# 1. Einführung in DDB-DC_RDF/XML

Das DDB-DC_RDF/XML Anwendungsprofil dient hauptsächlich zur DDB-internen Vorverarbeitung und Anreicherung von Datenlieferungen, die auf dem Metadatenstandard Dublin Core beruhen. Datenpartner werden jedoch ausdrücklich ermutigt, ihre Datensets bereits im DDB-DC_RDF/XML Anwendungsprofil zuzuliefern.

Beim Dublin Core™ Metadata Element Set (DCMES) handelt es sich um ein einfaches Metadatenformat, das ein Kernset an 15 Datenelementen definiert. Diese können durch zusätzliche, in den DCMI Metadata Terms empfohlene Felder, die eine detailliertere Beschreibung unterstützen, ergänzt werden.

Die in Dublin Core definierten Elemente sind zum Teil nicht differenziert genug, um bestimmten Anforderungen der DDB an die Granularität der gelieferten Metadaten gerecht zu werden, damit die Daten im DDB-Portal optimal dargestellt und durchsuchbar gemacht werden können. Um exaktere Aussagen und mehr Expressivität im DDB-DC-Eingangsformat zu ermöglichen, wurden zusätzlich zu den Begriffen aus dem DCMES und den DCMI Metadata Terms einige geeignete Properties und Klassen aus kompatiblen Metadatenstandards ausgewählt und in das Anwendungsprofil übernommen. Hierzu gehört das Europeana Data Model (EDM), aus dem insbesondere Properties für die Auszeichnung der verschiedenen Arten von digitalen Objekten (Vorschaubild, Mediendatei, Objekt im Medienviewer, Objekt im Kontext) übernommen wurden. Das im Bibliothekswesen verbreitete Datenformat BIBFRAME wurde für die Differenzierung verschiedener Klassen von Identifikatoren herangezogen. Weitere Properties und Klassen stammen aus den Datenformaten Machine-Readable Cataloging (MARC), dem Data Catalog Vocabulary (DCAT) sowie dem Simple Knowledge Organisation System (SKOS).

## Modellierung des Anwendungsprofils

Die Modellierung des Anwendungsprofils basiert auf dem Resource Description Framework (RDF) und folgt dem DCMI Abstract Model sowie den aktuellen Empfehlungen für die Umsetzung von Dublin Core in RDF. RDF ist syntaxunabhängig. Für das DDB-DC Anwendungsprofil verwendet die DDB eine Kodierung in XML. Für die Serialisierung von RDF in XML stellt das W3C eine eigenen Spezifikation bereit, die RDF XML Syntax.

### *Resource Description Framework (RDF)*

RDF gilt als einer der Bausteine des Semantischen Webs und als idealer Kodierungsstandard für Linked Data. Im RDF-Modell wird eine Aussage aus Subjekt, Prädikat und Objekt gebildet. Die Menge dieser sogenannten RDF-Tripel bildet einen RDF-Graphen. Subjekt und Objekt werden auch als „Knoten “ bezeichnet und die durch das Prädikat definierte Relation zwischen beiden als „Kanten“. Das Subjekt und das Prädikat der RDF-Aussage sind immer Ressourcen. Während das Prädikat immer durch einen URI eindeutig referenziert wird, kann das Subjekt sowohl eine URI-Referenz als auch ein „blinder Knoten“ (blank node) sein. Das RDF-Objekt kann sowohl eine durch einen URI referenzierte Ressource als auch ein blinder Knoten oder aber ein Literal (Freitext, Zahlen etc.) sein. Blinde Knoten sind Ressourcen ohne expliziten URI, d.h anonyme Ressourcen. Sie dienen zur Gruppierung von RDF-Objekten und ermöglichen so die Zuordnung von mehreren Werten zu einem Prädikat.

Der RDF-Spezifikation folgend unterscheidet DDB-DC_RDF/XML zwischen *Classes* (Klassen) und *Properties*. Classes sind Elemente, die Entitäten als Subjekt oder Objekt einer RDF-Aussage einer bestimmten Klasse zuordnen. *Properties* sind Elemente, die verwendet werden um die verschiedenen Eigenschaften/Prädikate einer Entität zu beschreiben (Kanten/Relationen). Für die bessere Lesbarkeit der Daten unterscheiden sich Classes und Properties durch Groß- und Kleinschreibung. Während Classes immer mit Großbuchstaben beginnen, werden Properties immer klein geschrieben.

In RDF ist es üblich, dass die Klassenzugehörigkeit des Subjekt-Knotens durch das Prädikat <rdf:type> spezifiziert wird (typed nodes). RDF/XML erlaubt es jedoch, dieses Tripel verkürzt auszudrücken, indem der Name des rdf:Description-Knotenelements, das in RDF/XML zur Gruppierung von Aussagen über eine Ressource dient, durch das Namespaced-Element ersetzt wird, das dem IRI (Internationalized Resource Identifier) des Werts der Typbeziehung entspricht.

**Klassenzugehörigkeit über rdf:type ausgedrückt**

```
<rdf:Description rdf:about="providerItemID_12345">
        <dc:creator>
                <rdf:Description rdf:about="http://d-nb.info/gnd/12201894X">
                        <rdf:type rdf:resource="http://purl.org/dc/terms/Agent"/>
                <skos:prefLabel>Weyer, Gabriel</skos:prefLabel>
                 </rdf:Description>
        </dc:creator>
</rdf:Description>
```

**Verkürzte Darstellung der Klassenzugehörigkeit mit Namespaced-Element**

```
<rdf:Description rdf:about="providerItemID_12345">
        <dc:creator>
                   <dcterms:Agent rdf:about="http://d-nb.info/gnd/12201894X">
                <skos:prefLabel>Weyer, Gabriel</skos:prefLabel>
             </dcterms:Agent>
        </dc:creator>
</rdf:Description>
```

### *DCMI Abstract Model*

Das DCMI Abstract Model baut in seiner formalen Semantik auf Konzepten von RDF auf. Es erlaubt, dass nicht-literale Werte einer Property selbst zur beschriebenen Ressource einer separaten Beschreibung (description) innerhalb eines Beschreibungssets (description set) werden können. In einer RDF-Umsetzung ist das Ergebnis eine verschachtelte Ressourcenbeschreibung, in der der Werte-Knoten (= RDF-Objekt) zum RDF-Subjekt einer Reihe von Tripels wird, die den Wert selbst beschreiben.

Diese Option wurde im Anwendungsprofil DDB-DC genutzt, um z.B. für Personen, Themenschlagwörter, Objekttypen oder Ortsangaben die literale Bezeichnung den zugehörigen URI aus kontrollierten Vokabularen eindeutig zuzuordnen. Die Modellierung der jeweiligen Properties nutzt dabei die oben dargestellte Möglichkeit der verkürzten Darstellung in RDF/XML.

Das DCMI Abstract Model berücksichtigt auch den Umstand, dass es für Anwendungen erforderlich sein kann, dass Beschreibungssets eine Beschreibung des Beschreibungssets selbst (Stichwort „administrative Metadaten“) enthalten. Im DDB-DC Anwendungsprofil wird dies genutzt, um beispielsweise zwischen den Rechteangaben für das beschriebene DDB-Objekt und denen für die beschreibenden Metadaten zu differenzieren oder auch zwischen dem Urheber des beschriebenen Objekts und dem Urheber des Datensatzes (Datenlieferant) zu unterscheiden.

**Administrative Metadaten als eingebettetes Beschreibungsset**

```
<rdf:Description rdf:about="providerItemID_12345">
        <dcterms:isReferencedBy>
                <dcat:CatalogRecord>
                        <!-- Identifikator für den Datensatz -->
                         <dc:identifier>providerItemID_12345</dc:identifier>
                         <!-- Urheber des Datensatzes (=Identifikator für den Datenpartner aus der DDB-
Registrierung) -->
                        <dc:creator>99900556</dc:creator>
                        <!-- Rechteangabe für die beschreibenden Metadaten -->
                       <dcterms:license rdf:resource="http://creativecommons.org/publicdomain/zero/1.0/"/>
                  </dcat:CatalogRecord>
        </dcterms:isReferencedBy>
        <dc:title>Putto auf einem Fisch</dc:title>
        <!-- Rechteangabe für das beschriebene Objekt -->
        <dcterms:license rdf:resource="http://creativecommons.org/licenses/by-nc-sa/4.0/"/>
        <!-- Urheber des beschriebenen Objekts -->
        <dc:creator>
                    <dcterms:Agent rdf:about="http://d-nb.info/gnd/12201894X">
                <skos:prefLabel>Weyer, Gabriel</skos:prefLabel>
             </dcterms:Agent>
        </dc:creator>
</rdf:Description>
```