In 7 Schritten zum Profil

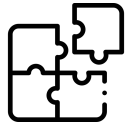
Die Lösungen finden Sie auf der nächsten Seite

Bitte bringen Sie die folgenden Fragestellungen in die korrekte Reihenfolge:

- 1 Wie wird mein Beschreibungssatzprofil maschinenlesbar?

- 2 Welche Metadaterme brauche ich?
 Was für Regeln gelten für Metadaterme?
 Was für Kodierungsschemas gelten für die einzelnen Terme?

- 3 Was will ich mit meinen Metadaten erreichen?
 Was will ich beschreiben?
 Welche Eigenschaften sind für meine Anwendung relevant?



Lösung

In 7 Schritten zum Profil

- 1 Was will ich mit meinen Metadaten erreichen?
Was will ich beschreiben?
Welche Eigenschaften sind für meine Anwendung relevant?
- 2 Welche Metadaterme brauche ich?
Was für Regeln gelten für Metadaterme?
Was für Kodierungsschemas gelten für die einzelnen Terme?
- 3 Wie wird mein Beschreibungssatzprofil maschinenlesbar?

LEGAL NOTICE

The online version of this publication is available via the following link: DOI: <https://doi.org/10.18452/22242.2>

Publisher: Deutsche Initiative für Netzwerkinformation e. V.

Contact details

DINI – Deutsche Initiative für Netzwerkinformation e. V.

Geschäftsstelle | c /o Niedersächsische Staats- und

Universitätsbibliothek Göttingen | Platz der Göttinger Sieben 1 |

37070 Göttingen |

Tel.: 0551 39-33857 | E-Mail: gs@dini.de | <https://www.dini.de>

Last updated: December 2020

DEUTSCHE INITIATIVE FÜR NETZWERKINFORMATION E. V.

All of the texts used in this publication, excluding citations, are licensed under a Creative Commons “Attribution 4.0 International” (CC BY 4.0) agreement. See <https://creativecommons.org/licenses/by/4.0/>.

