

Technischer Aufbau EDM

Bausteine des EDM

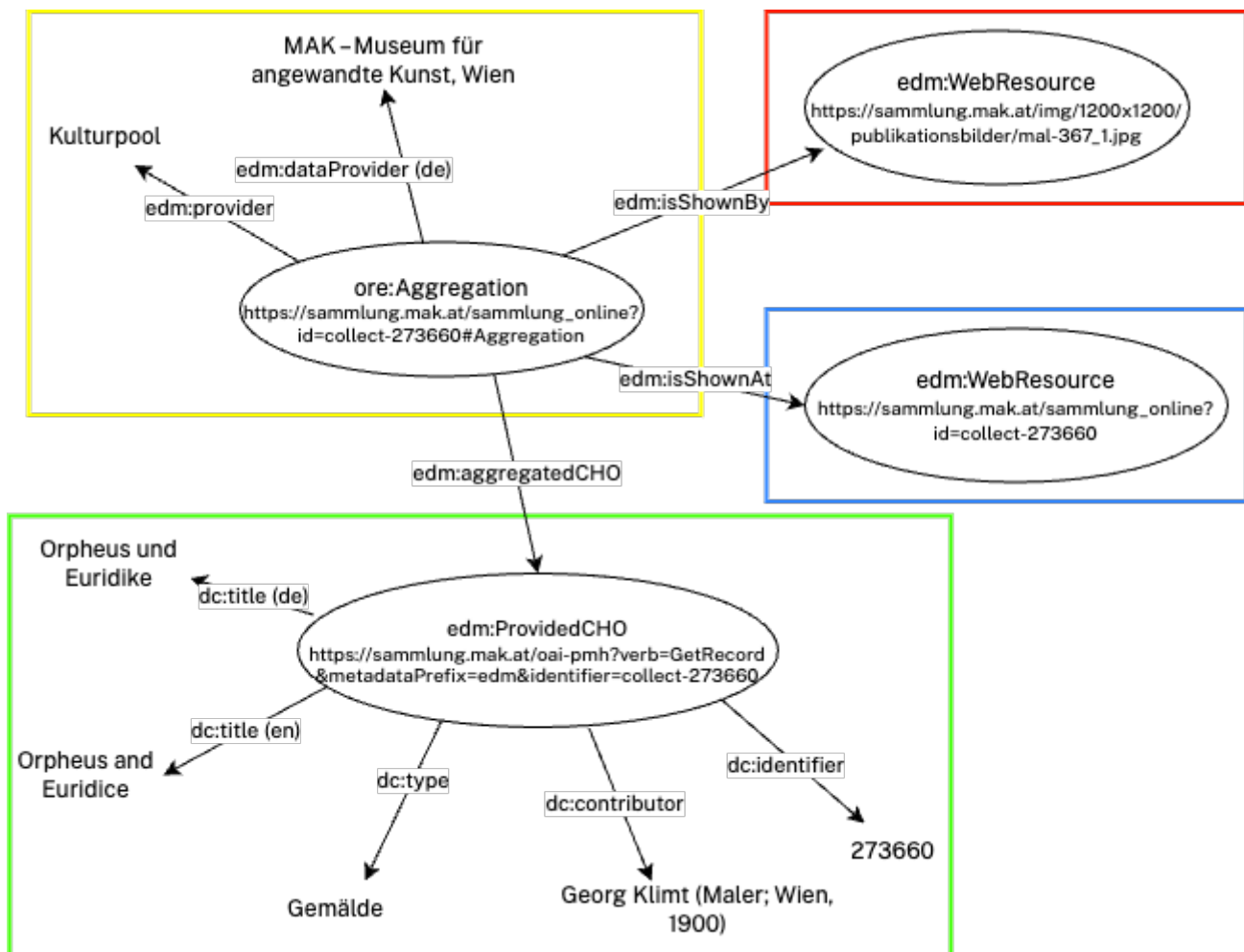
Ein [EDM](#)-Datensatz besteht grundsätzlich aus verschiedenen Elementen, die jeweils einen Teil der Informationen enthalten und in einem Verhältnis zueinander stehen.

Hauptklassen

Im EDM gibt es die drei Hauptklassen «ore:Aggregation», «edm:ProvidedCHO» und «edm:WebResource».

- **«ore:Aggregation»**: Dient als Haupteinstiegspunkt beim Lesen eines Datensatzes. Verweist weiter auf «edm:ProvidedCHO» und «edm:WebResource» und enthält ein paar Grundinformationen, wie die liefernde Institution. Muss genau einmal pro EDM-Datensatz vorkommen.
- **«edm:ProvidedCHO»**: Beinhaltet Informationen zum Kulturgut (**C**ultural **H**eritage **O**bject) selbst. Muss genau einmal pro EDM-Datensatz vorkommen.
- **«edm:WebResource»**: Beinhaltet Daten zu einem Digitalisat (z. B. Bild oder 3D-Modell des Kulturguts). Kann öfter vorkommen (z. B. Bild des Objekts von vorn, Bild des Objekts von der Seite).

Technisch basiert EDM auf dem [→ RDF-Standard](#), weshalb man die Daten leicht als Graphen darstellen kann:



Genaue Kenntnisse von RDF können ein tieferes Verständnis von EDM schaffen, die wichtigsten Grundlagen für EDM sind:

- Es gibt Objekte (`edm:ProvidedCHO`, `edm:WebResources` ...) mit **Properties** (z. B.: `dc:title`, `dc:type` ...). Diese Properties können entweder auf ein anderes Objekt verweisen oder auf einen **Literal** (z. B.: „Gemälde“), wodurch eine Graphenstruktur entsteht.
- Objekte werden über eine eindeutige ID (URI) identifiziert, die für den Verweis von anderen Objekten aus verwendet wird.

Grundsätzlich können RDF-Daten in vielen verschiedenen Formaten hinterlegt werden (RDF-XML, Turtle, JSON-LD ...). Da Europeana die Daten jedoch in der XML-Serialisierung erwartet und es für XML auch einige Tools gibt, werden wir uns hier nur auf diese Variante konzentrieren. Das obige Beispiel würde in RDF-XML folgendermaßen aussehen:

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/" xmlns:edm="http://www.europeana.eu/schemas/edm/"
xmlns:ore="http://www.openarchives.org/ore/terms/">

```

```

<ore:Aggregation rdf:about="https://sammlung.mak.at/sammlung_online?id=collect-273660#Aggregation">
  <edm:aggregatedCHO
    rdf:resource="https://sammlung.mak.at/oai-
pmh?verb=GetRecord&metadataPrefix=edm&identifier=collect-273660" />
    <edm:dataProvider xml:lang="de">MAK - Museum für angewandte Kunst, Wien</edm:dataProvider>
    <edm:isShownAt rdf:resource="https://sammlung.mak.at/sammlung_online?id=collect-273660" />
    <edm:isShownBy rdf:resource="https://sammlung.mak.at/img/1200x1200/publikationsbilder/mal-
367_1.jpg" />
    <edm:provider>Kulturpool</edm:provider>
    <edm:rights rdf:resource="http://creativecommons.org/licenses/by-sa/4.0/" />
  </ore:Aggregation>

  <edm:ProvidedCHO
    rdf:about="https://sammlung.mak.at/oai-
pmh?verb=GetRecord&metadataPrefix=edm&identifier=collect-273660">
    <dc:contributor>Georg Klimt (Maler; Wien, 1900)</dc:contributor>
    <dc:identifier>273660</dc:identifier>
    <dc:title xml:lang="de">Orpheus und Euridike</dc:title>
    <dc:title xml:lang="en">Orpheus and Euridice</dc:title>
    <dc:type>Gemälde</dc:type>
  </edm:ProvidedCHO>

  <edm:WebResource rdf:about="https://sammlung.mak.at/sammlung_online?id=collect-273660" />
  <edm:WebResource rdf:about="https://sammlung.mak.at/img/1200x1200/publikationsbilder/mal-367_1.jpg"
/>

</rdf:RDF>

```

- Das äußerste Element ist immer ein `rdf:RDF`-Tag, hier werden die vollen Namespaces für alle verwendeten *prefixes* angegeben (rdf, dc, edm ...)
- Die Objekte `ore:Aggregation`, `edm:ProvidedCHO` und `edm:WebResource` sind hierarchisch alle auf derselben Ebene definiert und bekommen über das Attribut `rdf:about` eine eindeutige URI zugewiesen. Diese muss nicht auflösen, sondern dient in erster Linie als ID, über die darauf verwiesen werden kann.
- Andere Properties wie `edm:aggregatedCHO` können über das Attribut `rdf:resource` auf ein Objekt mit der angegebenen URI verweisen.
- Für die `edm:WebResource`-Objekte ist die identifizierende URI auch gleichzeitig die URL, unter der die Ressource abrufbar ist.

- Wenn zu einem Property mehrere Werte vorhanden sind, können diese einfach ein weiteres Mal angegeben werden (mit wenigen Ausnahmen, siehe besonders `isShownBy` und `isShownAt`).
- Wenn die Sprache eines Felds bekannt ist, kann diese im Attribut `xml:lang` angegeben werden.

Sprach-Tags

Wie im vorhergehenden Beispiel kann, wo sinnvoll, die Sprache der Metadatenfelder per Attribut `xml:lang` angegeben werden. Durch Wiederholung ist es möglich, den selben Inhalt in mehreren Sprachen anzugeben:

```
<dc:type xml:lang="en">oil painting</dc:type>
<dc:type xml:lang="de">Ölgemälde</dc:type>
```

Generell wird empfohlen, alle Feldinhalten in so vielen Sprachen wie möglich anzugeben, da die Daten, besonders über die Weitergabe an Europeana, einem internationalen Publikum zugänglich gemacht werden sollen.

Der angegebene Sprachcode muss [→ BCP 47](#) konform sein. In Praxis reicht in den meisten Fällen die Auswahl des entsprechenden [→ ISO 639-1 oder 639-3 Codes](#).

Kontextuelle Klassen

Zusätzlich gibt es noch vier Kontextuelle Klassen, die genau gleich verwendet werden (aus technischer Sicht kein Unterschied zu den Hauptklassen).

Auf diese wird hauptsächlich vom `edm:ProvidedCHO` aus verlinkt und sie kapseln bestimmte Informationen zusammen. Die Rolle, die die Kontextuellen Klassen im Datensatz spielen, hängt von der Verlinkung ab. So kann eine Person (`edm:Agent`) beispielsweise `creator` sein, oder nur `contributor`.

Die verfügbaren Kontextuellen Klassen sind:

- **«edm:Agent»**: Eine natürliche oder juristische Person, oder Gruppe
- **«edm:Place»**: Ein Ort
- **«edm:TimeSpan»**: Eine Zeitspanne
- **«skos:Concept»**: Ein Konzept

In XML könnte es in der Verwendung zum Beispiel folgendermaßen aussehen:

```
<rdf:RDF>
  <edm:ProvidedCHO>
    <dc:contributor rdf:resource="AgentID_1"/>
  </edm:ProvidedCHO>

  <edm:Agent rdf:about="AgentID_1">
    <skos:prefLabel>Georg Klimt</skos:prefLabel>
    <skos:altLabel>Klimt, Georg</skos:altLabel>
    <rdaGr2:dateOfBirth>1867</rdaGr2:dateOfBirth>
    <rdaGr2:dateOfDeath>1931</rdaGr2:dateOfDeath>
  </edm:Agent>
</rdf:RDF>
```

Kontrolliertes Vokabular/Thesauri

[Kontrollierte Vokabulare und Thesauri](#) können referenziert werden, statt die Kontextklassen selbst zu definieren. Vorteile dabei sind, dass nur die URL angegeben werden muss, alle weiteren Informationen werden dann darüber abgerufen. Außerdem ermöglicht dies eine Standardisierung und Identifizierung von Entitäten über mehrere Datensätze und Institutionen hinweg.

Die Verlinkung erfolgt über das Attribut `rdf:resource`:

```
<dc:contributor rdf:resource="https://d-nb.info/gnd/136070213"/>
```

Der Tag darf in diesem Fall **keinen Inhalt** haben, da bereits auf komplexere Strukturen verlinkt wird.

[→ Liste der unterstützten Vokabulare](#)

FAQ

Was ist der Unterschied zwischen `rdf:about` und `rdf:resource`?

rdf:about muss bei der Definition von Instanzen («edm:ProvidedCHO», «oreAggregation», «edm:WebResource» ...) angegeben werden, über die dann mit den inneren Tags etwas ausgesagt wird.

rdf:resource wird hingegen verwendet, um auf eine Ressource zu verweisen, die an einem anderen Punkt definiert wird.

Wie vergebe ich URIs in rdf:about und rdf:resource?

rdf:about:

- Selbst gewählte **URLs** aus einer **eigenen Domäne**.
- Dienen in erster Linie als Identifikatoren und müssen großteils nicht auflösen.
Ausnahmen sind hierbei Web-Ressourcen, deren URLs immer auf eine Repräsentation des Objekts auflösen müssen (z. B. Link zum Objekt im Kontext der eigenen Online-Sammlung oder Direktlink zu einer Datei als Digitalisat, z. B. Bild-, Audio-, Video- oder Textdatei).
- **Global einzigartig**, auch über Klassen hinweg (nicht selber Identifier für Aggregation und ProvidedCHO).
- Sollte sich in Zukunft nicht mehr ändern.
- Beispiel: `https://museum.example/rdf/objects/cho_{identifizier}`, bzw.
`https://museum.example/rdf/objects/aggregation_{identifizier}`
- Durch die Angabe eines erfundenen Pfades, der nicht auflöst, kann die Option offen gehalten werden, die RDF-Daten in Zukunft dort abrufbar zu machen.

rdf:resource: Muss bei selbst definierten Ressourcen genau die URL sein, wie sie im zugehörigen rdf:about steht. Bei Referenz auf externe Ressourcen sollte eine URL aus einem [von Europeana unterstützten Vokabular](#) genutzt werden. Hierfür muss immer der Hauptlink der Ressource angegeben werden, der bei den meisten Vokabularen extra aufgelistet wird (z. B. als „Page Link“ oder „Link zum Datensatz“).

Wie kann ich mehrere Informationen in einem Feld angeben?

Bis auf wenige Ausnahmen können die Felder in EDM fast immer einfach wiederholt werden, um mehrere Werte anzugeben. Bei einem Objekt aus Porzellan mit Golddekor könnte das im

ProvidedCHO zum Beispiel so aussehen:

```
<dcterms:medium>Porzellan</dcterms:medium>
<dcterms:medium>Gold</dcterms:medium>
```

Die wichtigsten Ausnahmen sind:

- **«edm:type»**: Muss exakt einmal angegeben werden und darf nicht öfter vorkommen. Es kann sein, dass mehrere Werte passen, dann muss man sich auf einen davon festlegen. Zum Beispiel wäre bei Urkunden mit Bildern als Digitalisaten sowohl TEXT als auch IMAGE argumentierbar. In diesem Fall würden wir eher TEXT empfehlen, da der Textinhalt noch immer im Vordergrund steht.
- **«edm:rights»**: Hier darf pro Aggregation oder Web-Ressource jeweils nur eine Lizenz angegeben werden. Wenn es mehrere Digitalisate mit unterschiedlichen Lizenzen gibt, sind diese Lizenzen in der jeweiligen Web-Ressource anzugeben.
- **«edm:isShownBy»** und **«edm:isShownAt»**: Sollen jeweils nur einmal vorkommen für die Haupt-Web-Ressourcen. Wenn es weitere Digitalisate gibt, müssen diese in **«edm:hasView»** angegeben werden.
- **«edm:object»**
- **«edm:aggregatedCHO»**
- **«edm:dataProvider»** und **«edm:provider»**

Was ist der Unterschied zwischen xml:lang und «dc:language»?

- **xml:lang** ist ein Attribut, das für einzelne Felder angegeben werden kann, um die Sprache des angegebenen Werts zu kennzeichnen. Wenn möglich, sollte dies immer angegeben werden. Beispiel:

```
<dc:subject xml:lang="de">Porträt</dc:subject>
<dc:subject xml:lang="en">portrait</dc:subject>
```

- **«dc:language»** ist ein eigenes Feld, in dem die Sprache des eigentlichen Kulturerbe-Objekts angegeben werden kann, wenn es einen sprachlichen Aspekt gibt (z. B. bei Texten oder Audio). Beispiel:

<dc:language>de</dc:language>

Linktipp

- [W3C: RDF Spezifikation \(Englisch\)](#)

Version #51

Erstellt: 16 April 2024 11:24:48 von LG

Zuletzt aktualisiert: 17 September 2024 11:03:52 von LG