Metadatenmapping im Projekt Deutsche Digitale Bibliothek anhand von Dublin Core, Lido und EAD.

Magisterarbeit

Zur Erlangung des akademischen Grades Magister Artium im dem Fachbereich Historisch-Kulturwissenschaftliche Informationsverarbeitung der Universität zu Köln

Vorgelegt von Miyasat Alieva
Heumarkt 42 50667 Köln
Köln, den 12.09.2011
Gutachter: Prof. Dr. Manfred Thaller

Danksagung

Hiermit möchte ich mich bei dem ganzen DDB-Team am Fraunhofer Institut für Intelligente Analyse- und Informationssysteme für die tatkräftige Unterstützung bei meiner Abschlussarbeit herzlich bedanken.

Inhaltsverzeichnis

1 Vorbemerkungen zum Projektkontext: Metadatenmapping im Rahmen der Deutschen Digitalen Bibliothek

Das Projekt Deutsche Digitale Bibliothek (DDB) ist ein nationales Projekt, das 2006 ins Leben gerufen wurde, mit dem Ziel, das gesamte kulturelle Erbe der Bundesrepublik Deutschland im Web zugänglich zu machen. Gleichzeitig stellt die DDB den Beitrag zur Europäischen Digitalen Bibliothek (Europeana) dar.[1] Das Projekt wird von dem Beauftragten der Bundesregierung für Kultur und Medien (BKM)[2] in enger Zusammenarbeit mit Ländern und Gemeinden unterstützt. Die Bundeskanzlerin Dr. Angela Merkel definierte das Ziel der DBB mit folgenden Worten:

> "Ein gemeinsames Aktionsfeld von Bund und Ländern ist die Deutsche Digitale Bibliothek. Das heißt, wir wollen unseren nationalen kulturellen und wissenschaftlichen Reichtum international präsentieren. Das ist ein sehr spannendes Projekt. Wir wollen versuchen, dass jeder Bürger von seinem internetfähigen PC wirklich Zugang zu diesem Angebot bekommen kann."[3]

Der Grund der Errichtung einer solch globalen öffentlichen Plattform, liegt vor allem in dem Wunsch, den Benutzern einen effizienten Zugang zu den dezentral vorliegenden Digitalisaten zu ermöglichen.[4] Im Vergleich zu den gängigen Suchmaschinen, wie z.B. Google, offeriert die DDB dem Benutzer die Möglichkeit, seine Suche durch die Auswahl von Facetten[5] exakter zu definieren, wodurch auch die Vielfalt von Online-Angeboten unterschiedlicher Kultur- und Wissenschaftseinrichtungen eingegrenzt wird. Dank der einheitlichen und detaillierten DDB-Ressourceneinzelansicht wird dem Nutzer die sonst anstehende Auseinandersetzung mit den unterschiedlich organisierten Online-Datenbanken

---

[1] Zu dem Konzept und der Entstehung von Europeana siehe Kommission der Europäischen Gemeinschaften: Europas kulturelles Erbe per Mausklick erfahrbar machen, Stand der Digitalisierung und Online-Verfügbarkeit kulturellen Materials und seiner digitalen Bewahrung in der EU. Brüssel, v. 11.08.2008, S. 2 ff., <http://ec.europa.eu/information_society/activities/digital_libraries/doc/communications/progress/communication_de.pdf> (10. 06 2011).

[2] Im Auftrag des BKM wurde auch die Studie "Auf dem Weg zur DDB" von der Fraunhofer Gesellschaft herausgegeben. Diese beschreibt praktische Schritte und Investitionen, die für den Aufbau und Betrieb der DDB einzuplanen sind. Fraunhofer Institut für Intelligente Analyse- und Informationssysteme: Auf dem Weg zur DDB, v. 04.03.2008, <http://www.deutsche-digitale-bibliothek.de/pdf/auf_dem_weg_studie.pdf> (10.06.2011).

[3] Rede zur Eröffnung der Cebit 2007, zit. nach Bundesregierung Online: Deutsche Digitale Bibliothek, v. 04.12.2009, <http://www.bundesregierung.de/nn_774/Content/DE/StatischeSeiten/Breg/BKM/2008-02-26-deutsche-digitale-bibliothek.html> (19.06.2011).

[4] Zur Errichtung der DDB siehe Bund-Länder-Fachgruppe DDB: Fachkonzept zum Aufbau und Betrieb einer Deutschen Digitalen Bibliothek, v. 16.02.2008, <http://www.deutsche-digitale-bibliothek.de/dokumente.htm> (19.06.2011). Weitere Informationen sind auf der Homepage der Deutschen Digitalen Bibliothek zu erhalten: <http://www.deutsche-digitale-bibliothek.de> (19.06.2011).

[5] Filter, die für die Suche auf dem Portal angeboten werden. Weiter zu den Facetten im Kapitel 4.2.3.

gespart und die Ergebnisse der Suche einheitlich präsentiert. Der Ansatz, dem Nutzer nicht nur eine möglichst schnelle und individuell erweiterte Suche anzubieten, sondern ihm darüber hinaus die Ergebnisse möglichst klar und greifbar darzulegen, spricht für den stark benutzerorientierten Ansatz des Systems. Ein weiteres wichtiges Qualitätsmerkmal der DDB besteht in der kompetenten Auswahl der zur Verfügung gestellten Inhalte, die aus geprüften Quellen stammen und von Fachpersonal erschlossen werden.

Bei der Umsetzung des der DDB zugrundeliegenden Konzeptes werden zahlreiche existierende Vorreiter aus dem Bereich der Digitalisierung berücksichtigt und reflektiert. Dies gilt insbesondere für Erfahrungswerte aus dem deutschen digitalen Raum, innerhalb dessen eine Reihe von Kultur- und Wissenschaftseinrichtungen ihre Inhalte bereits digitalisiert und online zugänglich gemacht haben. Es wurden weiterhin spartenübergreifende Plattformen entwickelt, die darauf ausgerichtet sind, dem Nutzer Informationen aus mehreren unterschiedlichen Institutionen zur Verfügung zu stellen. Hier sei als Beispiel das BAM-Portal (Gemeinsames Internetportal für Bibliotheken, Archive und Museen) zu nennen, das von der Deutschen Forschungsgemeinschaft (DFG) im Jahre 2006 gefördert wurde.[6] Das Projekt erfasst bisher jedoch nur einen kleinen Teil der in Deutschland vorhandenen Kulturgüter und operiert nur mit bestimmten Metadatensätzen wie MAB2 für Bibliotheken, EAD für Archive und *museumdat* für Museen.[7]

Das Portal Europeana, das im September 2005 von der Europäischen Kommission ins Leben gerufen wurde, hat es sich ebenfalls zum Ziel gesetzt, die kulturellen und wissenschaftlichen Inhalte Europas der breiten Öffentlichkeit online zugänglich zu machen. Das Austauschformat Europeana Semantics Elements (ESE), mit dem Europeana ursprünglich operieren sollte, hat sich jedoch als ausdrucksschwach und für den Ansatz von *Linked Open Data* als ungeeignet herausgestellt. Die geplante Umstellung auf das Europeana Data Model (EDM), mit dem bestimmte Nachteile von ESE ausgeglichen werden sollen, befindet sich derzeit in Bearbeitung.[8] Das Konzept des Projekts DDB orientiert sich an dieser Vorarbeit, erkannte jedoch den Bedarf an neuen Ansätzen, welche

---

6 Weitere Informationen zu dem Portal unter: <http://www.bam-portal.de> (20.07.2011).

7 Schweibenz, W., & Sieglerschmidt, J. K.: Aktuelle Entwicklungen bei Kultur-Portalen: BAM-Portal, Deutsche Digitale Bibliothek und Europeana, 2010, S. 8. Zu vergleichbaren Einrichtungen siehe Bund-Länder-Fachgruppe DDB: Fachkonzept zum Aufbau und Betrieb einer Deutschen Digitalen Bibliothek, S. 6-7.

8 Siehe Doerr, M., Gradmann, S., Hennicke, S., Isaac, A., Meghini, C. & Van de Sompel, H.: The Europeana Data Model (EDM), IFLA 2010, S. 2, <http://www.ifla.org/files/hq/papers/ifla76/149-doerr-en.pdf> (27.07.2011).

die Schwierigkeiten und Schwachpunkte der Vorgänger aufgreifen und darüber hinaus den eigenen projektspezifischen Zielen gerecht werden.

Die DDB wurde als eine öffentliche Online-Plattform konzipiert, die Ressourcen aus Deutschlands Kultur- und Wissenschaftseinrichtungen im Sinne des *Semantic Web* verwaltet und unter der Berücksichtigung von lizenzbedingten Beschränkungen online zur Verfügung stellt.[9] Zu den verwalteten Ressourcen gehören u.a. Bücher, Handschriften, Bilder, Fotos, Filme, 3-D-Objekte, Digitalisate, Audio- und Videomaterialien. Die Realisierung des Portals erfordert enge Zusammenarbeit mit zahlreichen Kultureinrichtungen: Bibliotheken, Museen, Archiven, Denkmalschutzämtern, wissenschaftlichen Einrichtungen und AV-Archiven. Den ersten Einschätzungen zufolge rechnet man mit Metadaten im Umfang von ungefähr 300 Millionen Objektreferenzen. Als Zielgruppe werden für die DDB vor allem die Wissenschaftler anvisiert, daneben Lehrkräfte, Schüler und Studenten, aber auch die breite Öffentlichkeit.[10]

Die Projektleitung für die Ausbaustufe I liegt beim Fraunhofer Institut für Intelligente Analyse- und Informationssysteme in Sankt Augustin. Hier wird das System im Rahmen des DDB-Projekts voraussichtlich bis Ende 2011 entwickelt. Der Content, der von verschiedenen Einrichtungen für das DDB-Projekt zur Verfügung gestellt wird, ist eine Sammlung verschiedener Metadaten, die sich für die Beschreibung von Objekten in der Wissensverwaltung etabliert haben.[11] Aus dem Bereich der Museen u.a. stammen museumdat und Lightweight Information Describing Objects (LIDO), aus dem der Bibliotheken Dublin Core, und aus dem Bereich der Archive vor allem Encoded Archival Description (EAD). Das DDB-Projekt sieht vor, diese unterschiedlichen Formate in ein einheitliches Datenmodell zu transformieren, das von den einzelnen Datenformaten abstrahierend eine Abbildung der Semantik der Originaldaten in möglichst verlustfreier Form erlangt. Erschwerend kommt hinzu, dass die Formate jeweils in unterschiedlichen Versionen und auf verschiedene Weise verwendet werden (mehr dazu im Abschnitt 2.7.2).

---

[9] Das Konzept des *Semantic Web* wird in seinen wichtigsten Bestandteilen im Kapitel 2.5 behandelt.

[10] Zu den unterschiedlichen Szenarien der Nutzung der DDB von verschiedenen Zielgruppen und die zu erfüllenden Voraussetzungen dafür siehe Fraunhofer Institut für Intelligente Analyse- und Informationssysteme: Auf dem Weg zur DDB, 14-15.

[11] Binäre Inhalte (Streams), Digitalisate (digitalisierte Objekte) oder Derivate (formatierte Inhalte) können ebenfalls mitgeliefert werden.

Die Rolle des allgemeinen Datenmodells übernimmt das CIDOC Conceptual Reference Model (CRM).[12] Eine der wichtigsten Eigenschaften des CRM bietet seine problemlose Umwandlung in das Resource Description Framework (RDF) und das damit verbundene *Semantic Web* Potential sowie seine *Linked Data* Affinität. Die Datenlieferanten brauchen keine Anpassung der Datenformate vorzunehmen. Das Metadatenmapping wird innerhalb der DDB-Plattform von einer im DDB-Projekt entwickelten Software namens *Augmented SIP Creator* (ASC) durchgeführt.[13] Diese prozessiert die Mapping-Skripte, mit denen die Quellmetadaten in das DDB-Datenmodell übertragen werden.

Im Rahmen der vorliegenden Magisterarbeit wird das Konzept eines Metadatenmappings vorgestellt, das eine Transformation von deskriptiven Metadaten in das CRM erlaubt. Insbesondere wird auf das Mapping von EAD, LIDO und DC in das CRM eingegangen. Das Mapping orientiert sich an den generischen Standardmappings, die von den Entwicklern des CRM veröffentlicht wurden. Die existierenden Mapping-Empfehlungen werden berücksichtigt und um weitere Ansätze ergänzt.

Die Magisterarbeit ist wie folgt aufgebaut: Im einleitenden Teil (Kapitel II) werden die wichtigsten Konzepte der Kulturdatenmodellierung erläutert: Metadaten und Ontologien in ihren unterschiedlichen Ansätzen. Ferner werden weitere für das Mapping relevante Begriffe, wie Vokabulare und das Resource Description Framework behandelt, insbesondere in Hinblick auf ihre Bedeutung für das *Semantic Web*. Weiterhin werden die verschiedenen Probleme des Mappings skizziert und an einigen Beispielen diskutiert. Im Kapitel III wird auf das CRM im Detail eingegangen: auf seinen Aufbau und Anwendungsbereiche sowie in Bezug auf die Argumente, die das CRM zum Mapping-Zielformat ernannt haben. Mit dieser Grundlage wird anschließend im Kapitel IV ein Mapping-Ansatz eingeführt, der auf der Bereitstellung von konzeptuellen Mappings (Clustern) für die gängigsten Typen von Objekten und Events im Bereich des Kulturerbes basiert. Das Ergebnis dieses Mappings bildet einen wichtigen Bestandteil des DDB-Datenmodells, das im Kapitel IV zusammen mit weiteren wichtigen Konzepten des Mappings – wie Ressourcen, Identifier und Events – anhand von Beispielen unterlegt wird.

---

[12] Ausführliche Informationen sind auf der offiziellen Seite von CIDOC CRM verfügbar: The CIDOC Conceptual Reference Model, <http://www.cidoc-crm.org/> (20.07.2011).

[13] Siehe Stalmann K. & Budde R.: Projekt Deutsche Digitale Bibliothek (DDB): Grobkonzept für das Portal der Deutschen Digitalen Bibliothek, Version 2.0, v. 12.08.2010, S. 7, <http://www.iais.fraunhofer.de/uploads/media/DDBGrobkonzeptFinal.pdf > (20.07.2011).

2 Einleitung zu allgemeinen Aspekten der Datenmodellierung, Metadaten, Ontologien, Semantic Web und Mapping

2.1 Datenmodellierung

2.1.1 Grundkonzepte der Datenmodellierung

Unter Datenmodellierung versteht man ein Verfahren zur formalen Abbildung der Informationsobjekte mit Hilfe ihrer Attribute und Beziehungen. Die Datenmodellierung kommt zum Einsatz, wenn ein klarer Überblick über die Datensicht einer Wissensdomain benötigt wird. Bei der Erstellung eines Datenmodells können auch Lücken und Inkonsistenzen der entwickelten Konzepte entdeckt werden, die bis dato als vollständig und ausgereift galten. Datenmodelle an sich dienen grundsätzlich als Kommunikationshilfe zwischen Fachleuten und Programmierern.[14] In der Regel werden für den Entwurf einer Datensicht Objekttypen benutzt und keine einzelne Exemplare der Informationsobjekte. Dies lässt die Möglichkeit zu, generelle Eigenschaften und Zusammenhänge aufzuweisen und identifizierbar zu machen. Um ein Exemplar jedes Objekttyps eindeutig identifizieren zu können, werden Primärschlüssel definiert. Mit Hilfe von Primärschlüsseln werden die Beziehungen zwischen den Exemplaren der verschiedenen Objekttypen hergestellt (siehe Abbildung 2). Ein Primärschlüssel kann entweder ein künstlich generierter Wert sein oder eine Kombination aus vorhandenen Attributen eines Objekttyps.[15] Relevant für einen Primärschlüssel ist es, dass er in einem bestimmten Kontext nur ein einziges Mal vorkommt. Nach diesem Prinzip werden auch die lokalen Identifier für DDB-Ressourcen beim Mapping erzeugt (siehe Kapitel 4.2.2).

Für die Erstellung eines Datenmodels werden in der Regel drei Typen von Elementen verwendet: Objekttypen (Entitäten), ihre Attribute und die Beziehungen (Relationen/Eigenschaften) zwischen diesen Objekttypen. Das Ergebnis des Modellierungsprozesses bildet ein sog. Datenschema, das u.a. graphisch visualisiert

[14] Für eine ausführliche Beschreibung der Datenmodellierung sei hingewiesen auf: Wiborny, W.: *Datenmodellierung, CASE, Datenmanagement.* Bonn, München: Reading, Mass. [u.a.]: Addison-Wesley, 1991, S. 3 ff.

[15] Mehr dazu Lackes, R., & Siepermann, M.: Datenmodellierung, v. 01.10.2010, http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Datenmanagement/Daten-/Datenmodellierung-/> (20.07.2011).

werden kann. Abbildung 1 zeigt ein Entity-Relationship-Modell (ERM) der Beziehung "Ausleihe".[16]

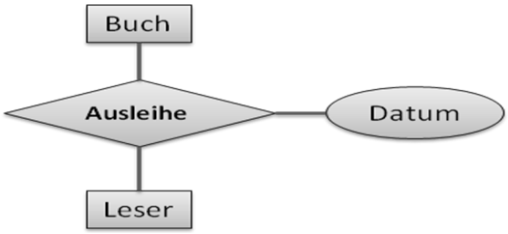


Abbildung 1: Entwurf der Beziehung "Ausleihe"
Entität, Beziehung, Attribut.

Ferner kann das Datenschema für die Erstellung einer einsatzfähigen Datenbank benutzt werden, in der die konkreten Datensätze gespeichert werden können (Abbildung 2).

Buch

InvNr	Autor	Titel	Verlag	Jahr
027-2408	Jones	Algorithms	PH	2003
176-2709	Peters	Java Programming	MKP	2006
182-2709	Peters	Java Programming	MKP	2006
188-2887	Jameson	Web Design	JO	2006

Ausleihe

InvNr	LeserNr	Datum
027-2408	428456	30-09-2006
176-2709	389476	20-06-2006

Leser

LeserNr	Name	Telefon
389476	Johnson	02-6674563
428456	Andrews	07-8446524

Abbildung 2: Ausschnitt aus einer relationalen Datenbank.[17]

[16] Siehe Vossen, G.: *Datenmodelle, Datenbanksprachen und Datenbankmanagement-systeme.* München, Wien: Oldenbourg Wissenschaftsverlag GmbH, 2008, S. 75. Zum ERM siehe ebd., S 60-80.
[17] Siehe ebd., S. 96.

Die Abfragesprachen erlauben komplexe Abfragen (Queries) auf dieses Schema. Man kann mit den Queries gezielt einzelne Datenbankfelder abfragen und bekommt als Ergebnis die Datensätze ausgegeben, die sie beinhalten.

2.1.2 Kulturdatenmodellierung

Datenbanksysteme werden bereits seit Jahrzehnten im Kulturbereich eingesetzt, vor allem im Bereich des Bibliothekswesens. Als Beispiel hierfür sind sog. *Online Public Access Catalogues* (OPACs) zu nennen. Hierunter versteht man Online-Bibliothekskataloge, die den Publikations- und Bücherbestand einer Bibliothek verzeichnen und im Internet zur Recherche bereitstellen.[18] Auch andere Institutionen nutzen Datenbanken, um zu den Objekten dazugehörige Daten zu speichern, darunter auch die Digitalisate. Die Organisation der Ressourcen über ein Datenbanksystem bietet u.a. den Vorteil, die Informationen schnell und ohne Umwege auffinden zu können, sowie den mehreren Benutzern gleichzeitig den Zugriff auf die gleichen Daten zu gewährleisten. Für die Archive kommt außerdem als Vorteil hinzu, dass die zugrundeliegenden Originale in Archiven aufbewahrt werden, und keinen weiteren, unnötigen mechanischen Belastungen ausgesetzt werden.[19]

Für Zwecke der Interoperabilität, der Möglichkeit, Daten von einem System zu einem anderen zu übertragen, wurden spezielle Austauschformate entwickelt.[20] Die Bibliotheken nutzen u.a. Machine-Readable Cataloging (MARC) als Format, um den Datentransfer zu vollziehen. Beim MARC handelt sich um ein Katalogisierungsformat, das 1955/1956 von der *Library of Congress* entwickelt wurde, um maschinenlesbare Katalogdaten zu erstellen. Seit 1968 wird MARC als universelles Austauschformat für bibliographische Informationen eingesetzt. MARC ist ein sehr detailliertes und bewährtes Austauschformat und bietet heute die Grundlage für viele gängige Online- Bibliothekskataloge. Derzeit gelten besonders seine Versionen UNIMARC und MARC 21 als populär.[21]

---

[18] Siehe Rowley, J. E. & Hartley, R. J.: Organizing knowledge: an introduction to managing access to information. Aldershot, Hants, England; Burlington, VT: Ashgate, 2007, S. 275.

[19] Siehe Archive und Datenbanken unter: <http://survival-mediawiki.de/dewiki/index.php/Archive_und_Datenbanken> (20.07.2011).

[20] Der Begriff "Interoperabilität" wird im Kapitel III im Zusammenhang mit CRM wieder aufgenommen und detaillierter erklärt.

[21] Zu dem MARC Format im Einzelnen siehe Byrne, D. J.: *MARC manual: understanding and using MARC records.* Englewood, Colo. : Libraries Unlimited, 1998.

Weitere Austauschformate, die im Bibliothekswesen vor allem in Deutschland genutzt werden, sind Maschinelles Austauschformat für Bibliotheken (MAB)[22] und seine Version MAB2,[23] Dublin Core, Metadata Object Description Schema (MODS)[24], etc. Für die Beschreibung von musealen Objekten werden museumdat und LIDO eingesetzt. Für Online-Findbücher wird EAD benutzt. Im Folgenden wird auf das Metadatenkonzept im Allgemeinen und auf die einzelnen Metadaten sowie eventuelle für das Mapping relevante Probleme eingegangen.

2.2 Metadaten und Retrieval

2.2.1 Was sind Metadaten?

Unter Metadaten versteht man "Daten, die strukturierte Informationen über andere Daten enthalten und die diese damit in Informationssystemen besser auffindbar machen".[25] Obschon der Begriff "Metadaten" relativ neu ist, nimmt das zugrunde liegende Prinzip seinen Ursprung in der bibliothekarischen Praxis der Katalogkarten. Diese wurden (und werden zum Teil immer noch) von Bibliothekaren nach bestimmten Regeln erfasst und in ein zugrundeliegendes System eingeordnet. Dabei beinhaltet eine Katalogkarte Informationen zu einem Bestandstück, in der Regel einem Buch, bspw. die Signatur, den Namen des Autors, das Erscheinungsjahr, die Auflage und den Verlag.[26] Diese Praxis hat sich im Laufe der Zeit stark ausgearbeitet und wurde vor allem in Hinblick auf Anforderungen des digitalen Zeitalters optimiert, sodass Metadaten im heutigen Sinne in einzelnen Dokumenten eingebettet werden.[27] Voraussetzung hierfür ist die Realisierung eines gewissen Standardisierungsgrades, nämlich die Strukturierung der Metadaten gemäß einem bestimmten Datenmodell, welches bestimmen soll, wie die einzelnen Datenfelder zu

---

[22] Weiterführende Informationen über MAB und MAB2 sind auf der Seite der Deutschen Nationalbibliothek zu finden: <http://www.d-nb.de/standardisierung/formate/mab.htm> (20.07.2011).

[23] Die Verwendung vom MAB Format im deutschsprachigen Bibliotheksbereich wird durch einen Umstieg auf MARC abgelöst. Zu diesem Beschluss siehe das Protokoll der Deutschen Bibliothek, Arbeitsstelle für Standardisierung: 9. Sitzung des Standardisierungsausschusses am 15. Dezember 2004, <http://www.d-nb.de/standardisierung/pdf/p_sta_20041215_vonpdf> (20.07.2011).

[24] Zu dem MODS Format siehe <http://www.loc.gov/standards/mods> (20.07.2011).

[25] Thaller, M. et. al.: Retrospektive Digitalisierung von Bibliotheksbeständen – Evaluierungsbericht über einen Förderschwerpunkt der DFG. Köln, 01.2005, S. 27. <http://www.dfg.de/forschungsfoerderung/wissenschaftliche_infrastruktur/lis/download/retro_digitalisierung_eval_050406.pdf> (10. 06 2011).

[26] Siehe Stock W. G. & Stock, M.: *Wissensrepräsentation: Informationen auswerten und bereitstellen.* München : Oldenbourg, 2008, S. 105.

[27] Eine eindeutige Namenstrennung zwischen den Metadaten im Sinne von einzelnen Informationen und dem Dokument, in dem diese Informationen abgespeichert werden, wird nicht gemacht. Beide Konzepte werden in der Fachliteratur als "Metadaten" bezeichnet. Letzteres auch als "Austauschformat", "Metadatenformat", "Standard".

nutzen sind.[28]

Als häufigste Form für die Darstellung dieser Datenmodelle wird *Extensible Markup Language* (XML)[29] gewählt. Im Sinne der Katalogkarten-Tradition enthält der Begriff der Metadaten weiterhin Informationen über den Titel, den Autor, das Erstellungsdatum der Ressource, darüber hinaus Informationen zu den Zugangs- und Nutzungskriterien, Informationen über die Struktur, den Umfang und die Beziehungen zu anderen Ressourcen, Informationen über die Herkunft und die Geschichte der Ressource sowie die urheberrechtliche und die Zielgruppeninformation. Metadaten sind so konzipiert, dass sie sowohl von Suchmaschinen als auch von Menschen gelesen und interpretiert werden können. In Rahmen dieser Arbeit werden folgende Metadatenbeispiele aufgegriffen: DC, EAD und LIDO. Im folgenden Abschnitt werden diese Metadaten aus der Sicht auf das bevorstehende Mapping erläutert.

2.2.2 Dublin Core (DC)

Dublin Core stellt seit etwa 1995 einen Standard dar, der von Wissenschaftlern und Bibliothekaren gemeinsam für die Beschreibung von Ressourcen aller Art erarbeitet wurde. Sein größter Vorteil liegt in seiner Einfachheit und Reduktion. Die Benennung von Dublin Core Elementen ist leicht verständlich definiert und ihre Anzahl ist auf 15 Elemente beschränkt.[30] Alle Dublin Core Elemente sind optional und können sich beliebig oft wiederholen. Darüber hinaus ist dieser Standard international einsetzbar und um weitere Spezifikationen erweiterbar.

Das Prinzip der Erweiterbarkeit des DC besteht darin, dass man mit zusätzlichen Qualifikatoren Konzepte stark kontextabhängig definieren kann und somit den Informationsstand grundsätzlich vertiefbar macht.[31] Es werden zwei Typen von Qualifikatoren differenziert: Element-Spezialisierungen (*Element Refinements*) und Kodierungsschemata (*Encoding Schemes*). Unter *Element Refinement* versteht man: "a property of a resource which shares the meaning of a particular DCMI Element but with

[28] Siehe Thaller, M. et. al.: Retrospektive Digitalisierung von Bibliotheksbeständen, S. 27.

[29] The *Extensible Markup Language* ist eine textbasierte Auszeichnungssprache zur Repräsentation hierarchisch strukturierter Daten und wird zum Datenaustausch im Web eingesetzt. Weiterführende Informationen sind auf der W3C Spezifikation von XML zu finden: <http://www.w3.org/TR/2008/REC-xml-20081126/> (20.07.2011).

[30] Siehe Dublin Core Metadata Element Set Version 1.1 unter: <http://dublincore.org/documents/dces/> (20.07.2011).

[31] Informationen zu den Dublin Core Qualifikatoren und ihrer Nutzung bietet: Dublin Core Qualifiers, <http://dublincore.org/documents/usageguide/qualifiers.shtml> (20.07.2011).

narrower semantics."[32] Das Dublin Core Element <dc:date> kann bspw. durch die folgenden Qualifikatoren spezifiziert werden: created, valid, available, issued, modified, Date Accepted, Date Copyrighted, Date Submitted.[33]

Dem gleichen Zweck der Spezifikation dienen auch die Kodierungsschemata, welche die Interpretation der DC Elemente regeln:

> "An Encoding Scheme provides contextual information or parsing rules that aid in the interpretation of a term value. Such contextual information may take the form of controlled vocabularies, formal notations, or parsing rules."[34]

Man unterscheidet zwischen zwei Typen von Kodierungsschemata.[35] Die *Vocabulary Encoding Schemes* weisen darauf hin, dass der beinhaltete Feldeintrag einem Begriff aus einem kontrollierten Vokabular entspricht. Bei der Eingabe des Kodierungsschemas *MeSH* bei dem Element <dc:subject> wird bspw. erwartet, dass der Wert des Feldes wie im folgenden Beispiel mit einem Begriff aus dem kontrollierten Vokabular "Medicine Medical Subject Headings"[36] identisch ist:

```
<dc:subject>
   <dcterms:MESH> Formate Dehydrogenase  </dcterms:MESH>
</dc:subject> [37]
```

Die *Syntax Encoding Schemes* betreffen die Syntax, welcher der Feldeintrag entsprechen soll. Bei der Datumseingabe entspricht der internationale Standard bspw. der Zeichenkette "1999-09-25T14:20+10:00"; dies kann mit der Kodierungsschema W3CDTF bestimmt werden:

```
<dc:date>
  <dcterms:W3CDTF>
     <rdf:value>1999-09-25T14:20+10:00</rdf:value>
  </dcterms:W3CDTF>
</dc:date>
```

---

[32] DCMI Grammatical Principles unter: <http://dublincore.org/usage/documents/principles/> (20.07.2011).

[33] Bedeutungen der einzelnen Datumsqualifikatoren unter: <http://dublincore.org/documents/usageguide/qualifiers.shtml> (20.07.2011).

[34] DCMI Grammatical Principles unter: <http://dublincore.org/usage/documents/principles/> (20.07.2011).

[35] Einen Überblick über die gängigsten Kodierungsschemata bietet: DCMI Encoding Schemes – a current list: <http://dublincore.org/documents/2002/10/06/current-schemes/> (20.07.2011).

[36] Begriffsvokabular von der U.S. National Library of Medicine: <http://www.nlm.nih.gov/mesh/> (20.07.2011).

[37] Diese und weitere Beispiele sind zu finden unter: Expressing Qualified Dublin Core in RDF / XML: <http://dublincore.org/documents/dcq-rdf-xml/> (20.07.2011).

Solche detaillierten Beschreibungen einer Ressource dienen dazu, die Ressource schneller und einfacher wiederzufinden und inhaltlich richtig zu interpretieren. Bei den an das DDB-Projekt gelieferten Dublin Core Metadaten wird dennoch häufig auf eine Datentypisierung verzichtet. In Folge dessen weisen die erschlossenen Informationen eine unscharfe Semantik auf. Dieses Problem wird in Abschnitt 2.7.2 wieder aufgegriffen.

2.2.3 Lightweight Information Describing Objects (LIDO)

LIDO ist ein XML-basiertes Austauschformat zur Beschreibung unterschiedlicher Sammlungsobjekte aus dem musealen Bereich. Das Format wurde entwickelt, um Museen die Möglichkeit zu geben, Informationen über ihre Sammlungsobjekte in Internet-Portalen problemlos zur Verfügung stellen zu können. Beim LIDO handelt es sich um ein sehr detailliertes Format, mit dem eine ganze Reihe an Metadaten für die Beschreibung eines musealen Objektes zusammengetragen werden kann.[38] Die datenliefernden Institutionen können grundsätzlich selber entscheiden, welche Metadaten sie an die Portale weitergeben und welche Informationen sie den Portalen vorenthalten möchten.
LIDO-Elemente werden in zwei Hauptgruppen unterteilt: die erste Gruppe trägt einen beschreibenden Charakter (*descriptive metadata*), die zweite trägt einen administrativen Charakter (*administrative metadata*). Zu den beschreibenden Informationen gehören u.a. die Angaben zu dem Typ, dem Titel, dem Umfang der Ressource, Angaben zu der Repository (dem Institut, wo die Ressource sich befindet), außerdem Beschreibungen der Ressource in Textform, Angaben zum Anzeigebild der Ressource im Portal sowie Quellangaben und Beziehungen zu den anderen Ressourcen.

Einen sehr wichtigen Teil im LIDO bildet die Beschreibung der sog. Events, dessen Konzept aus dem CIDOC CRM ins LIDO übernommen wurde. Bspw. werden solche Zeitereignisse wie die Erwerbung (Acquisition), die Herstellung (Production), die Geistige Schöpfung (Creation), die Bearbeitung (Modification), das Sammelereignis (Collection) und viele andere Konzepte als Ereignisse konzipiert.[39] Für die Beschreibung der Events werden folgende Angaben benutzt: Event ID, Event Type, Event Name, Event Date, Event

[38] Die komplette Beschreibung des Standards, seine geschichtliche und Strukturinformationen bietet: Coburn, E., Light, R., McKenna, G., Stein, R., & Vitzthum, A.: LIDO – Describing Objects Version 1.0, November 2010, <http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf> (28.07.2011).

[39] Die Liste der LIDO Events und ihre deutsche Übersetzung unter: <http://lido.vocnet.org/eventType> (13.07.2011).

Place, Event Actors (die an dem Ereignis beteiligten Personen oder Institutionen) sowie diverse andere Umstände, die zu dem Ereignis geführt haben.[40]

Der entscheidende Vorteil von LIDO gegenüber dem Dublin Core besteht in seiner grundsätzlichen Detailliertheit. Diese kann bei dem Mapping allerdings zu Komplikationen führen, da die detaillierten Informationen in LIDO sehr tiefgreifend verschachtelt werden und dabei ihre Ausprägung nicht in den Elementennamen, sondern in ihren Werten finden. LIDO Elemente werden abstrakt benannt und ihre Spezifizierung erfolgt in ihren Werten. Diese Spezifizierung kann bspw. durch die Eingabe eines Identifiers aus dem LIDO Vokabular[41] erfolgen, der diesem Objekttyp entspricht. Auf der Abbildung 3a verweist der Wert *lido00001* auf einen Acqusition Event. Eine weitere Möglichkeit der Spezifizierung besteht in der Eingabe von Uniform Resource Identifier (URI)[42] des Objekttyps (Abbildung 3b). Da in den beiden Fällen sich die CRM-Klasse erschließen lässt, die dem Objekt entspricht, wird das Mapping wesentlich erleichtert:

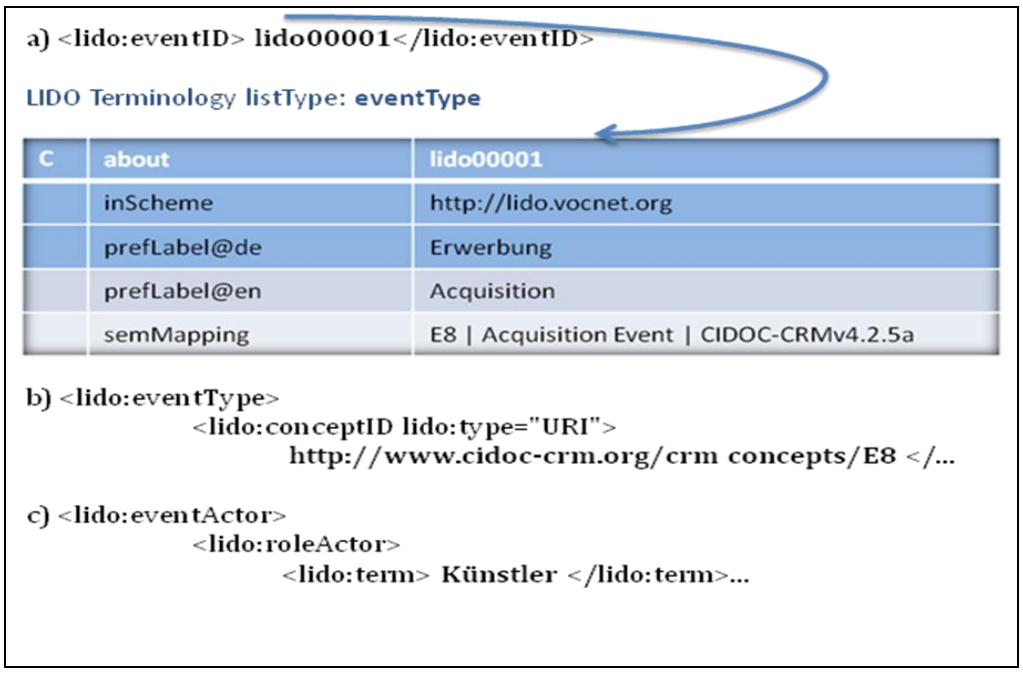


Abbildung 3: Spezifizierung der LIDO Konzepte.

Dennoch ist häufiger eine andere Notation anzutreffen. Anstelle eines Identifiers oder einer URI wird ein Begriff aus der LIDO Terminologie angegeben, eine für das Mapping durchaus beeinträchtigende Gegebenheit. So werden z.B. für das Erschließen des Urhebers

40 Siehe Coburn, E. & etc.: LIDO – Lightweight Information Describing Objects Version 1.0, S. 4.

41 Die komplette Liste von Begriffen und das Begriffssuchformular unter: LIDO Terminology, <http://lido.vocnet.org/lidoTerminologySearch.php> (5.06.2011).

42 Das Konzept der URI wird im Kapitel 2.6.1 näher behandelt.

einer Ressource, dem in DC nur ein einziges Feld <dc:creator> entspricht, im LIDO eine Reihe an Alternativen aus dem eigenen Vokabular verwendet. Das Feld <lido:eventActor> verweist zunächst auf einen abstrakten Ereignisakteur. Der Typ des Ereignisakteurs wird erst durch die Angabe seiner Rolle in dem Kinderelement <lido:roleActor> konkretisiert (Abbildung 3c). Der Term "Künstler" kann zwischen "Künstlerin" und "Künstler/in" variieren. Als Urheber der Ressource kann auch eine Reihe von weiteren Begriffen in Frage kommen, bspw. *Autor, Architekt, Bildhauer, Fotograf, Hersteller, Maler*.[43] Auf die exakt gleiche Weise werden auch Akteure in anderen Rollen definiert: *Auftraggeber*, *Verleger*, *Herausgeber*. Dies entspricht unterschiedlichen Darstellungen im CRM und wirkt sich enorm auf den Mapping-Aufbau aus.

Für die Online-Darstellung der Metadaten auf dem Portal ist es von Vorteil, wenn die Informationen so spezifisch wie möglich definiert werden. Anstelle des einfachen und allgemein definierten Begriffs *Urheber* kann ein konkreter Ausdruck – *Fotograf* oder *Bildhauer* – verwendet werden. Für die Interpretation der Metadaten, die sich als Grundlage der Mapping-Prozesse erweist, wird diese Spezifizierung jedoch zum Problem. In diesem Fall muss man Zeichenkettenabfragen durchführen, um festzustellen, um welchen Typ des Ereignisses und Ereignisakteurs es handelt. Als letzte Anmerkung diesbezüglich sei der Hinweis darauf gegeben, dass die Abgleiche stets in Abhängigkeit von der gegebenen natürlichen Sprache durchzuführen sind.

2.2.4 Encoded Archival Description (EAD)

Das Format EAD ist ein Kodierungsstandard für maschinenlesbare Findmittel.[44] Das Format wurde in den 1990er Jahren von der *Society of American Archivists* entwickelt. Weniger Jahre später wurde EAD in Deutschland eingesetzt. Das Bundesarchiv und vor allem seine Vizepräsidentin, Frau Prof. Menne-Haritz, haben in diesem Zusammenhang eine besonders wichtige Rolle gespielt.[45] Mit der Übernahme des EAD Formats ergab sich für die deutschen Archivare die Möglichkeit, Informationen über die Findmittel detailliert zu erfassen**.** Das EAD Format kann in internationale Kontexte integriert werden und bietet

[43] Die Auflistung aller Akteurrollen siehe unter: <http://lido.vocnet.org/roleActor> (5.06.2011).

[44] Die komplette Beschreibung des Standards bietet: *Encoded Archival Description Tag-Library Version 2002*, Arbeitsgruppe EAD Society of American Archivists & des Büros für Netzwerkentwicklung und MARC Standards der Library of Congress. Chicago, 2003. Aus dem Englischen von Löbnitz, A. & Menne-Haritz, A., 2006.

[45] Siehe Menne-Haritz, A.: EAD in Germany. Bundesarchiv, Berlin, v. 29.08.2008, <http://www.archivists.org/publications/proceedings/EAD@10/MenneHaritz-EAD@10.pdf> (10.06.2011).

eine vielschichtige Präsentation, die die ganze Bestandsübersicht abbilden lässt. Außerdem lässt es Erläuterungen und Bearbeitungsdokumentationen auf allen Ebenen und an beliebigen Stellen zu.[46]

Im Unterschied zu den Dublin Core Elementen, die durch zusätzliche Qualifikatoren spezifiziert werden (Abschnitt 2.2.2) und LIDO, dessen Elemente durch die Anwendung der LIDO Terminologie detaillierter bestimmt werden (Abschnitt 2.2.3), erfolgt die Spezifikation des EAD direkt mittels seiner Syntax, nämlich durch die Verschachtelung seiner Elemente. Die Angabe des ganzen Pfades, unter Umständen mit Attributen, macht ein genaues Retrieval möglich.

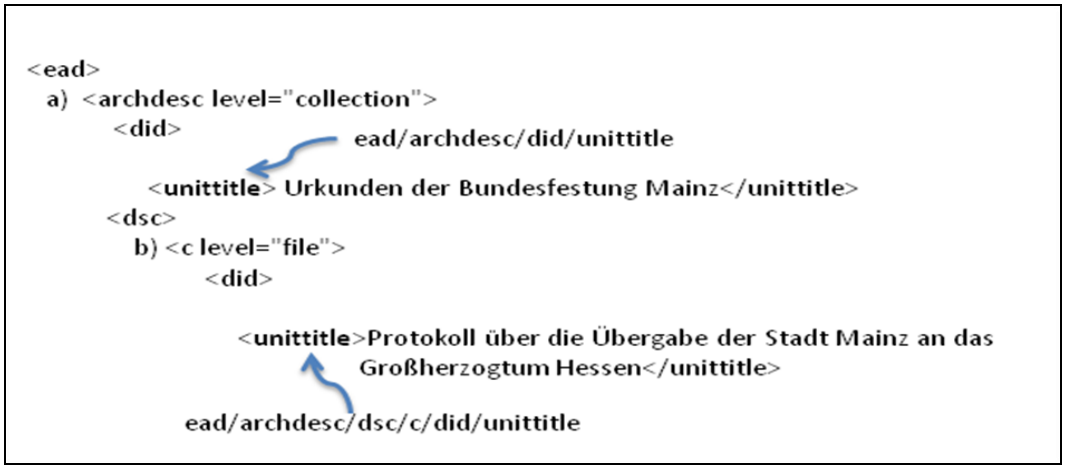


Abbildung 4: Ausschnitt aus der EAD Datei: Sammlung und Gegenstand Ebenen.

Wie auf der Abbildung 4 gezeigt wird, kann das Element <unittitle> mehrmals vorkommen. Mit Ausnahme von wenigen Elementen trifft es auf einen Großteil der EAD-Elemente zu. Im ersten Beispiel bezieht sich das Element (siehe Abbildung 4a) auf den Titel einer Archivaliensammlung. Seine zweite Anwendung findet auf der Ebene der einzelnen Archivalien statt: hier deutet das Element auf den Titel einer einzelnen Archivalie. Dabei kann eine Sammlung aus mehr als hundert Archivalien bestehen, diese verfügen idealerweise über ihren eigenen Titel. Solche hierarchischen Strukturen bergen für das Mapping eine Reihe an Schwierigkeiten und erfordern den Aufbau von rekursiven und dementsprechend komplexen Querries.

[46] Zur Funktion, Nutzen und Verwendung des EAD-Standards unter: <http://www.bundesarchiv.de/imperia/md/content/bundesarchiv_de/fachinformation/ead-strategie.pdf> (18.07.2011).

2.3 Metadaten und Ontologien: Unterschiede und Ansatz im Bereich der Wissensrepräsentation

In den vorherigen Abschnitten dieses Kapitels wurde auf die Nachteile der einzelnen Austauschformate im Hinblick auf das bevorstehende Mapping eingegangen. Die besprochenen Austauschformate sind teilweise von hoher Komplexität, wie bspw. das EAD. Sie weisen eine zu große Menge an Feldern auf (MARC). Sie sind zu spezifisch (LIDO), oder zu allgemein (DC), um als generelles Modell für alle Kulturdaten verwendet werden zu können. Um die Informationen aus den unterschiedlichen Metadatenstrukturen zu harmonisieren, transformiert man sie in dem DDB-Projekt in ein einheitliches CIDOC CRM Datenmodell. Das CIDOC CRM wurde speziell für Kulturdatenmodellierung mit dem folgenden Ziel entwickelt: "*for the exchange and integration of heterogeneous scientific documentation of museum collections*"[47]. Außerdem stellt es eine Ontologie in Sinne der Informatik dar.[48] Der Unterschied zwischen Ontologien auf einer Seite und Metadaten wie DC, EAD und LIDO auf der Anderen besteht vor allem in ihrem jeweiligen funktionalen Zweck:

> "Metadata have a completely different scope and function in comparison with ontologies. Metadata are used to describe, identify, facilitate the access, usage and management of (digital) resources. Ontologies define entities in a more abstract level, with the intention of conceptualizing a domain of interest. They do not provide specific elements for the description of a resource, but a general definition of the basic notions of a field and the relations between them."[49]

Während Metadaten für die Beschreibung der Ressourcen eingesetzt werden, um ein maschinengestütztes Suchen zu ermöglichen, definieren Ontologien die Begriffe einer Wissensdomain und Beziehungen zwischen ihnen und unterstützen die Maschinen dabei, die Inhalte im Web interpretieren zu können. Die Suche lässt sich dadurch viel gezielter gestalten. Darüber hinaus wird es mittels Ableitungsregeln, die einen wichtigen Bestandteil der Ontologien bilden, möglich, weitere Schlüsse aus vorhandenen Informationen zu ziehen (Abschnitt 2.4.2).

---

[47] Doerr M., Crofts, N., Gill, T., Stead S. & Stiff M.: Definition of the CIDOC Conceptual Reference Model, Version 5.0.2, Januar 2010, S. ii.

[48] Siehe das anschließende Kapitel 2.4 über Ontologien.

[49] Doerr, M., Stasinopoulou, T., Bountouri, L., Kakali, C., Lourdi, I., Papatheodorou C. & Gergatsoulis, M.: Ontology-Based Metadata Integration in the Cultural Heritage Domain, in: *Asian digital libraries: looking back 10 years and forging new frontiers*: *10th International Conference on Asian Digital Libraries, ICADL, Hanoi, Vietnam.* Berlin, New York: Springer, 2007. S. 166.

Ferner unterscheiden sich die beiden Aspekte in den menschlichen Faktoren, die für ihr Design von Bedeutung sind. Metadaten werden im Allgemeinen durch Menschen erstellt, bearbeitet und betrachtet. Deshalb spielt das menschliche Verständnis eine große Rolle bei der Erstellung von Metadaten. Im Gegensatz hierzu wird das Design einer Ontologie vielmehr von Vollständigkeit (*completeness*) und logischer Korrektheit (*logical correctness*) als vom menschlichen Verständnis geprägt.[50]

Eine Demonstration der unterschiedlichen Ansätze der beiden Konzepte im Bereich der Wissensorganisation wird von Martin Doerr geliefert.[51] Es handelt sich dabei um zwei unterschiedliche Metadatensätze, die einen gewissen Bezug auf die Jalta Konferenz, die im Februar 1945 stattfand, aufweisen. Dies war die Veranstaltung, die offiziell das Ende des Zweiten Weltkriegs bedeutete. Das erste Metadatensatz beschreibt das Jalta- Abkommen:

Type:	Text
Title:	Protocol of Proceedings of Crimea Conference
Title.Subtitle:	II. Declaration of Liberated Europe
Date:	February 11, 1945.
Creator:	The Premier of the Union of Soviet Socialist Republics
	The Prime Minister of the United Kingdom
	The President of the United States of America
Publisher:	State Department
Subject:	Postwar division of Europe and Japan

Das zweite Metadatensatz betrifft das Foto, das dieses Ereignis festhält, und stellt sich wie folgt dar:

Type:	Image
Title:	Allied Leaders at Yalta
Date:	1945
Publisher:	United Press International (UPI)
Source:	The Bettmann Archive
Copyright:	Corbis
References:	Churchill, Roosevelt, Stalin

[50] Siehe Doerr, M., Hunter, J. & Lagoze, C.: Towards a Core Ontology for Information Integration, v. 2003, S. 2, <http://journals.tdl.org/jodi/article/view/92/91> (27.07.2011).

[51] Siehe Doerr, M.: The CIDOC CRM – an Ontological Approach to Semantic Interoperability of Metadata, v. 2002, S. 3-4, <http://www.cidoc-crm.org/docs/ontological_approach.pdf> (17.07.2011).

Der entscheidende Punkt dabei ist, dass beide Metadatensätze bis auf das Jahr "1945" keine weitere Übereinstimmung aufweisen. Diese Übereinstimmung ist jedoch nicht ausreichend, um diese zwei Einträge mit einander in Verbindung zu setzen. Die gesuchte verbindende Information (*integrating piece of information*) kommt aus dem dritten Metadatensatz, der aus dem Getty Thesaurus of Geographic Names (TGN)[52] stammt:

TGN Id:	7012124
Names:	Yalta (C,V), Jalta (C,V)
Types:	inhabited place(C), city (C)
Position:	Lat: 44 30 N,Long: 034 10 E
Hierarchy:	Europe (continent) <– Ukrayina (nation) <– Krym (autonomous republic)
Note:	Located on S shore of Crimean Peninsula; site of conference between Allied powers in WW II in 1945; is a vacation resort noted for pleasant climate, & coastal & mountain scenery; produces wine, canned fruit & tobacco products.
Source:	TGN, Thesaurus of Geographic Names

Durch den dritten Eintrag lässt sich die Verbindung zwischen *Crimea*, *Yalta* und *Krym* herstellen. Mit weiteren Thesauren werden sich eventuell auch weitere Beziehungen erschließen lassen, dass bspw. "Premier of the Union of Soviet Socialist Republics" und Joseph Stalin, "Prime Minister of the United Kingdom" und Churchill sowie "President of the United States of America" und Roosevelt sich jeweils auf eine Person beziehen.

Mit Hilfe von Thesauren wurde es möglich, Verbindungen zwischen einzelnen Objekten festzustellen, die beim Verarbeiten der ursprünglichen Metadaten nicht unmittelbar erkennbar waren. Ein anderes Problem dieser Suche bleibt dennoch bestehen, weil das gesuchte Ereignis, die Konferenz von Jalta, nur indirekt in diesen Metadatensätzen gewahrt wird. Eine Ontologie bietet dagegen die Möglichkeit, solche versteckte Informationen (*hidden constants)* explizit auszudrücken, was die Suche nach ihnen wesentlich vereinfacht.[53] Im CIDOC CRM kann der ganze Sachverhalt bspw. folgendermaßen ausdrückt werden:

[52] Getty Thesaurus of Geographic Names Online: <http://www.getty.edu/vow/TGNServlet?english=Y&find=7012124&place=&page=1&nation=> (12.07.2011).

[53] Siehe Doerr, M.: The CIDOC CRM – an Ontological Approach to Semantic Interoperability of Metadata, S. 4.

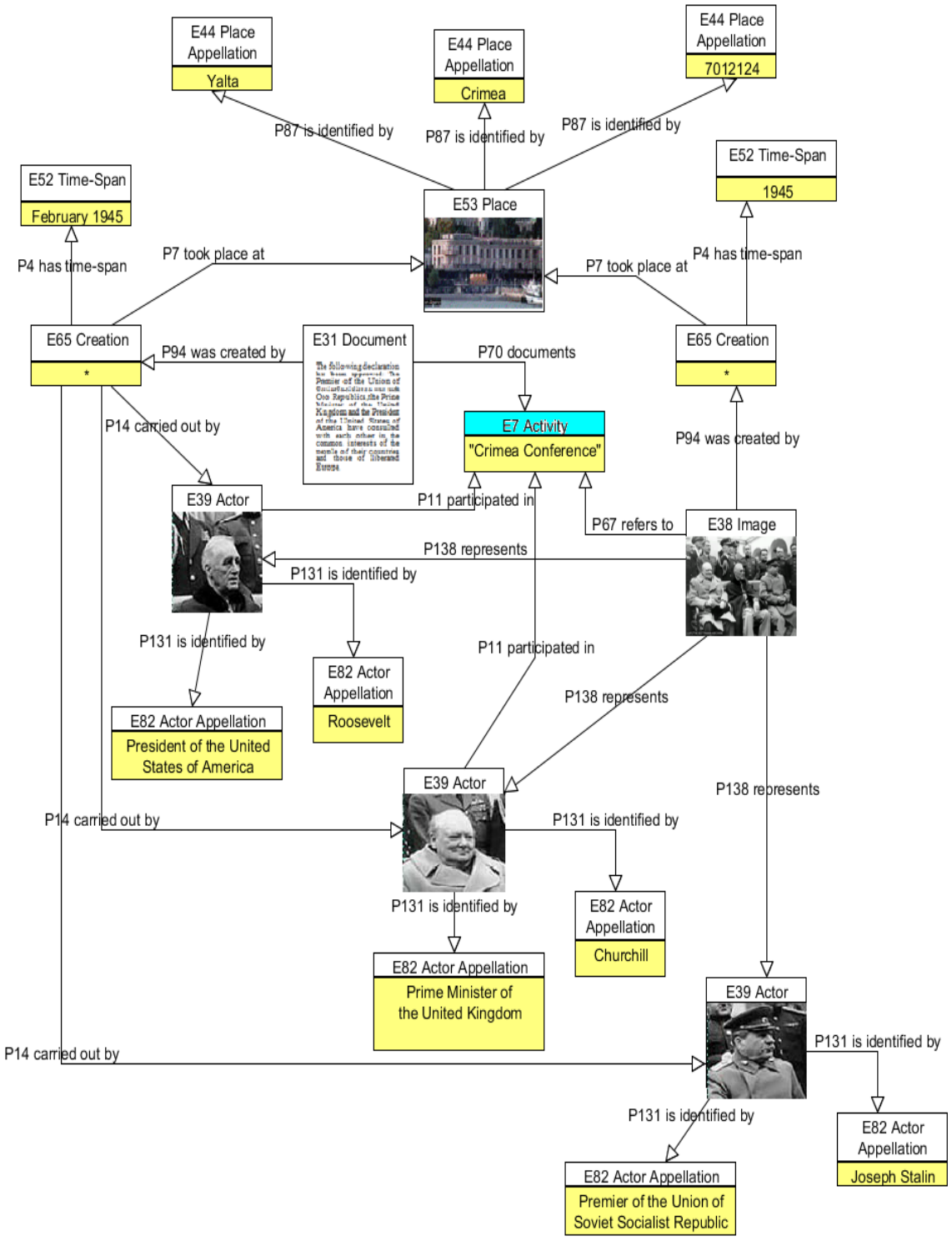


Abbildung 5: Beispiel der CRM Ontologie

In diesem Zusammenhang muss dennoch betont werden, dass eine derartige Struktur nur unter der Voraussetzung entstehen kann, dass für die Beschreibung der Objekte von Anfang an eine Ontologie angewendet wird. Mittels einfacher Mappings von o.g. Metadatensätzen

auf das CIDOC CRM lässt sich nicht feststellen, dass es sich bei der *Crimea Conference* um eine Aktivität handelt. Diese Begebenheit kann eventuell nach dem Mapping mittels Reasoning-Mechanismen korrigiert werden. Im Folgenden werden die wichtigsten Ontologie-Konzepte näher erläutert.

2.4 Ontologien

2.4.1 Was ist eine Ontologie?

Der Begriff "Ontologie" findet seinen Ursprung in der Philosophie und bezeichnet dort "die Lehre vom Sein – genauer: von den Möglichkeiten und Bedingungen des Seienden"[54] und zeichnet sich in dem Versuch aus, "Objekte der realen und gedachten Welt in Kategorien zu unterteilen sowie deren Eigenschaften und Abhängigkeiten zu analysieren".[55] Auch Ontologien im Sinne der Informationstechnologie beschäftigen sich mit der Kategorisierung von Objekten.

Für Ontologie im Sinne der Informationstechnologie gibt es zahlreiche Definitionen. Die populärste stammt von Thomas Gruber: "*An ontology is an explicit specification of a shared conceptualization.*"[56] Der Begriff "Ontologie" bezieht sich dabei auf eine formale Abbildung eines bestimmten Anwendungsbereiches, mittels einer maschinenlesbaren Menge von Begriffen, die für diesen Anwendungsbereich relevant sind, und der zwischen diesen Begriffen bestehenden Beziehungen und gegebenenfalls Ableitungsregeln. Die Ontologien werden zum Wissensaustausch zwischen Menschen und/oder Maschinen eingesetzt. Voraussetzung dafür ist die Einigung aller Beteiligten auf die jeweiligen Begriffe und Beziehungen zwischen ihnen (*shared conceptualization*).[57] An dieser Stelle sei auch auf die Definition der Ontologie der Entwickler des CRM verwiesen: "*An*

[54] Hesse, W.: Aktuelles Schlagwort Ontologie(n), in: Informatik-Spektrum, Vol. 25, N. 6, 2007, S. 477.

[55] Stuckenschmidt, H.: Ontologien : Konzepte, Technologien und Anwendungen. Berlin: Springer, 2010, S. 3.

[56] Hier zit. nach: Hesse,W., Krzensk, B.: Ontologien in der Softwaretechnik, 2004, S. 2. Original: Gruber, T. R.: A Translation Approach to Portable Ontology Specifications, 1993, S. 1.

[57] Der Begriff "shared" wird in dem Originaltext von Gruber (1993) nicht verwendet, wird jedoch aufgrund seiner besonderen Aussagekräftigkeit in den meisten Literaturquellen angegeben. Stuckenschmidt weist jedoch darauf hin, dass die Voraussetzung einer gemeinsamen Nutzung von Ontologien bestimmte Modelle ausschließt, wie z.B. Modelle, die von nur einer Person bedient werden. Siehe Stuckenschmidt, H.: *Ontologien: Konzepte, Technologien und Anwendungen*, S. 22.

ontology can be seen as a model of possible states of affairs."[58] Diese Erkenntnis spiegelt sich in der Organisation der CRM Klassenhierarchie wieder (siehe Kapitel III). Im Folgenden wird die Grundstruktur einer Ontologie kurz erklärt.[59]

2.4.2 Ausgangselemente

Lexikon: Das durch Ontologien ausgedrückte Wissen wird explizit in standardisierten Vokabularen in Form von lexikalischen Einträgen angegeben, welche ihrerseits Begriffe sowie Beziehungen zwischen ihnen und Ableitungsregeln bezeichnen.

Begriffe: Mit einem Begriff beschreibt man eine Menge von Dingen, die gemeinsame Eigenschaften aufweisen und für einen bestimmten Anwendungsbereich relevant sind. Begriffe werden auch als Klassen bezeichnet. Diese können in einer taxonomischen Struktur – mit Über- und Unterbegriffen – angeordnet werden. Beispiele für Begriffe in der Abbildung 6 sind *Person*, *Angestellter* und *Projekt*. Dabei können verschiedene Worte auf denselben Begriff verweisen. Die Worte "Angestellter" und "Mitarbeiter" verweisen beide auf den Begriff *Angestellter*.

Relationen: Begriffe stehen in einer auf semantischen Relationen basierenden Beziehung zueinander. Zum Beispiel bindet die Relation *hat Mitarbeiter* den Begriff *Projekt* und *Person* zusammen. Die umgekehrte Verbindung ist die Relation *arbeitet in* (Abbildung 6). Die häufigste Art einer Ontologie im Kontext des *Semantic Web* ist eine Taxonomie. Eine unabdingbare Beziehung einer Taxonomie ist eine *is-a-Relation*, die für das Vererbungskonzept steht: durch diese Relation können Unterbegriffe mit ihren Überbegriffen in Verbindung gesetzt werden. Für Abbildung 6 gilt, dass *Angestellter* der Unterbegriff von *Person* ist und folglich alle seine Relationen erbt.

Ableitungsregeln: Mittels Ableitungsregeln wird neues Wissen aus den vorhandenen Informationen abgeleitet. Dies soll am folgenden Beispiel der Entwickler des *Semantic Web* veranschaulicht werden:

---

[58] Doerr, M., Plexousakis, D., Kopaka, K. & Bekiari, C.: Supporting Chronological Reasoning in Archaeology, v. 2004, S. 2.

[59] Basiert zum Teil auf: Ehrig, M. & Studer, R.: Wissensvernetzung durch Ontologien in: Pellegrini, T.: *Semantic Web X Wege zur vernetzten Wissensgesellschaft,* mit 4 Tabellen, Berlin, Heidelberg, New York: Springer, 2006, S. 472-472. Daraus stammt auch die Abbildung 6.

> "If a city code is associated with a state code, and an address uses that city code, then that address has the associated state code." A program could then readily deduce, for instance, that a Cornell University address, being in Ithaca, must be in New York State, which is in the U.S., and therefore should be formatted to U.S. standards."[60]

Ableitungsregeln werden weiter in der Arbeit an dem Reasoning Konzept im Kapitel 2.4.2 wieder aufgegriffen.

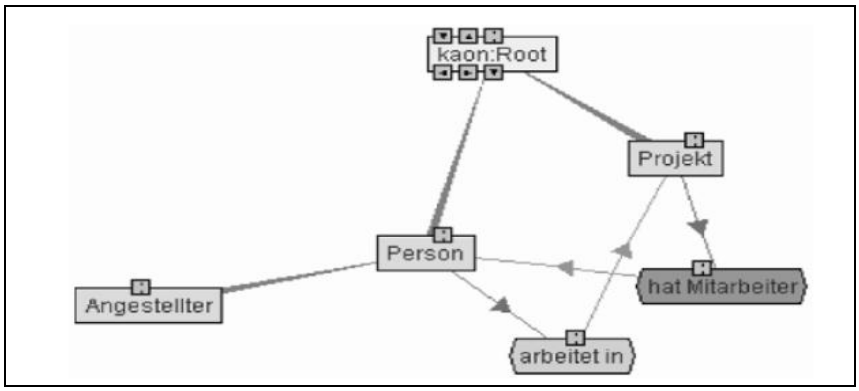


Abbildung 6: Beispiel einer Ontologie.

2.4.3 Ontologietypen

Das Wissen, das in Ontologien abgebildet wird, kann sich sowohl auf das Allgemeinwissen als auch auf spezielle Themengebiete und Vorgänge beziehen. Nach Guarino wird je nach Abstraktionsgrad (*level of generality*) zwischen vier Ontologie-Typen unterschieden (siehe Abbildung 7):[61]

1) *Top-level ontologies*: Allgemeine, domainunabhängige Konzepte;
2) *Domain ontologies:* Begriffe, die sich auf generische Domänen beziehen und dabei die Begriffe aus den entsprechenden Top-Level-Ontologien spezifizieren;
3) *Task ontologies:* Begriffe, die sich auf allgemeine Aktivitäten beziehen und dabei die Begriffe aus den entsprechenden Top-Level-Ontologien spezifizieren;

---

[60] Berners-Lee,T., Hendler, J. & Lassila,O.: The Semantic Web, *Scientific American*, Mai 2001, <http://old.hki.uni-koeln.de/temp/SemWebSeminal.pdf > (10.06.2011).

[61] Siehe Guarino, N.: Formal ontology in information systems, in: Guarino, N. (hrg.): *Formal ontology in information systems*. *Proceedings of the first international conference (FOIS'98), Trento, Italy, 6-8 June*. Amsterdam; Washington, DC: IOS Press; Tokyo: Omsha, 1998. S. 9-10. Die Abbildung 7 ist aus derselben Quelle entnommen.

4) *Application ontologies*: Begriffe, die auf konkrete Domäne/Aufgaben zugeschnitten sind, die in der Regel die Begriffe aus den entsprechenden Domain/Task Ontologien spezifizieren.

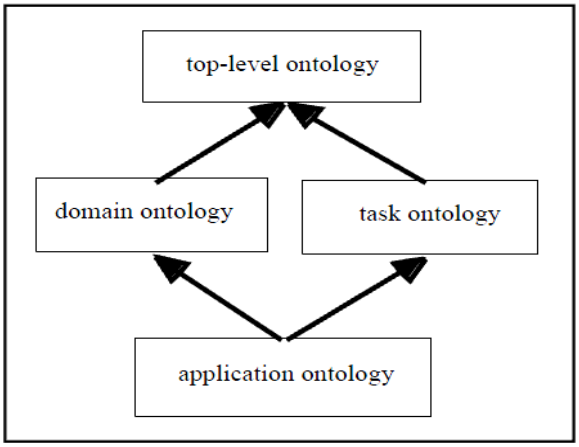


Abbildung 7: Ontologietypen nach Guarino.

2.4.4 Ontologiesprachen (OL)

In dem Kapitel III dieser Arbeit wird die Rede von CIDOC CRM sein. Es gibt jedoch eine Vielzahl von derzeit vorhandenen formalen Beschreibungssprachen, mit welchen Ontologien erstellt und verteilt werden können. Im Kontext des *Semantic Web* basieren diese Beschreibungssprachen auf der Repräsentationssprache RDF und ihren Modellierungsprinzipien. Hier ist u.a. Resource Description Framework Schema (RDFS) zu nennen, das eine Erweiterung des RDF darstellt.[62] Mittels RDFS kann eine sehr einfache Hierarchie von Klassen, Eigenschaften und Ableitungsregeln definiert werden. RDFS ist die einfachste Ontologiesprache im *Semantic Web* Kontext. Alle anderen OL bauen auf den Modellierungsprimitiven von RDFS auf und erweitern diese, wie z.B. Ontology Inference Layer (OIL), eine webbasierte Ontologiesprache, die für die Repräsentation von Ontologien im Rahmen des von der Europäischen Union geförderten Projektes *On-To-Knowledge* entwickelt wurde.[63] DARPA Agent Markup Language (DAML)[64] stellt ebenfalls eine Erweiterung von RDFS dar und wurde von der amerikanischen Regierung gefördert. Eine weitere mächtige OL ist die Web Ontology Language (OWL), eine

[62] Siehe RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation vom 10 February 2004, <http://www.w3.org/TR/rdf-schema/> (12.07.2011).
[63] Siehe Horrocks, I. & etc.: The Ontology Inference Layer OIL unter: <http://xml.coverpages.org/OIL-inference.pdf> (12.07.2011).
[64] Siehe DARPA Agent Markup Language unter: <http://www.daml.org/> (12.07.2011).

Spezifikation des W3C, die weit über die Ausdrucksmächtigkeit von RDFS hinaus geht.[65]

2.4.5 Anwendungsgebiete

Zu den Anwendungsgebieten von Ontologien zählen alle sich mit der Kommunikation, dem automatischen Schließen und der Wissensrepräsentation befassenden Bereiche der Informatik: Künstliche Intelligenz, Informationssysteme, Knowledge Engineering, maschinelle Sprachverarbeitung, Electronic Commerce, intelligente Informationsintegration, Information Retrieval und Wissensmanagement. Auch in der Wirtschaftsinformatik, Softwaretechnik, Multimedia-Kommunikation haben Ontologien in letzter Zeit eine große Bedeutung erlangt.[66] Das weltweite Interesse an den Ontologien verdankt sich vor allem der *Semantic Web Initiative*, die das Thema des nächsten Kapitels sein wird.

2.5 Semantic Web

2.5.1 Idee und Zielsetzung

Wie es in der Einleitung schon erwähnt wurde, wird DDB als ein sehr wichtiger Datenlieferant für das *Semantic Web* angesehen.[67] Von seinem Entwickler, dem WWW-Schöpfer Tim Berners-Lee und seinen Kollegen, wird *Semantic Web* wie folgt beschrieben:

> "A new form of Web content that is meaningful to computers …
> The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users."[68]

Die meisten Inhalte im Web sind für den Menschen zum Lesen konzipiert. Computer sind fähig solche Informationen wie Layoutvorgaben zu interpretieren, sind jedoch nicht im Stande, die inhaltlichen Informationen ohne weiteres semantisch zu verarbeiten (to

---

65 Siehe Web Ontology Language (OWL), W3C Recommendation 10 Februar 2004, <http://www.w3.org/TR/owl-features/> (13.07.2011).

66 Siehe Hesse, W.: Aktuelles Schlagwort Ontologie(n), 2007, S. 477.

67 Siehe Christen, M.: Machbarkeitsstudie zum Aufbau und Betrieb einer "Deutschen Digitalen Bibliothek". Die IT-Architektur der DDB auf der Basis des Fachkonzeptes der Bund-Länder-Fachgruppe, Frankfurt am Main, den 23.07.2008, S. 10, <http://www.deutsche-digitale-bibliothek.de/pdf/machbarkeitsstudie_20080723.pdf> (11.08.2011).

68 Berners-Lee,T., Hendler, J. & Lassila,O.: The Semantic Web, *Scientific American*, Mai 2001, <http://old.hki.uni-koeln.de/temp/SemWebSeminal.pdf > (10.06.2011).

manipulate meaningfully).[69] Im Rahmen des *Semantic Web* sollen dagegen die Web-Dokumente mit "Semantik" in Form von Metadaten versehen werden, die dazu dienen, die Informationen, die diese Dokumente enthalten, auch für Maschinen verständlich zu machen. Die Suchmaschinen und Agenten werden somit bei der Aufgabe unterstützt, geforderte Informationen unmittelbar und effizient zu finden und miteinander zu verknüpfen. Die benötigten Grundlagen an Metadaten und Verknüpfungsregeln werden dafür durch Ontologien geliefert.[70] Mit Hilfe von Ontologien können zwei Programme über ihre Aufgaben und Ergebnisse miteinander kommunizieren und die Informationen zum menschlichen Nutzen verarbeiten:

> "The computer doesn't truly "understand" any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user."[71]

2.4.2 Das Reasoning

Von besonderer Bedeutung für die Nutzung der Ontologien für das *Semantic Web* sind die zusätzlichen Informationen über die Konzepte, die sich aus den existierenden Daten ableiten lassen. Dieser Ableitungsprozess wird als Reasoning bezeichnet.[72] Im Folgenden wird ein äußerst simples Reasoning-Beispiel demonstriert, das mittels einer taxonomischen Regel durchgeführt wurde:

Aus den Beziehungen *Flipper isA Dolphin* und *Dolphin isA Mammal* lässt sich eine neue Beziehung folgern: *Flipper isA Mammal*.[73]

Es existieren verschiedene Reasoning-Mechanismen, mit deren Hilfe man aus bereits bekanntem Wissen neues Wissen generieren kann. Mittels des sog. kausalen Reasoning lassen sich z.B. die Ursache-Wirkungsbeziehungen zwischen Events darstellen, die auf den ersten Blick nicht zu sehen sind. Dadurch wird den Menschen ermöglicht, eine sinnvolle

---

69 Siehe ebd.
70 Siehe Hesse, W.: Aktuelles Schlagwort Ontologie(n), S. 478.
71 Berners-Lee, T., Hendler, J. & Lassila, O.: The Semantic Web, *Scientific American*, Mai 2001, <http://old.hki.uni-koeln.de/temp/SemWebSeminal.pdf > (10.06.2011).
72 Auch Schlussfolgerung, Folgerung, Deduktion, Inferenz, Rückschluss
73 Siehe das Beispiel unter: <http://www.w3.org/standards/semanticweb/query> (13.07.2011).

Reihenfolge in Ereignissen zu erkennen, die den Anschein des Zufälligen haben. Das kausale Reasoning, das 1980 von dem Philosophen John Mackie als "*the cement of the universe*"[74] definiert wurde, lässt einige Erkenntnisse zu: bspw. können die Ereignisse, die regelmäßig zusammen auftreten, als kausal verbunden angesehen werden. Die Events, die eine Ursache darstellen, gehen logischerweise den Events, die als Effekt eintreffen, voraus.[75] Im Kapitel 3.3.1 wird die Rede von temporalem und räumlichem Reasoning sein, die sich im Kontext der CIDOC CRM Ontologie anbieten.

2.5.3 Vokabulare

Eine weitere wichtige Komponente zur Realisierung des *Semantic Web* sind die Vokabulare, die wie folgt definiert werden:

> "On the Semantic Web, vocabularies define the concepts and relationships (also referred to as "terms") used to describe and represent an area of concern. Vocabularies are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms."[76]

Verschiedene Einrichtungen setzen Vokabulare ein, um ihr Wissen zu organisieren. Ein weiteres Beispiel ihrer Anwendung betrifft die Ambiguität der Begriffe, die aus verschiedenen Datenbeständen stammen. Der TGN Auszug im Kapitel 2.3 ermöglicht es, die Verbindung zwischen den Begriffen *Yalta* und *Jalta* herzustellen. Neben den Ableitungsregeln dienen auch Vokabulare zum Zwecke des Reasoning, wie das folgende Beispiel der WC3 Homepage deutlich werden lässt:

> "Consider, for example, the application of ontologies in the field of health care. Medical professionals use them to represent knowledge about symptoms, diseases, and treatments. Pharmaceutical companies use them to represent information about drugs, dosages, and allergies. Combining this knowledge from the medical and pharmaceutical communities with patient data enables a whole range of intelligent applications such as decision support tools that search for possible treatments; systems that monitor drug efficacy and possible side effects; and tools that support epidemiological research."[77]

---

[74] Mackie, J. L.: The Cement of the Universe. Oxford: Clarendon Press, 1980, S. 2.

[75] Siehe Magliano, J. P., Pillow, B.H.: Causal Reasoning, in: Guthrie, J. W.: Encyclopedia of education, Volume 1, New York [u.a.]: Macmillan Reference USA [u.a.], 2003, S. 1425-1427, <http://groups.psych.northwestern.edu/gentner/papers/GentnerLoewenstein02a.pdf > (13.07.2011).

[76] Siehe Vocabularies: <http://www.w3.org/standards/semanticweb/ontology> (29.07.2011). Im Prinzip passt diese Definition von Vokabularen auch auf Vokabulare im Bezug auf Ontologien. Die Ontologien verwenden jedoch eine Sammlung von komplexeren und äußerst formalen Begriffen, siehe ebd.

[77] Ebd.

Neben Ontologien und Vokabularen benötigt das *Semantic Web* eine weitere Komponente: das Resource Description Framework (RDF), das im anschließenden Kapitel 2.6 behandelt wird.

2.6 Resource Description Framework (RDF)

RDF bildet eine weitere wichtige Basiskomponente des *Semantic Web* und wird zur Beschreibung, Repräsentation und zum Austausch von Ressourcen im Web eingesetzt.[78] Mittels RDF wird die Suche im Web und die automatische Verarbeitung von Inhalten unterstützt. RDF kann sowohl als ein graphisches Modell dargestellt werden als auch eine Repräsentation in der XML-Syntax haben. Die grundlegende Idee vom RDF ist, dass alle Objekte durch sogenannte *Uniform Resource Identifiers* (URIs) identifiziert und durch einfache Aussagen (*RDF-Statements*) beschrieben werden können.[79] Im Folgenden werden die wichtigsten RDF-Begriffe näher erläutert.

2.6.1 RDF-Ressource

Eine Ressource ist ein Objekt, das über einen eindeutigen Bezeichner "URI" identifiziert wird. Dieser kann bspw. die Form eines URN (Uniform Resource Name) annehmen, der stabile Referenzen auf digitale Objekte mittels Namensräume herstellt. So kann z.B. ein Buch mit dem URN urn:nbn:de:gbv:089-3321752945 identifiziert werden. Die Vergabe eines URN erfolgt in Deutschland zentral bei der Deutschen Nationalbibliothek.[80] Eine weitere Form der URI, Uniform Resource Locator (URL), bestimmt dagegen Ressourcen durch Identifikation ihres Speicherorts.[81] So können die Ressourcen im Internet erreicht werden: http://www.example.org/book oder lokal: file:///C:/temp/document.txt.

Neben URNs und URLs existieren auch URI References[82], die sich aus einer URL und einem in einem bestimmten Kontext eindeutigen Identifier (*fragment identifier*) zusammensetzen. So werden auch die betreffenden Ressourcen im DDB-Projekt über die

---

78 Weiterführende Informationen über RDF sind auf der Website des W3C zu finden: <http://www.w3.org/TR/rdf-syntax/> (13.07.2011).

79 Siehe Eckstein, R. & Eckstein, S.: XML und Datenmodellierung, Dpunkt-Verl., 2004, S. 235.

80 Zur Strategie der Deutschen Nationalbibliothek bei der Vorgabe von URNs und weitere Informationen zu dem Schema siehe Persistent Identifier: <http://www.persistent-identifier.de/?link=3352> (10.06.2011).

81 Siehe Berners-Lee, T.: Uniform Resource Locators (URL): <http://tools.ietf.org/pdf/rfc1738.pdf> (13.07.2011).

82 Mehr dazu im RDF Primer, W3C Recommendation 10 Februar 2004 unter: <http://www.w3.org/TR/rdf-syntax/> (13.07.2011).

Projekt-URL *http://www.ddb.de/* identifiziert, indem direkt nach dem Schrägstrich ein eindeutiger DDB-Identifier der Ressource folgt.[83]

2.6.2 RDF-Tripeln

Eine Aussage wird im RDF mit einem Tripel repräsentiert. Jedes Tripel besteht aus drei notwendigen Teilen: DOMAIN, PROPERTY und RANGE. Diese Beziehung entspricht im grammatikalischen Sinne einer Subjekt-Prädikat-Objekt Beziehung, verläuft vom Subjekt zum Objekt und wird mit dem Prädikat benannt. Subjekt und Prädikat bestehen in einem Tripel immer als Ressourcen. Das Objekt kann dagegen entweder eine Ressource oder ein Literal sein. Unter Literalen versteht man einfache Zeichenketten, die unter Umständen noch durch die Angabe des Attributs *rdf:type* spezifiziert werden können. Bspw. wird der Typ einer Instanz der CRM Klasse *E67 Birth* auf "date" bestimmt. Im Folgenden wird ein RDF-Tripel anhand eines Dublin Core Beispiels näher erläutert.

Der Informationssatz "Harry Potter and the Order of the Phoenix wurde von Joanne K. Rowling im Jahre 2000 geschrieben" kann in DC wie folgt strukturiert werden:

```
<dc:title> Harry Potter and the Order of the Phoenix </dc.title>
<dc:creator> Joanne K. Rowling </dc.creator>
<dc:date> 2000 </dc:date>
```

Dieser DC Eintrag lässt sich mit drei Subjekt-Prädikat-Objekt Beziehungen beschreiben und dementsprechend auf drei RDF-Tripeln abbilden. Das RDF-Modell ist nicht abhängig von einer speziellen Darstellungsform, am häufigsten wird jedoch die XML Notation verwendet:

```
<rdf:Description rdf:about="http://www.ddb.de/102">
        <dc:title>Harry Potter and the Order of the Phoenix </dc:title>
        <dcterms:created>2000</dcterms:created >
        < dc:creator>Joanne K. Rowling</dc:creator>
</rdf:Description>
```

Graphisch lässt sich das RDF-Modell mit benannten Knoten und Kanten darstellen. Jeder Tripel wird durch einen Graph visualisiert und besteht jeweils aus zwei Knoten und einer gerichteten Kante. Letztere verbindet beide Knoten miteinander. Die Ressourcen werden graphisch durch Ellipsen veranschaulicht, und die Literale durch Rechtecke symbolisiert.

[83] Zu den DDB-Identifier Kapitel 4.2.2.

Die Verbindung zwischen einem Subjekt und einem Objekt wird durch eine mit dem Prädikat bezeichnete, gerichtete Kante dargestellt. Die Abbildung 8 visualisiert das benannte RDF-Model in der vorgestellten grafischen Konvention.

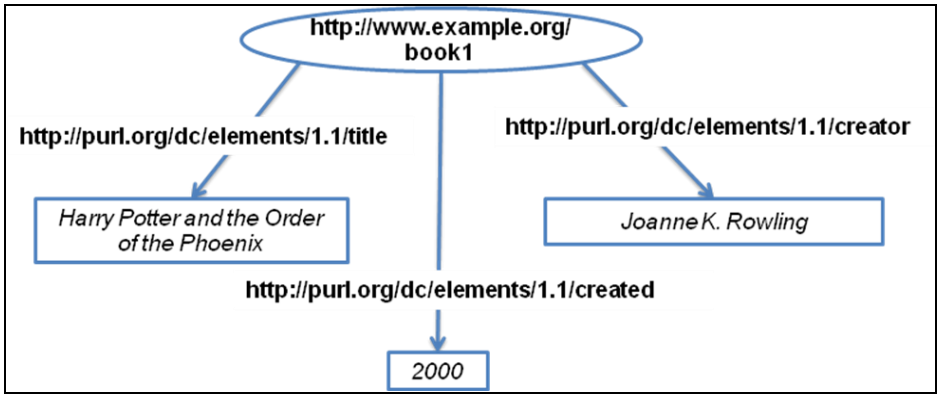


Abbildung 8: RDF-Graph.

Zusammenfassend bietet das RDF die Möglichkeit, Ressourcen durch ihre Eigenschaften zu beschreiben und diese durch ihre eindeutige Identifizierung mit anderen Ressourcen in Verbindung zu setzen. Die dadurch entstandene Vernetzung der Daten, ihre Freistellung und Aufrufbarkeit im Web spiegelt das Konzept von *Linked Open Data* wider.

2.6.3 Linked Open Data (LOD)

Beim Linked Open Data handelt es sich um ein Konzept, das im Wesentlichen vom Tim Berners-Lee stammt, und bezeichnet:

> "…data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets, and can in turn be linked to from external data sets."[84]

Mit RDF und URIs werden diese Daten über Domänen und Organisationsgrenzen hinweg mit Hilfe des World Wide Web miteinander verknüpft. Durch diese Verknüpfung entsteht ein Netz freier Datenbestände, die ohne jegliche Restriktionen kombiniert und weiterverwendet werden können.[85] Die Visualisierung von *Linked Open Data* erfolgt durch

[84] Berners-Lee, T.; Bizer, C. & Heath, T.: Linked Data – The Story So Far, to appear in Special Issue on Linked Data, <http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf> (29.07.2011).

[85] Siehe Von Lucke, J., Geiger, C. P.: Open Data Government – Frei verfügbare Daten des öffentlichen Sektors – Gutachten zur T-City Friedrichshafen, v. 03.12.2010,

die *Linked Open Data* Wolke. Ein Beispiel für eine derartige Wolke liefern Cyganiak und Jentzsch.[86] Einen Bestandteil ihrer Wolke bilden u.a. die *DBPedia*, die im Wesentlichen den Inhalt von Wikipedia enthält, und GeoNames, eine geografische Datenbank, die alle Länder umfasst und mehr als acht Millionen Ortsnamen enthält. [87]

2.6.4 Tripelstores und SPARQL

Für die Speicherung und Auswertung von RDF-Tripeln werden sog. Tripelstores eingesetzt, die zur Verwaltung der Aussagen des *Semantic Web* dienen. Die RDF Tripelstores bieten *Application Programming Interface*s (APIs)[88], Abfrage-Interfaces und integrierte Reasoning-Mechanismen an.[89] Die Tripel werden in dem Tripelstore indexiert und können dadurch abgefragt werden. Mit der SPARQL Query-Maschine werden die Abfragen an den Tripelstore durchführt.[90] SPARQL formuliert Anfragen in Form von Tripelmustern an RDF-Datenbestände und bekommt die Ergebnisse zurück, die diesem Muster entsprechen. Es existieren z.B. in einem RDF Datenbestand die Tripel *Flipper isA Dolphin* und *Dolphin isA Mammal*. Mit SPARQL kann der Benutzer eine Anfrage formulieren, die wie folgt aussieht: *Flipper isA ?species.* Dabei steht "*?species*" für eine Variable. Die Anfrage-Engine liefert dann den Wert *Dolphin* zurück. Als Ergebnis eines einfachen Reasoning (siehe Kapitel 2.4.2) entsteht das Tripel *Flipper isA Mammal*. In diesem Fall wird auch das Wert *Mammal* als Ergebnis der o.g. Abfrage zurückgegeben. Durch die Bereitstellung mehrere Mustern können komplexe Abfragen erstellt und verwendet werden.

In den vorherigen Abschnitten dieser Arbeit wurden solche Konzepte wie Metadaten und Ontologien skizziert, die entsprechend das Quellformat und Zielformat des Mappings darstellen. Im nächsten Abschnitt wird das Mapping selbst detaillierter behandelt, u.a.

<http://www.zeppelin-university.de/deutsch/lehrstuehle/ticc/TICC-101203-OpenGovernmentData-V1.pdf> (10. 06 2011).

[86] Siehe Cyganiak R. & Jentzsch A.: The Linking Open – Data cloud diagram, v. 22.09.2010, <http://richard.cyganiak.de/2007/10/lod/> (13.07.2011).

[87] Siehe GeoNames: <http://www.geonames.org/> (10.06.2011).

[88] Programmierschnittstelle für den Programmierer, auf der bestimmte interne Funktionsabläufe abstrahiert werden. Siehe <http://www.itwissen.info/definition/lexikon/application-programming-interface-API-Programmierschnittstelle.html> (13.07.2011).

[89] Siehe Semantic Web Company: RDF Triple Store: <http://www.semantic-web.at/1.32.catchword.243.rdf-triple-store.htm> (13.07.2011).

[90] Weiterführende Informationen über SPARQL unter: SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/> (13.07.2011).

werden seine Ziele und Methoden untersucht. Anschließend werden einige Probleme eingeführt, die sich im Verlaufe von Mapping-Verfahren herauskristallisiert haben.

2.7 Metadatenmapping

2.7.1 Zwecke, Erwartungen und Methodik

Als Endergebnis des Mappings, mit dem diese Arbeit sich beschäftigt, wird ein Datenmodell erwartet, das sich aus unterschiedlichen Mapping-Transformationen modular zusammensetzt.[91] Die einzelnen Mapping-Schritte werden im Kapitel IV skizziert. Genauer betrachtet wird jedoch der Algorithmus zur Transformation von einfachen Metadatenformaten EAD, DC und LIDO ins CIDOC CRM. Die Abbildung 9 stellt das von den Entwicklern des CRM definierte Mapping-Schema dar.[92]

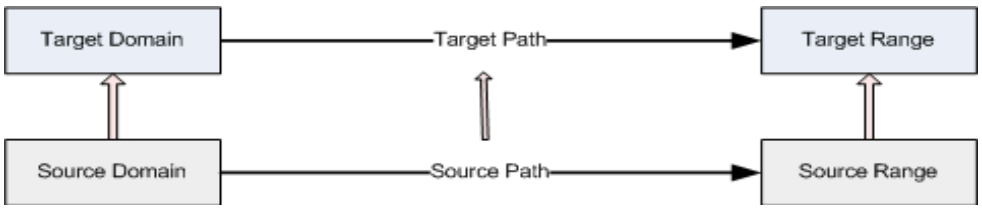


Abbildung 9: Das Basismapping-Schema.

Dementsprechend wird vom Mapping erwartet, dass alle Instanzen und Relationen aus dem Quellformat ohne Datenverluste in semantisch äquivalente Instanzen und Relationen des Zielformats transformiert werden. Die Struktur der Quellmetadaten besteht aus Elementen, ihren Attributen und Werten. Diese Ausgangselemente sollen im CRM in entsprechende Klassen und Properties semantisch interpretiert werden.

Als Mapping-Sprache wird XSLT (*Extensible Stylesheet Language Transformation)* Version 2.0 eingesetzt.[93] XSLT ist eine leistungsfähige funktionale Sprache zur Transformation von XML. Eine XSLT-Transformation wird in Form eines Stylesheets ausgedrückt, dessen Syntax einer wohlgeformten XML-Datei entspricht. In den Stylesheets werden die Umwandlungsregeln definiert, nach welchen die Elemente des

[91] Siehe das DDB-Datenmodell im Kapitel 4.1.

[92] Siehe Doerr, M., Kondylakis, H., Plexousakis, D.: Mapping Language for Information Integration, Technical Report 385, December 2006.

[93] Zur Syntax und Semantik von XSLT siehe auf: XSL Transformations (XSLT) Version 2.0 W3C Recommendation 23 January 2007, <http://www.w3.org/TR/2007/REC-xslt20-20070123/> (13.07.2011).

Quelldokumentes in gewünschte Elemente des Zieldokumentes transformiert werden. Es wird ein neues Dokument erstellt und das Original bleibt unverändert. XSLT operiert mit XML Path Language (XPath) auf der logischen Baumstruktur eines XML-Dokumentes. Im Kontext von XSLT wird XPath dazu verwendet, um Elemente durch die Pfadeingaben zu adressieren, außerdem Zahlberechnungen durchzuführen und Zeichenketten zu manipulieren.[94]

Der pfadorientierte Ansatz (*path-oriented approach*) ist besonders für das betreffende Mapping geeignet, da sowohl die Quellmetadaten als auch das Zielformat CIDOC CRM einer XML-Notation unterliegen, deren Datenwerte über Pfadausdrücke abgefragt werden können. Ein weiterer Vorteil des pfadorientierten Ansatzes besteht darin, dass die Elemente in den Quellmetadaten, je nach ihrer Stellung im Dokument eine unterschiedliche Semantik aufweisen können.[95] In Abbildung 4 wird dieses Phänomen an dem EAD-Element <unittitle> veranschaulicht.

Der Mapping-Ansatz in dieser Arbeit stellt konzeptuelle Mappings von gängigsten Typen von Objekten im Kulturbereich dar. Diese konzeptuellen Mappings, auch Cluster genannt, werden durch parametrisierte Templates realisiert, die aufgerufen werden, wenn eine Bedingung erfüllt ist. Der Mapping-Ansatz wird im Kapitel 4.3 erneut aufgegriffen. Im nächsten Abschnitt werden einige Mapping-Probleme skizziert und mit Beispielen erläutert.

2.7.2 Besondere Herausforderungen an Mappings heterogener Metadaten auf das CRM

Unterschiede in der Erschließung der gleichen Elemente: Je nach Typ der Ressource können dieselben Metadatenelemente unterschiedliche Bedeutungen aufzeigen. Die Vielfalt unterschiedlicher Bedeutungen führt beim Mapping zu einem wichtigen Problem. Ein gutes Beispiel zur Veranschaulichung des Problems bietet das Mapping von Dublin Core Elementen.[96] Das Dublin Core Element <dc:subject> umfasst u.a. folgende

[94] Siehe Kay, M.: *XSLT 2nd Edition: programmer's reference*, Canada, Wrox Press, 2004, S. 25-26.

[95] Siehe Doerr, M., Stasinopoulou, T., Bountouri, L., Kakali, C., Lourdi, I., Papatheodorou C. & Gergatsoulis, M.: Ontology-Based Metadata Integration in the Cultural Heritage Domain, S. 169.

[96] Siehe Beispiele in: Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: DC.type mapping to CIDOC/CRM. 26/01/2007.

Bedeutungen: "the topic of the resource" oder "the field of knowledge to which the resource belongs".[97] Wenn die Ressource vom Typ *Image*[98] ist, stellt das Element <dc:subject> den "Gegenstand des Bildes" dar. Wenn der Typ der Ressource dagegen auf *Text*[99] gesetzt ist, deutet das besagte Element auf das "Thema des Textes". Dieser Unterschied in der Bedeutung wirkt sich unter Umständen auch auf das Mapping aus, wie im folgenden Beispiel demonstriert wird:

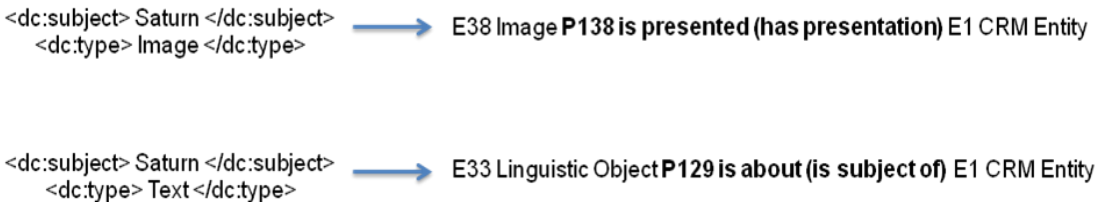


Abbildung 6: Das Mapping vom Dublin Core Element <dc:subject> ins CRM unter Berücksichtigung des Ressourcentyps.

Demzufolge hängt das Mapping einer Ressource eng mit dem Typ der Ressource zusammen. Die Schwierigkeiten, die mit der Feststellung des Ressourcentyps zusammenhängen, werden im Kapitel 4.4 zur Sprache gebracht.

Unterschiede zwischen verschiedenen Standards und Institutionen: Quellmetadatenstandards können, abhängig von den Institutionen, die sie verwalten, voneinander abweichen. Für die Prozesse des Mappings bedeutet das ein nächstes Problem, das aufgegriffen werden muss. Die Abweichungen in Quellmetadatenstandarte betreffen vor allem die Pfadabfragen. Das Element EAD Element <origination> enthält z.B. die Angaben über die Person oder die Organisation, die für die Erstellung der erschlossenen Materialien verantwortlich sind.[100] Der Name der Organisation oder Person kann in unterschiedlichen Kinderelementen vom Element <origination> vorkommen: <persname> oder <famname> oder <corpname>. Die Pfade werden auch entsprechend

[97] Siehe die Beschreibung der Dublin Core Elemente unter: Dublin Core Metadata Element Set, <http://dublincore.org/documents/dces/> (13.07.2011).

[98] "A visual representation other than text", siehe ebd.

[99] "A resource consisting primarily of words for reading", siehe ebd.

[100] Siehe den Eintrag Provenienz (Origination) in: Encoded Archival Description Tag-Library. Version 2002 herausgegeben und erarbeitet von der Arbeitsgruppe Encoded Archival Description der Society of American Archivists und des Büros für Netzwerkentwicklung und MARC Standards der Library of Congress Chicago 2003, übersetzt von Anke Löbnitz und Angelika Menne-Haritz. Berlin 2006. S. 186

verschieden ausfallen: *origination*/*persname* oder *origination*/*famname* oder *origination*/*corpname.*

Eine aus der Perspektive des DDB-Projekts denkbare Lösung wäre die Einigung der verschiedenen Institutionen auf eine verbindliche Norm und die Angabe aller vorhandenen Abweichungen bei der Datenlieferung. In diesem Sinne ist auch ein Vorschlag des Landesarchivs Baden-Württemberg vom 3. Dezember 2010 entwickelt worden, der einen Überblick über die wichtigen EAD-Elemente und Angaben zu den Feldinhalten gibt. Die Abstimmung mit anderen Archiven steht jedoch aus. Summa summarum fällt es in den Bereich des Mappings, die Unterschiede zwischen verschieden Metadatenversionen und Standardabweichungen festzustellen und entsprechend zu behandeln. Unter Umständen sind mehrere Mappings für das gleiche Quellformat erforderlich.

Datenqualität: Zeichenketten, die in den Metadatenfeldern vorkommen, können von unterschiedlicher Qualität sein und von abweichender Konsequenz in der Feldnutzung zeugen. Das folgende Beispiel verdeutlicht die Problematik, die daraus für das Mapping entsteht: Das EAD Feld <physloc> wird laut EAD-Tag Library folgendermaßen definiert:

> PHYSLOC: Informationen über den Ort, an dem die erschlossenen Materialien gelagert werden, wie z.B. der Name oder die Nummer des Gebäudes, Raumes, Magazins, Regals oder einer anderen konkreten Aufbewahrungsstelle.[101]

Das Element *physloc* entspricht demzufolge dem Aufbewahrungsort der Ressource und wird entsprechend ins CRM auf eine Instanz der Klasse E52 Place gemappt:

> E22 Man-Made Object ***P53 has former or current location*** E53 Place.

Oft wird jedoch das Feld mit zusätzlichen Informationen gefüllt, wie z.B. die Laufzeit der Ressource:

> <physloc> Standort: K 1.Mag.2.001.2337.06 Laufmeter am Lagerort: 0,4 Anzahl und Art Sonstige: 89 Postkarten, 1 Buch Vermerke: Laufzeit: 1915 – 1918. </physloc>

In diesem Fall passt das oben vorgeschlagene Mapping nicht mehr auf die Klasse *E52 Place* und muss umgedacht werden. Das Problem bleibt weiter bestehen, weil eine

[101] Siehe den Eintrag Lagerort (Physical Location) in: Encoded Archival Description Tag-Library. Version 2002 herausgegeben und erarbeitet von der Arbeitsgruppe Encoded Archival Description der Society of American Archivists und des Büros für Netzwerkentwicklung und MARC Standards der Library of Congress Chicago 2003 übersetzt von Anke Löbnitz und Angelika Menne-Haritz Berlin 2006. S. 196.

Normalisierung von solchen Zeichenketten vor dem Mapping nicht vorgenommen wird. Die Suche nach solchen "deplatzierten" Informationsinhalten wie Laufzeit der Ressource oder Anzahl und Art der Items, aus denen sich die Ressource zusammen setzt, wird vermutlich keinen Treffer erzielen, da nach solchen Informationen nicht in den Aufbewahrungsortangaben gesucht wird. Deshalb wird in solchen Fällen das Mapping an sich abstrakter gehalten und die Normalisierung auf den Zeitraum nach der Mapping-Phase verschoben. Das Feld *physloc* findet folglich seine Entsprechung nicht in der Klasse *E52 Place*, sondern in der nicht typisierten Klasse *E1 CRM Entity*:

E22 Man-Made Object ***P3 has note*** E1 CRM Entity

Das Problem ist teilweise nach dem Mapping lösbar, und zwar mit Hilfe von kontrollierten Vokabularen, von denen die Rede im Kapitel 4.2.3 sein wird. Eine weniger realistische Variante wäre, dass die Provider bei der Lieferung der Metadaten Hinweise auf die denormalisierte Feldernutzung geben.

Zugehörigkeit zu den verschiedenen Klassen: Mit der Frage nach der Klassenzugehörigkeit der Objekte im CRM geraten wir an ein weiteres grundlegendes Problem des Mappings. Da die Mehrfachvererbung im CRM möglich ist, können die Objekte Zugehörigkeit nicht nur zu einer Klasse, sondern auch zu mehreren Klassen gleichzeitig aufweisen. Das Wurzelelement <ead> entspricht z.B. sowohl der Klasse *E33 Linguistic Object* als auch der Klasse *E31 Document*.[102] Dies soll bedeuten, dass ein EAD-Dokument ein Sprachliches Objekt ist, das etwas Bestimmtes dokumentiert und erbt folglich die Eigenschaften der beiden Klassen (Abbildung 10).

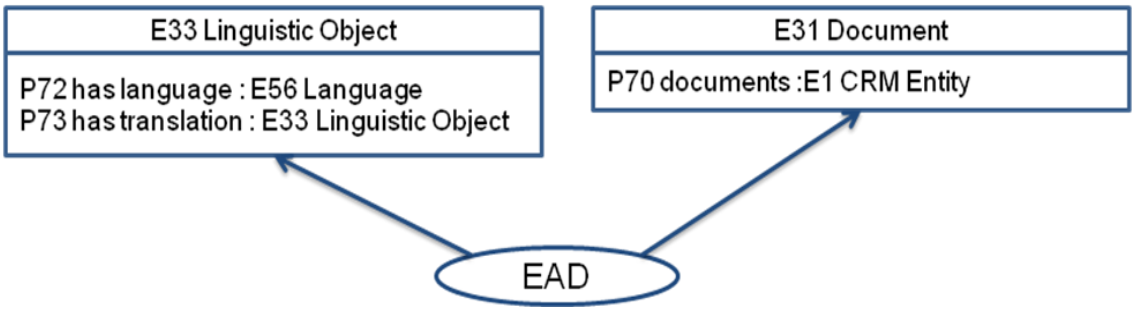


Abbildung 10: Die Zuweisung der CRM-Klassen dem Element <ead>.

[102] Siehe Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: WP5-Task 5.5. EAD mapping to CIDOC/CRM, Version 0.2, 02.03.2007, S. 7.

Diese Doppelzuweisung kann in RDF Tripels, die die Basiskomponente des Mapping-Ergebnisses darstellen, nicht ohne weiteres realisiert werden, da als Domain nur eine Ausgangsklasse fungieren kann. Die Lösung dafür wird im Kapitel 4.3.2 am Beispielmapping vorgestellt.

3 Das CIDOC Conceptual Reference Model (CRM)

3.1 Ursprung und Zielsetzung

Das CIDOC CRM ist das Resultat einer mehr als zehnjährigen Zusammenarbeit des Internationalen Ausschusses für Dokumentation (kurz CIDOC) und des Internationalen Museumsrates ICOM[103] im Bereich der Standardisierung kultureller Informationen. Das CRM stellt eine formale Ontologie im Sinne der Informatik dar und gilt seit 2006 als Norm für den kontrollierten Austausch von Informationen im Bereich des kulturellen Erbes.[104] Das Modell überzeugt mit seinem weit umfassenden ontologiebasierten und objektorientierten Datenmodel, das auf der Grundlage einer Klassen- und Properties-orientierten Syntax semantische Beziehungen aufbaut, die für den kulturellen Bereich relevant sind. Damit wird eine wichtige Voraussetzung für die Informationsintegration erfüllt. Das CRM spielt dabei die Rolle des semantischen Kleisters (*semantic glue)*, welcher dazu dient, den Datenaustausch heterogener Daten zu ermöglichen und ihre Integration in eine zusammenhängende institutionsübergreifende Informationsressource möglichst verlustfrei zu verwirklichen.[105]

Das CRM versteht sich nicht als Dokumentationsleitfaden für Kulturinstitutionen, sondern interpretiert das, was von den Kulturinstitutionen derzeit dokumentiert wird, und schafft dadurch die Bedingungen für semantische Interoperabilität. Unter semantischer Interoperabilität versteht man die "Fähigkeit unterschiedlicher Informationssysteme zur Kommunikation von Information folgerichtig zu ihrer beabsichtigten Bedeutung."[106] Die beabsichtigte Bedeutung bezieht sich dabei auf die Datenstruktur der involvierten Elemente, die benutzte Terminologie und die Identifikation der Objekte. Ausschließlich solche Informationen, die die Ressourcen und ihren historischen, geographischen und theoretischen Background betreffen, fallen in den Beschreibungsbereich des CRMs.

---

[103] ICOM (*International Council of Museums* auf Deutsch Internationaler Museumsrat) ist eine internationale Organisation für Museen, die 1946 in Zusammenarbeit mit der UNESCO ins Leben gerufen wurde. Ihr Ziel umfasst die Interessen von Museen weltweit. Siehe dazu: <http://www.chin.gc.ca/Applications_URL/icom/#> (13.07.2011).

[104] Zu den weiterführenden Informationen über das CRM, seinen Zwecken und Anwendungsgebieten siehe die Homepage von CIDOC CRM unter: <http://www.cidoc-crm.org/> (19.07.2011).

[105] Siehe Doerr, M., Hunter, J. & Lagoze, C.: Towards a Core Ontology for Information Integration, S. 4.

[106] Doerr M., Lampe, K. & Krause S. (hrsg. & übers.): Definition des CIDOC Conceptual Reference Model, Version 5.0.1, ICOM Deutschland, 2010, S. 9.

Dagegen werden Informationen, die Administration und Management von Kulturinstitutionen betreffen, im CRM nicht behandelt.[107]

Außer der gezielten Suche, einem einheitlichen Zugriff und Datenaustausch zwischen heterogenen kulturellen Quellen bietet CRM Techniken zur Lösung des Problems der Heterogenität, das im Abschnitt 3.3.2 behandelt wird.

3.2 Klassen und Properties

Das CRM Version 5.0.2 besteht aus 86 Klassen und 137 Properties.[108] Die Klassen und Properties stellen einen Satz von Begriffen dar, der die wichtigsten Konzepte und Beziehungen des Kulturbereichs abdeckt. Die CRM-Klassen werden durch die allen Objekten dieser Klassen gemeinsamen Properties definiert. Der Aufbau des CRM Models entspricht einer Klassenhierarchie, deren oberste Klasse *E1 CRM Entity* heißt und die aus Eltern- und Kinderklassen besteht.[109] Im CRM Model wird die Mehrfachvererbung zugelassen, worauf schon im Kapitel 2.7.2 kurz eingegangen wurde. Eine Kinderklasse kann die Spezialisierung von mehr als nur einer Elternklasse sein und dementsprechend die Properties von allen ihren Elternklassen erben.

Die CRM Klassen lassen sich in zwei essentielle Bereiche aufteilen: die Klassen der materiellen Objekte und die Klassen der immateriellen Objekte. Im Anhang A wird die Klassenhierarchie abgebildet, die aus den wichtigsten Klassen der beiden Bereiche besteht. So gehört z.B. die Klasse *E33 Linguistic Object* in den Bereich des Immateriellen und umfasst:

> "…identifiable expressions in natural language or languages. Instances of E33 Linguistic Object can be expressed in many ways: e.g. as written texts, recorded speech or sign language."[110]

---

[107] Siehe Doerr M., Crofts, N., Gill, T., Stead S. & Stiff M.: Definition of the CIDOC Conceptual Reference Model, S. ii.

[108] In der Fachliteratur kursieren unterschiedliche Notationen für Elemente des CRM. Klassen werden auch als Einheiten oder Entitäten bezeichnet; Properties wiederum als Eigenschaften, Relationen und Beziehungen. Hier wird sich einheitlich auf "Klassen" und "Properties" beschränkt.

[109] Ausgenommen von der Klasse *E59 Primitive Value* und ihren Subklassen, welche nicht von der Klasse *E1 CRM Entity* abgeleitet werden, siehe *CIDOC CRM Class Hierarchy* in: Doerr M., Crofts, N., Gill, T., Stead S. & Stiff M.: Definition of the CIDOC Conceptual Reference Model, Version 5.0.2, Januar 2010. S. xxii-xxiv.

[110] Ebd., S. 15.

Dabei ist zu beachten, dass Instanzen der Klasse E33 im CRM unabhängig von dem Medium oder der Methode behandelt werden, mit der sie ausgedrückt wurden.[111] Das bedeutet, dass für die Beschreibung eines Buches die Properties der Klasse E33 nicht ausreichen werden, da sie nur den Text des Buches betreffen und nicht das Buch als ein materielles Objekt beschreiben. So können die Informationen, die den Standort der Ressource betreffen, nicht an die Instanz der Klasse *E33*, sondern nur an eine Instanz eines materiellen Sachverhalts geknüpft werden. Da im CRM mehrfache Vererbung zugelassen ist, erscheint es an dieser Stelle sinnvoll, ein solches Objekt wie ein Buch als eine Instanz sowohl der Klasse *E33* als auch der Klasse *E22 Man-Made Object* zu definieren. Somit können für die Beschreibung eines Buches die Properties der beiden Klassen verwendet werden.

Properties stellen Relationen zwischen den Klassen dar. Um das Retrieval effizienter zu machen, werden die Relationen bei den Suchanfragen berücksichtigt. In der Einleitung wurde schon das Problem der klassischen Suchmaschinen angesprochen, die eine enorme Vielfalt von Online-Angeboten zurückgeben. Dies soll anhand des folgenden Beispiels der Deutschen Museumsbunds Gruppe illustriert werden:

> "Sucht man z.B. in einem großen Datenbestand mit den üblichen Techniken der Suchmaschinen nach Informationen, in denen die Worte "Violine" und "Stradivari" vorkommen, wird man eine unpraktikabel große Zahl von "Treffern" bekommen, die infolge ihrer großen Zahl kaum hilfreich sein dürften, Violinen zu finden, die von Antonio Stradivari in einer bestimmten Zeitperiode gebaut wurden. Es wäre vielmehr nötig, die qualifizierte Beziehung zwischen dem Instrument und seinem Erbauer bei der Formulierung der Suchanfrage berücksichtigen zu können."[112]

Diese Beziehung zwischen dem Instrument und seinem Schöpfer wird im CRM mit einem Event (Ereignis) ausgedrückt. Events spielen im CRM eine wichtige Rolle und werden im Abschnitt 3.3 näher erklärt.

---

111 Siehe ebd.

112 Deutscher Museumsbund: Das CIDOC Conceptual Reference Model: Eine Hilfe für den Datenaustausch? <http://www.cidoc-crm.org/docs/cidoc_paper_german.pdf> (23.07.2011).

3.2 CRM Paradigma

In diesem Abschnitt wird der Aufbau von Beziehungen im CRM erläutert.[113] Ferner könnte das CRM Modell als folgendes Paradigma dargestellt werden:

Dieses Paradigma folgt der Grammatik einer Subjekt-Prädikat-Objekt Beziehung. Eine CRM-Klasse kann die Rolle eines Subjekts oder auch die eines Objekts annehmen. Dementsprechend kann jede CRM Klasse die Ausgangsklasse oder Zielklasse von keiner, einer oder mehreren definierten Properties sein. Dabei ist es zu beachten, dass eine CRM Klasse ohne Properties nicht existiert. Auch wenn es Klassen gibt, denen formal keine Properties zugeschrieben werden, verfügen diese Klassen über Properties, die sie von ihren Oberklassen erben. CRM Properties drücken die Relationen zwischen zwei CRM Klassen aus und spielen somit die Rolle des Prädikats. Eine Reversion der Ordnung, sozusagen eine passivische Verwendung der Properties, wird vom Modell ebenfalls unterstützt.

Den Klassen sind Properties zugewiesen und diese können von den Klassen fakultativ verwendet werden. Die Properties verfügen dagegen notwendigerweise über eine bestimmte Ausgangs- und eine bestimmte Zielklasse. Daraus ergibt sich die Konsequenz, dass eine Property weder alleine stehen noch von einer einzelnen Instanz regiert werden kann. Im darauffolgenden Beispiel wird die formale Beschreibung einer Property gezeigt, welche in der Klasse *E31 Document* definiert ist:

Da E1 CRM Entity die Mutterklasse aller CRM-Klassen ist, kann als Zielklasse dieser Relation jedes Objekt in Frage kommen. Wie es oben angesprochen wurde, können Domain und Range in ihrer Abfolge ausgetauscht werden, was folgendes ergeben würde:

---

113 Basiert im Wesentlichen auf verschiedenen Dokumenten von der CIDOC CRM Homepage: The CIDOC Conceptual Reference Model, <http://www.cidoc-crm.org/> (23.07.2011).

Aufgrund seiner tripelbasierten Struktur, lässt sich das CRM problemlos in das RDF einbetten.

3.3 Events

Das CRM Modell ist als event-zentriert (*event aware*) konzipiert und wird von seinen Entwicklern als Modell möglicher Zustände angesehen (siehe Kapitel 2.4.1):

> "The following ideas are an interpretation and extension of the CIDOC CRM, a model of possible states of affairs in the real world, in which historical and archaeological phenomena are abstracted as a network of persistent items that meet in space and time." [114]

Mit der Klasse *E4 Period* und ihren Unterklassen wird die Idee von "*coherent phenomena or cultural manifestations bounded in time and space*"[115] widergespiegelt. Die Kinderklasse *E5 Event* ist kompatibel mit der Klasse *E4* und umfasst dazu die Zustandsänderungen (*changes of state*), die durch Begegnungen von lebendigen und nicht lebendigen Objekten verursacht werden.[116] Über einen Event werden verschiedene Konzepte verknüpft. So wird bspw. ein Artefakt (*E22 Man-Made Object*) über einen Herstellungsevent (*E12 Production*) mit seinem Urheber (*E39 Actor*), seinem Erstellungsdatum (*E52 Time-Span*) und seinem Erstellungsort (*E55 Place*) verbunden:

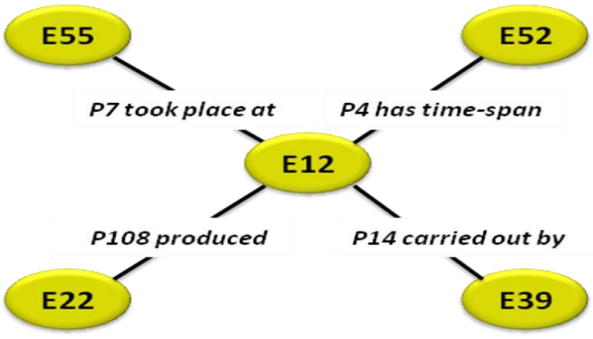


Abbildung 11: Herstellungsevent

[114] Doerr M., Crofts, N., Gill, T., Stead S. & Stiff M.: Definition of the CIDOC Conceptual Reference Model, S.3.

[115] Ebd., S. 3.

[116] Siehe *Events as meetings* in: Doerr, M., Plexousakis, D., Kopaka, K. & Bekiari, C.: Supporting Chronological Reasoning in Archaeology, v. 2004, unter: <http://www.cidoc-crm.org/docs/caa2004_supporting_chronological_reasoning.pdf> (13.07.2011).

Die Klasse *E12 Production* wird von der Klasse *E5 Event* abgeleitet. In der *E5* Klasse und ihren Unterklassen werden durch mannigfaltige Beziehungen Informationen über Ereignisse strukturiert. Sie verweisen stets auf einen *E39 Actor*, der in irgendeiner Rolle an dem entsprechenden Ereignis teilgenommen hat. Auch die Lebensdaten einer Person werden als Beziehungen zu den Ereignissen *E67 Birth* und *E69 Death* (Unterklassen der Klasse *E5 Event*) definiert:

E67 Birth *P98 brought into life (was born):*E39 Actor
E69 Death *P100 was death of (died in):* E39 Actor

3.3.1 Das Reasoning im CRM

Events bieten eine gute Voraussetzung zum *zeitlichen* und *räumlichen* Reasoning im CRM. Es lässt sich bspw. schlussfolgern, dass nicht-temporale Entitäten, die an einem Ereignis teilgenommen haben, in dem Zeitraum, in dem das Ereignis stattgefunden hat, existierten und sich an dem Ereignisort befunden haben.[117] Auch fehlende Lebensdaten einer Person können unter Umständen durch das Reasoning eingeschätzt werden. Dies soll am Beispiel von der legendären Begegnung von Papst Leo des Großen mit Attila, König der Hunnen, in Mantua erläutert werden (Abbildung 12).

Nach Ansicht der CRM Entwickler, ist das Wissen, dass durch Reasoning erlangt wurde, häufig zuverlässiger als die expliziten historischen Angaben.[118] Die Abbildung 12 stellt fünf Zeitangaben dar, zwei davon sind unbekannt. Mit Hilfe des zeitlichen Reasonings lassen sich alle Zeitangaben dank gegenseitiger Einschränkung in eine Zeitspanne zu zuordnen. Die beiden Todesdaten können logischerweise zeitlich erst nach dem Begegnungsdatum und nach den beiden Geburtsdaten folgen. Darüber hinaus schränkt das Begegnungsdatum beide Todes- und beide Geburtsdaten ein, indem eine maximale Lebensdauer angenommen wird.

[117] Siehe den Abschnitt *Participation and Spatiotemporal Reasoning* in: Doerr, M.: The CIDOC CRM – an Ontological Approach to Semantic Interoperability of Metadata, S. 11.
[118] Siehe ebd., S. 11-12.

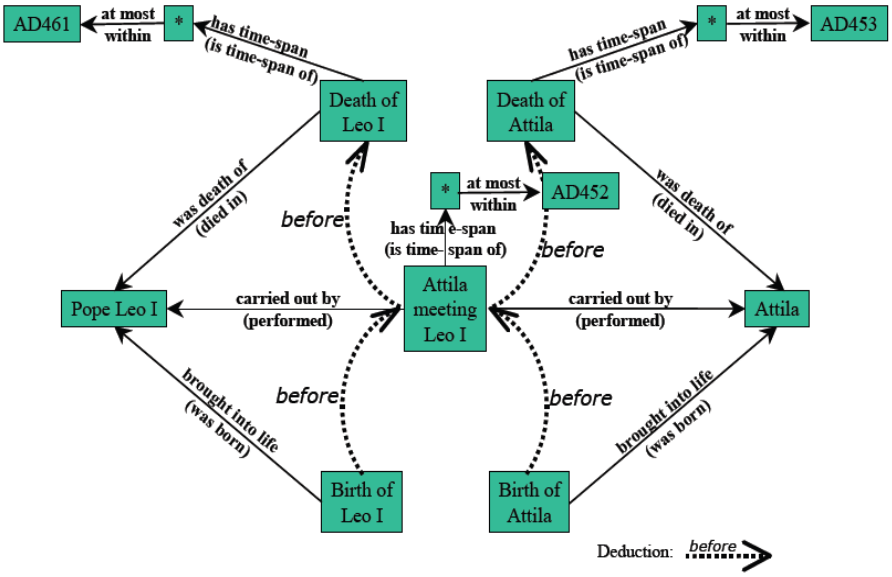


Abbildung 12: Die Begegnung zwischen Attila und Leo I.[119]

Dabei ist zu beachten, dass das CRM keine Aussagen über Zeitpunkte trifft, sondern nur über Zeitspannen/Zeiträume. Außerdem ist das Reasoning kein Bestandteil des Modells.[120]

3.3.2 Lösungsansatz zur Heterogenitätsproblematik

Events ermöglichen es, Objekte besser zu identifizieren und tragen u.a. zur Lösung des Problems der Duplikate bei. Unter Duplikaten hier versteht man "Objekte, die in identischer Form bereits vorhanden sind".[121] Ungeachtet davon, dass die Duplikate bei verschiedenen Institutionen untergebracht sind (was bei Büchern öfters der Fall ist) und außerdem in unterschiedlichen Metadatenformaten erfasst wurden, besteht jedoch die Möglichkeit, ihre Identität anhand der erschlossenen Events festzustellen.[122] So müssen bspw. bei zwei identischen Objekten auch ihre Erstellungsevents übereinstimmen: unter Umständen die Namen der Personen, die an der Erstellung der Objektes teilgenommen haben, der Erstellungsort und das Erstellungsdatum.

Die Schwierigkeit dabei besteht darin, dass die aus den Quellmetadaten erschlossenen Informationen nicht immer vollständig sind. Ein noch größeres Problem besteht darin, dass

---

119 Die Abbildung stammt aus ebd.

120 Siehe CRM-Darstellung des zeitlichen und räumlichen Reasoning in: Doerr M., Crofts, N., Gill, T., Stead S. & Stiff M.: Definition of the CIDOC Conceptual Reference Model, Version 5.0.2, Januar 2010, *fig*. 2 und *fig*. 3.

121 Stalmann K. & Budde R.: Projekt Deutsche Digitale Bibliothek (DDB): Grobkonzept für das Portal der Deutschen Digitalen Bibliothek, Version 2.0, v. 12.08.2010, S. 35.

122 Zum Problem des Duplikathandlings siehe ebd., S. 35-36.

diese Informationen nicht immer einer normierten Form entsprechen. Beim Vergleich zweier Zeitangaben, die in unterschiedlichen Formen vorkommen, muss man sie zuerst in eine Form überführen, sprich normalisieren. Das Problem der nicht normierten Zeichenketten wurde bereits im Kapitel 2.7.2 dieser Arbeit besprochen. Dieses Problem wird teilweise durch Verwendung der kontrollierten Vokabulare – insbesondere der sog. Normdateien – gelöst (siehe Kapitel 4.2.3).

Events bilden einen wichtigen Teil des Mappings. Es wäre daher anstrebenswert, dass jede Ressource vor allem über einen Erstellungsevent verfügt. Deswegen ist es wichtig, beim Mapping nach entsprechenden Informationen sorgfältig zu suchen. Da nicht alle Ausgangsformate eventbasiert sind, greift man beim Mapping auf die Erzeugung der sogenannten Pseudo-Elemente zu (siehe Kapitel 4.2.4).[123]

3.3.3 Rich Semantics

Ein weiterer Vorteil, den das Konzept des Events mit sich bringt, liegt in der explizit ausgedrückten Semantik (*rich semantics*), die bei den Quellmetadaten des Öfteren fehlt.[124] Die unterschiedlichen Felder <dc:creator> (DC), <origination> (EAD), <lido:eventActor> (Lido) verweisen bspw. alle auf den "Urheber der Ressource" hin.[125] Die impliziten Zusammenhänge, wie z.B. der Typ der Ressource, die Tatsache, dass die Ressource durch einen Erzeugungsprozess ins Leben gerufen wurde, der von einer Gruppe von Menschen, einer Person oder einer Institution hervorgerufen wurde, sind nur für die Menschen nachvollziehbar.

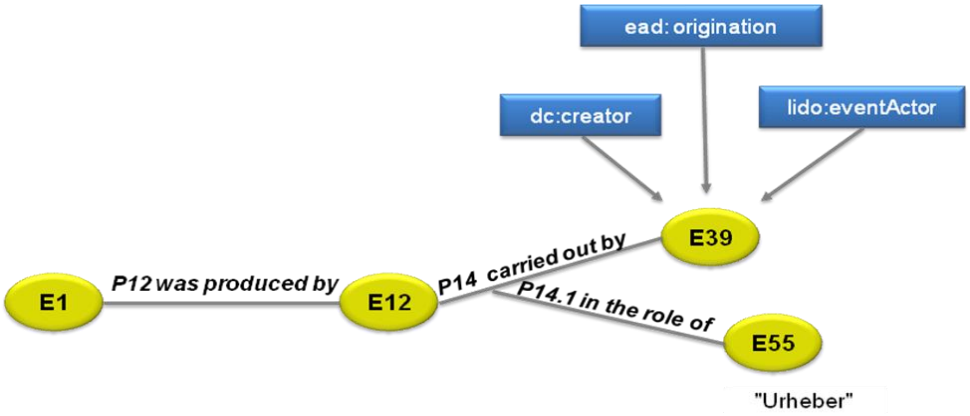


Abbildung 13: Das Mapping des Urhebers einer Ressource ins CRM

123 Mit der Ausnahme vom LIDO (siehe Kapitel 2.2.3).

124 Siehe Doerr, M., Stasinopoulou, T., Bountouri, L., Kakali, C., Lourdi, I., Papatheodorou C. & Gergatsoulis, M.: Ontology-Based Metadata Integration in the Cultural Heritage Domain, S. 167.

125 Allerdings weist bei LIDO, wie es schon im Anschnit 2.2.4 gesagt worden ist, das Element <lido:eventActor> auf einen unbestimmten Akteur hin, und wird nachfolgend typisiert.

Im CRM dagegen werden diese Informationen explizit in Klassen und Properties ausgeprägt und werden dadurch auch für Maschinen umwertbar. Somit werden diese Instanzen eines Urhebers im CRM zu Instanzen der Klasse *E39 Actor*, die an einem Erzeugungsevent (*E12 Production*) in der Rolle eines *Urhebers* teilgenommen haben (siehe Abbildung 13).

4 Das DDB-Datenmodell und die praktische Umsetzung der Metadatenmappings für DC, EAD und LIDO

4.1 DDB-Datenmodell

Das DDB-Datenmodell setzt sich aus mehreren Mapping-Bausteinen zusammen (Abbildung 14). Im Folgenden werden die einzelnen Schritte kurz erläutert.

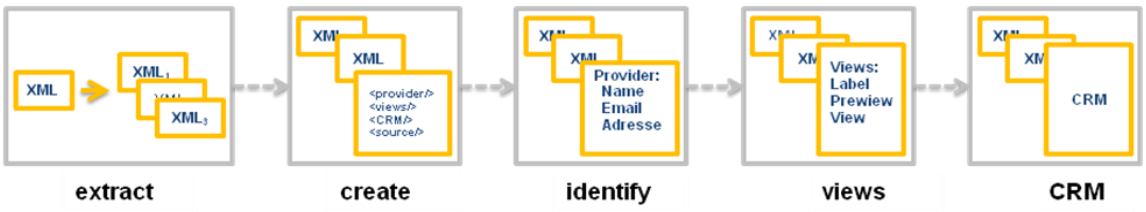


Abbildung 14: Abbildung der einzelnen Transformationsschritte.

extract: Zu Beginn des Mappings werden Dateien, die Metadaten zu mehreren Ressourcen enthalten, aufgesplittet, sodass jede der Teilung entsprungene Datei jeweils nur Metadaten zu einer einzigen Ressource enthält. Dadurch wird die Vergabe der lokalen Identifier bei dem CRM-Schritt (siehe unten) wesentlich erleichtert.

create: In dem "create" Schritt des Mappings wird das noch informationsleere DDB-Datenmodell angelegt, das künftig mit folgenden Informationen gefüllt wird: die Provider-Informationen (das Ergebnis des identify-Schritts), das Label, die Preview und View der Ressource (das Ergebnis des views-Schritt), eine Kopie der originalen Metadaten und das Ergebnis der Transformation der Quellmetadaten ins CRM (CRM-Schritt).

identify: In dem "identify" Schritt werden die Informationen über den Provider der Ressource gesammelt. Der wichtigste Eintrag hierfür ist der Provider-Identifier der primären Ressource.[126] Dieser Eintrag ist unabdingbar für die Generierung des DDB-Identifiers der Ressource (siehe dazu Kapitel 4.2.2).

views: Man unterscheidet zwischen drei Arten von Views: die Description, das Preview und das View. Die *Description* stellt eine knappe, literale Beschreibung einer Ressource

[126] Genaueres zu den Ressourcen im Rahmen des DDB-Projekts entnehme man dem Kapitel 4.2.1.

dar und eignet sich als Label. Das *Preview* ist eine kurze Beschreibung der Ressource. Es beinhaltet u.a. den Titel, den Autor und das Erzeugungsjahr der Ressource und wird in der Suchergebnisliste angezeigt. Das *View* stellt die Repräsentation der Ressource in der Einzelsicht dar.

CRM: Dieser Teil beinhaltet die Skripte zur Transformation von Quellmetadaten ins CIDOC CRM. Jedem Quellmetadatenformat ist mindestens ein Skript zugeteilt. Diese verschiedenen Skripte benutzen die gleichen konzeptuell vorgefertigten Cluster, dessen Konzept im Kapitel 4.3 an Beispielen genauer erläutert wird. Das Ergebnis dieser Transformationen bildet den wesentlichen Teil des DDB-Datenmodells und stellt folglich einen Metadatensatz dar, der aus RDF-Tripeln mit CRM-Syntax besteht. Im folgenden Abschnitt wird die Struktur der im Schritt CRM entstandenen Metadaten im Detail dargelegt, darüber hinaus werden die wichtigsten Mapping-Konzepte skizziert.

4.2 DDB-Metadaten und wichtige Mapping-Konzepte

4.2.1 Einfache und primäre Ressourcen

Grundsätzlich wird im DDB-Projekt zwischen den primären Ressourcen und den einfachen Ressourcen differenziert. Die primären Ressourcen bilden die Gruppe der Informationsobjekte, die beschrieben werden: Bücher, Videos, Bilder, etc. Mit dem Begriff der einfachen Ressourcen ist eine Gruppe von Ressourcen gemeint, die zur Beschreibung der primären Ressourcen dienen, bspw. solche Eigenschaften wie Titel, Aufbewahrungsort, Format, Erstellungsdatum und weitere. Die Transitionen der einfachen Ressourcen werden mittels ihrer Relationen zu der ihnen entsprechenden primären Ressource realisiert. Für jede einfache Ressource werden zwei RDF Tripel erzeugt. Im ersten Tripel wird die primäre Ressource mit der einfachen Ressource mittels einer CRM-Property verbunden. Im zweiten Tripel findet die Typisierung der einfachen Ressource statt. Auf der Abbildung 15 wird dieser Sachverhalt am Beispiel von der Eigenschaft "Titel" erläutert.

Abbildung 15: Mapping der Eigenschaft "Title".

Die CRM-Property *P102 has title* verbindet in dem ersten Tripel die primäre Ressource ("local_id:this") mit der einfachen Ressource (local_id:title1). Im zweiten Tripel bezeichnet die Dublin Core Property *dc:title* den Typ der einfachen Ressource, deren Wert (Harry Potter and the Order of the Phoenix) in dem RDF-Tripel als ein Literal abgespeichert wird:

Wie im Kapitel 2.6.2 angesprochen wurde, werden Literale im Unterschied zu den Ressourcen nicht identifiziert und können in einem RDF-Tripel nur als Objekte auftauchen. Sie werden jedoch mit den jeweiligen Relationen zusammen wahrgenommen und werden auf dem DDB-Portal als Ergebnis der Userrecherche zurückgegeben. Die entsprechenden Relationen fungieren dabei als Facetten, die als Filter für die benutzerorientierte Suche in dem Portal eingebaut werden. Diese Relationen werden mit Begriffen aus speziell dafür vorgesehenen kontrollierten Vokabularen ausgedrückt.[127]

4.2.2 Lokale und persistente DDB-Identifier

Lokale Identifier: Um die einzelnen Ressourcen identifizieren und zueinander in Verbindung setzen zu können, werden ihnen im Rahmen des Mappings lokale Identifier vergeben, die in den XSLT Transformern automatisch generiert werden. Die primäre Ressource, die pro Mapping nur einmal vorkommt (siehe Mapping-Schritt "abstract" im Kapitel 4.1), wird mit dem lokalen Identifier "this" versehen. Die lokalen Identifier der einfachen Ressourcen setzen sich zusammen aus einem eindeutigen String und einer Zahl, die sich aus dem Hochzählen der einfachen Ressourcen des gleichen Typs ergibt. Wenn eine Sammlung bspw. über zwei Urheber verfügt, bekommt jede Urheber-Instanz (E39 Actor) einen entsprechenden lokalen Identifier: bspw. "creator1" und "creator2". Somit

[127] Zu den kontrollierten Vokabularen im Rahmen des Mappings siehe Kapitel 4.2.3.

wird es möglich, den Ressourcen zusätzliche Informationen zuzuordnen, wie z.B. die Adresse des jeweiligen Urhebers. Events werden auch als Ressourcen betrachtet und verfügen dementsprechend auch über lokale Identifier.

DDB-Identifier: Sämtliche beim Mapping erzeugte lokale Identifier werden von der DDB-Software in persistente, im Rahmen der DDB einmalige, DDB-Identifier übersetzt. Die DDB-Identifier bilden zusammen mit der URL des DDB-Projekts *URI References*, mit denen die betreffenden Ressourcen im Projekt identifiziert werden können; sie entsprechen dem folgenden Muster:

4.2.3 Kontrollierte Vokabulare und die Klasse E55 Type

Eine wichtige Rolle bei einem effektiven Informationsretrieval spielen kontrollierte Vokabulare. Die Idee der kontrollierten Vokabulare wurde schon in den vorherigen Kapiteln dieser Arbeit im Kontext von Ontologien und Semantic Web dargelegt. Zusammenfassend definiert man kontrollierte Vokabulare als Sammlungen von Wörtern, die eindeutigen Begriffen zugeordnet werden, um Mehrdeutigkeiten der natürlichen Sprache zu reduzieren. Man unterscheidet zwischen verschiedenen Typen von kontrollierten Vokabularen, angefangen bei einfachen Wortlisten bis zu komplexeren, hierarchisch strukturierten Taxonomien, Thesauri, Klassifikationen und weiteren.[128]

Die kontrollierten Vokabulare werden beim Mapping für die Charakterisierung und Klassifikation der Ressourcen benutzt. Das CRM bietet mit der Klasse E55 Type die Verbindung zu domainspezifischen Ontologien und Thesauren, wie das folgende Beispiel aus dem Mapping von DC ins CRM demonstrieren soll:

[128] Zu den Typen, Struktur und Funktion der kontrollierten Vokabulare siehe Lindenthal, J.: Kontrollierte Vokabulare – Grundlagen, Hochschule für Angewandte Wissenschaften Hamburg, <http://www.bui.haw-hamburg.de/pers/ulrike.spree/remind/vokabulare.htm#4> (18.07.2011).

E36 Visual Item *P2 has type:*
E55 Type *P2 has type (is type of)* Video/IMTypes (E55 Type)[129]

Die Klasse E55 Type weist jedoch viele unterschiedliche Anwendungen auf. Neben Typisierung und Subtypisierung der Ressourcen wird die Klasse E55 für jegliche Format-, Material-, Maß- und Sprachangaben benutzt. Aus diesem Grund erfolgt die Typisierung der Ressourcen im Rahmen des DDB-Projekts dadurch, dass die entsprechenden Begriffe in dem RDF Tripel explizit abgespeichert werden:

```
<rdf:Description rdf:about="local_id:this">
  <crm:P2F.has_type rdf:resource="local_id:type1" />
</rdf:Description>

<rdf:Description rdf:about="local_id:type1">
  <dc:format> Video/IMTypes </ dc:format>
</rdf:Description>
```

Die Facetten, die dem Nutzer für das Retrieval als Filter angeboten werden, lassen sich aus dem indexierten Modell dadurch leichter extrahieren. Neben den gängigsten dcterms wie dc:identifier, dc:format, dc:language und dc:title werden speziell für die Facetten erstellte eigene DDB-Begriffe verwendet: facets:Urheber, facets:Herausgeber, facets:Beitragende, facets:Quelle, facets:Halter und weitere.

Die kontrollierten Vokabulare bleiben nach dem Mapping weiterhin nutzbar, indem sie die Möglichkeit zur "globalen" (außerhalb der Rahmen des DDB-Projekts nutzbaren) Referenzierung der Objekte geben. Das ermöglicht es, Beziehungen zu weiteren Ressourcen festzustellen, ähnlich wie es mit Hilfe des TGN Eintrags möglich wurde, ein Dokument mit einem Foto in Verbindung zu setzen, obwohl die Metadaten nichts Gemeinsames aufwiesen (Kapitel 2.3). Zu den wichtigsten Vertretern der kontrollierten Vokabulare zählen im diesem Zusammenhang die sog. Normdateien (*authority files*), die der eindeutigen Referenzierung von Objekten und der Auflösung der Mehrdeutigkeiten und Unklarheiten dienen, welche aufgrund der Verwendung von unterschiedlichen Bezeichnungen für gleiche Konzepte auftreten.[130] Vor allem die Normdateien für Personen- und Ortsnamen sind in diesem Kontext von Bedeutung. In der sog. Personennamendatei

[129] Siehe Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: DC.type mapping to CIDOC/CRM, Technical Report, DELOS WP5-Task 5.5, Ontology Driven Interoperability, January 2007, S. 26.

[130] Siehe Lülfing, D., Benkert, H., & Siebert, S.: *95. Deutscher Bibliothekartag in Dresden 2006 : Netzwerk Bibliothek*, Frankfurt am Main: Vittorio Klostermann, 2007, S. 71.

(PND) werden "alle Personennamen zusammengeführt, die für Formal- und Sacherschließung sowie nationale Katalogisierungsunternehmungen relevant sind."[131]

4.2.4 Pseudo-Elemente

Da CRM eventzentriert ist und die Quellmetadaten meistens nicht, sollen bei Bedarf sog. Pseudo-Elemente erzeugt werden. Das Prinzip eines Pseudo-Elements soll am Beispiel des EAD Elements <author> erklärt werden. Das Element <author> umfasst Namen von Personen oder Institutionen, die für den Inhalt des Findbuchs verantwortlich sind. Nach Ansicht der Entwickler des CRM lassen sich solche Konzepte wie Bearbeiter der Ressource nicht direkt mit der Ressource verbinden, sondern lediglich durch eine Instanz der Klasse *E7 Activity*:

> "As the CIDOC CRM, in fact, is an event-centred model, it is impossible the <author> to be linked directly to the <titlestmt> through a certain property, whereas the latter is mapped to entity E31 Document and the <author> must be linked to an entity that denotes activity. Therefore we consider creating a „pseudo – element", named *****, mapped to the entity E65 Creation Event."[132]

Demzufolge wird ein Pseudo-Element angenommen, das die Erzeugung der Ressource symbolisiert und der Instanz der CRM Klasse *E65 Creation* entspricht. Konzepte wie Urheber der Ressource, Erstellungsdatum, Erstellungstechnik, Erstellungsort und weitere Informationen, die die Ereignisse betreffen, werden mit dieser Instanz verknüpft.

4.2.4 Appellations

Des Weiteren werden für das Mapping aggregierende CRM Konzepte benötigt wie Appellations. Die Klasse *E41 Appellation* umfasst:

> "…alle richtigen Namen, Worte, Ausdrücke oder Kodierungen, bedeutungsvoll oder auch nicht, die benutzt werden oder benutzt werden können, um eine bestimmte Instanz einiger Klassen in einem bestimmten Zusammenhang zu identifizieren."[133]

---

131 Deutsche Nationalbibliothek, <http://www.d-nb.de/standardisierung/normdateien/pnd.htm>

132 Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: EAD mapping to CIDOC/CRM, Technical Report, DELOS WP5-Task 5.5., Version 0.2, Ontology Driven Interoperability, 2007, S. 11.

133 Doerr M., Lampe, K. & Krause S. (hrg. & übers.): Definition des CIDOC Conceptual Reference Model, Version 5.0.1, ICOM Deutschland, 2010, S.78.

Ihre Kinderklasse *E82 Actor Appellation* enthält Konzepte wie Name, Versicherungsnummer, PND Nummer und weitere Konzepte, die zur Bezeichnung und Identifizierung eines Akteurs dienen (siehe Abbildung 5). Auch die Klasse *E44 Place Appellation* ist für das Mapping von großer Bedeutung: durch diese Klasse und ihre Unterklassen (*E45 Address*, *E47 Spatial Coordinates*, *E48 Place Name* oder *E46 Section Definition*) wird die Identifizierung eines Ortes (E53 Place) vorgenommen.

4.3 Cluster

Der Mapping-Ansatz, der in dieser Arbeit vorgestellt wird, beruht auf in der Idee der vorgefertigten konzeptuellen Mappings (Clustern). Das Konzept der Cluster wurde von dem Mapping-Vorschlag der CRM Entwickler für die Dublin Core Ressourcentypen inspiriert.[134] In diesem Vorschlag werden unterschiedliche Mappings für elf Dublin Core Objekttypen vorgestellt. So werden auch im Rahmen des DDB Projekts die Mappings der gängigsten Objekttypen im Bereich des Kulturerbes bereitgestellt – u.a. Text, Sound Text, Sound, Image, Video, Sammlung, Physisches Objekt und weitere. Diese Objekttypen werden um einige Event- und Akteurtypen bereichert. Der Ansatz wird folglich für unterschiedliche Metadatenformate angewendet u.a. DC, EAD und LIDO.

Die Funktionsweise dieses Clusteransatzes besteht darin, dass man für den gleichen Ressourcentyp, unabhängig von dem Quellmetadatenformat, das gleiche Mapping anwenden kann. Das führt dazu, dass die Mappings für alle Ressourcen des gleichen Typs gleich ausfallen. Somit werden Austausch, Interoperabilität und Ressourcenvergleich erleichtert.

4.3.1 Cluster Library

Die Cluster bilden eine Library im Sinne der Informatik, genauer eine Hierarchie, die aus konkreten, allgemeinen und Basisclustern besteht. Für jeden geeinigten Ressourcentypen wird ein Cluster erzeugt. Die Cluster werden in XSLT als Templates realisiert und können wie folgt generisch aufgerufen werden:

Konkrete Templates → Allgemeine Templates → Basistemplates

[134] Siehe Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: DC.type mapping to CIDOC/CRM, Technical Report, DELOS WP5-Task 5.5, Ontology Driven Interoperability, January 2007, <http://www.cidoc-crm.org/crm_mappings.html> (13.07.2011).

Konkrete Cluster sind metadatenformatabhängig: in ihnen wird entschieden, welche Informationen aus den Quellmetadaten ins DDB-Metadaten übersetzt werden sollen. Die Namen der konkreten Templates werden durch den Namen des Metadatenformats und den Typ der Ressource gekennzeichnet: [dc_text], [lido_image], [ead_collection], [lido_text] und weitere. Den konkreten Templates wird die CRM Semantik komplett vorenthalten.

Allgemeine Cluster sind Cluster, welche CRM Semantik beinhalten, die einem bestimmten Ressourcentyp zu Grunde liegt. Diese Cluster werden als parametrisierte Templates realisiert, deren Parameter den CRM Beziehungen dieses bestimmten Ressourcentyps entsprechen. Die Parameterwerte entsprechen den Metadaten aus der Quelldatei. Jeder belegte Parameter wird in dem allgemeinen Template in die DDB-Metadatenstruktur übersetzt. Die Namen der Templates stimmen mit der CRM-Klassennotation überein, die diesem Ressourcentyp entspricht: [linguistic_object], [image], [collection], [physical_object] und weitere. Die Parameter, die den CRM- Beziehungen entsprechen, die dieser Ressourcentyp von seiner Elternklasse erbt, werden an das entsprechende Basistemplate weitergereicht.

Basiscluster beinhalten die Semantik, die mehreren Ressourcentypen zu Grunde liegt. Das Basistemplate [information_object] wird bspw. von solchen allgemeinen Templates wie [image], [linguistic_object] und [visual_item] aufgerufen (siehe Abbildung C1). Diese Cluster-Hierarchie entspricht der CRM-Klassenhierarchie: die Klasse *E73 Information Object* ist die Elternklasse der Klassen *E33 Linguistic Object*, *E38 Image* und *E36 Visual Item*. Durch die Auslagerung der gemeinsamen Eigenschaften in BasisTemplates wird einerseits das Vererbungskonzept simuliert, andererseits sorgt es für die Einheitlichkeit des Mappings.

Der Ansatz von den verschiedenen Templates wird im Folgenden an einem Dublin Core Textbeispiel erklärt.

4.3.2 Clusterbeispiel

Das DC Element <dc:type> beinhaltet die Informationen zu dem Typ der Ressource.[135] In dem Dublin Core Beispiel auf der Abbildung B1 handelt es sich um einen Text, dementsprechend wird das konkrete Template [dc_text] aufgerufen.[136] In das Textprofil fallen Bücher, Manuskripte, Abschriften, Dokumente und andere schriftliche textbasierte Zeugnisse. Im CRM entspricht diesem Profil vor allem die Klasse *E33 Linguistic Object.* In dem T. [dc_text] wird das allgemeine T. [linguistic_object] aufgerufen. Das T. [linguistic_object] beinhaltet die Semantik der Klasse *E33 Linguistic Object*. Als Parameter dieses Templates werden die Eigenschaften der *E33* Klasse definiert (siehe Abbildung 16). Die Parameternotation entspricht in der Regel den Namen der Dublin Core Elemente, ohne die Namenspace *dc*. Dies hat den Grund, dass diese Notation allgemein bekannt und leicht verständlich ist.

Die wichtigsten Parameter betreffen solche Konzepte wie Typ und Titel der Ressource, ihre Erstellungsdaten[137] und Auslagerungsort sowie Relationen zu anderen Ressourcen. Auch zusätzliche nicht typisierte Inhaltsangaben und Beschreibungen der Ressource sind von Bedeutung, weil man später in selbigen bestimmte Angaben finden könnte, die helfen könnten, weitere Beziehungen zu anderen Ressourcen im Kontext des *Semantic Web* festzustellen.

```
<xsl:template name="dc_text">
        <xsl:param name="id" select="'this'" />
        <xsl:param name="provider_id" select ="'provider'" />

        <xsl:call-template name="linguistic_object">
                <xsl:with-param name="id" select="$id" />
                <xsl:with-param name="identifier" select="dc:identifier" />
                <xsl:with-param name="description" select="dc:description" />
                <xsl:with-param name="subject" select="dc:subject" />
                <xsl:with-param name="rights" select="dc:rights" />
                <xsl:with-param name="source" select="dc:source" />
                <xsl:with-param name="title" select="dc:title" />
                <xsl:with-param name="language" select="dc:language" />
                <xsl:with-param name="type" select="dc:type" />
                <xsl:with-param name="coverage" select="dc:coverage" />
                <xsl:with-param name="relation" select="dc:relation" />
                <xsl:with-param name="isShownAt" select="dc:identifier" />
            </xsl:call-template>
```

135 Siehe im Kapitel 4.4.1 die Bestimmung der DC Ressourcentyps
136 Das Wort Template wird im Folgenden als "T." abgekürzt.
137 Zu diesen zählen die Angaben, wo, wann und von wem sie erstellt wurde.

```
<xsl:call-template name="man_made_object">
        <xsl:with-param name="id" select="$id" />
        <xsl:with-param name="provider_id" select="$provider_id" />
        <xsl:with-param name="format" select="dc:format" />
      </xsl:call-template>

<xsl:call-template name="creation">
        <xsl:with-param name="id" select="'creation'" />
        <xsl:with-param name="object_id" select="$id" />
        <xsl:with-param name="date" select="dc:date " />
        <xsl:with-param name="creator" select="dc:creator" />
          <xsl:with-param name="publisher" select="dc:publisher" />
        <xsl:with-param name="contributor" select="dc:contributor " />
    </xsl:call-template>
```

Abbildung 16: Ausschnitt aus dem Template "dc_text".

Der Titel der Ressource wird als Parameter *title* an das T. [linguistic_object] übergeben (Siehe Abbildung 16). Das gleiche T. [linguistic_object] kann beim Mapping einer LIDO Ressource folgendermaßen aufgerufen werden:

```
<xsl:template name="lido_text">
    <xsl:call-template name="linguistic_object">
      <xsl:with-param name="title" select=
          "//lido:titleSet/lido:appellationValue[not(@lido:pref='alternate')]" />
```

In dem T. [linguistic_object] werden in die DDB-Metadatenstruktur nur in Parameter übersetzt, die speziell für die Klasse *E33 Linguistic Object* gelten, nämlich *language* und *translation*. Auf der Abbildung der CRM Hierarchie (siehe Anhang A) ist zu sehen, dass die Klasse *E33* nur über zwei Beziehungen – *P72 has language* und *P73 has translation* – verfügt.

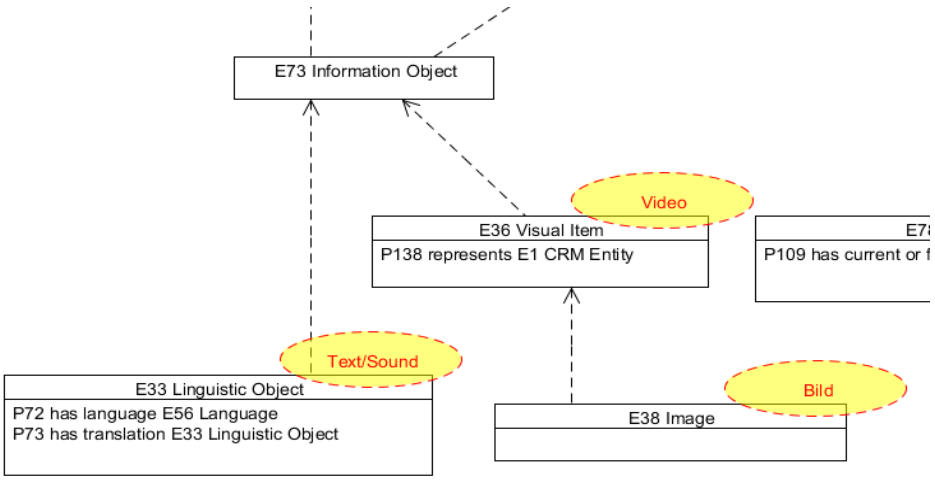


Abbildung 17: Ausschnitt aus der Mapping-relevanten CRM-Klassenhierarchie (siehe Anhang A).

Alle anderen Methoden, u.a. auch die Methode *P102 has title*, erbt die Klasse *E33* von ihrer Superklasse *E73 Information Object*. So werden die restlichen Parameter inklusive *title* an das Basistemplate [information_object] übergeben.

Jedoch lassen sich mit der Klasse *E33 Linguistic Object* nur die Eigenschaften des immateriellen Textes behandeln und nicht des Mediums, das diesen Text beinhaltet (siehe Kapitel 3.2). Um das Objekt mit seinem Provider (*E39 Actor*) oder seinem Aufbewahrungsort (*E53 Place*) in Verbindung zu setzen, weist man ihm auch die Klasse *E22 Man-Made Object* zu. In unserem Mapping-Ansatz wird dies dadurch realisiert, dass in dem T. [dc_text] neben dem T. [linguistic_object] auch ein T. [man_made_object] aufgerufen wird, und zwar mit dem gleichen lokalen Identifier der Ressource ($id). Als einer der Parameter bekommt das T. [man_made_object] den lokalen Identifier der Provider-Instanz ($provider_id*)* übergeben (Abbildung 16). Zu dem Parameter *provider_id* wird in dem T. [man_made_object] eine Beziehung erzeugt, die die Ressource mit dem Provider verbindet:

```
<rdf:Description rdf:about="local_id:this">
<crm: P50F.has_current_keeper rdf:resource="local_id:provider"/>
</rdf:Description>
```

Die Typisierung der Provider-Instanz erfolgt in dem T. [actor]. An das T. [actor] neben dem Parameter *provider_id* werden weitere Informationen über den Provider, wie z.B. sein Name, URI, Email und Adresse übergeben (Siehe Anhang D). Das Actor-Profil symbolisiert alle Instanzen von Akteuren, seien es einzelne Personen, Institutionen oder Gruppen von Personen. Das Profil entspricht der Klasse *E39 Actor* im CRM. Die weiteren Unterscheidungen in *E21 Person* oder *E74 Group* wären auch möglich. Wichtig sind in dem Profil die Eigenschaften, die diese spezielle Actor-Instanz von den anderen unterscheiden, z.B. Name, Versicherungsnummer, URI oder PND Nummer.

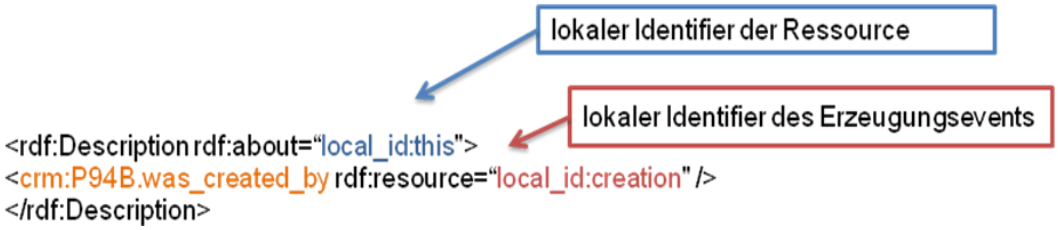
Bei dem Aufruf des T. [actor] wird auch die Rolle der Actor-Instanz als Parameter übergeben. Die Akteure, die bspw. in einem *E12 Production* Event mitgewirkt haben, bekommen entsprechend die Rolle "Urheber", "Verleger" oder "Mitwirkende". Die Provider-Instanzen bekommen die Rolle "Halter", die Institutionen, bei denen die Ressourcen sich befinden, die Rolle "Repository". Die Actor-Rollen finden ihre

Ausprägung in den DDB-kontrollierten Vokabularen und werden als Facetten wahrgenommen:

```
<rdf:Description rdf:about="local_id:creator1">
    <crm: P131F.is_identified_by rdf:resource="local_id:creator1_name" />
  </rdf:Description>

  <rdf:Description rdf:about="local_id:creator1_name">
    <facets:Urheber>Thomas Pynchon</facets:Urheber >
  </rdf:Description>
```

Wie in dem Kapitel III angesprochen wurde, spielen die Events im CRM eine große Rolle: durch sie wird die Ressource mit ihrem Urheber, Erstellungsort, Erstellungszeitraum verbunden. Um das Event mit der Ressource zu verbinden, wird der lokale Identifier der Ressource an das T. [creation] übergeben. Dazu wird im T. [creation] ein RDF-Tripel erzeugt:



Die Informationen zu dem Erstellungsprozess der Ressource werden an das T. [creation] übergeben. In [creation] wird wiederum das Basistemplate [activity] aufgerufen, in dem die Eigenschaften für alle Events definiert werden (Siehe Anhang C). Das Ergebnis dieses DC Mappings ist graphisch auf der Abbildung B3 zu sehen.

4.4 Bestimmung des Ressourcentyps

In der Regel basiert eine Clusteranwendung auf dem Typ der Ressource. Wenn es sich um Digitalisate handelt, ist mit dem Typ der ursprüngliche Typ der digitalisierten Ressource gemeint. Die Feststellung dieses Typs erweist sich als eines der grundlegendsten Probleme des Mappings. In Fällen jedoch, bei denen der Typ der Ressource nicht genau bestimmt werden konnte, wird ein hierfür von den Entwicklern des CRM vorgeschlagener Mapping-Vorgang eingesetzt. In den nun folgenden Abschnitten wird das Problem der Ressourcentypbestimmung an Beispielen aus den Quellmetadaten ausführlicher diskutiert.

4.4.1 Dublin Core

Beim Dublin Core erfolgt die Feststellung des Typs einer Ressource in simpler Art und Weise. Wie an dem Beispiel im Kapitel 4.3.2 gezeigt wird, wird der Typ der Dublin Core Ressource in dem Element <dc:type> angegeben. Das Element wird wie folgt definiert:

> "The type of the original analog or born digital object as recorded by the content holder, this element typically includes values such as photograph, painting, sculpture etc."[138]

Die Phrase *"original analog or born digital object"* weist darauf hin, dass es sich tatsächlich um den Typ der Informationsressource handelt und nicht um das Medium oder Format der Ressource, was in DC dem Element <dc:format> entspricht. Grundsätzlich unterscheidet man zwischen elf Dublin Core Elementen, darunter Collection, Text, Image, Video, Sound und Physical Object.[139]

4.4.2 LIDO

Der Typ einer LIDO Ressource wird bei der Metadatenerfassung in der Regel in dem LIDO Element <lido:category> angegeben und entspricht primär einer der drei CRM Klassen: *E22 Man-Made Object*, *E20 Biological Object* oder *E25 Man-Made Feature*. Die Möglichkeit zur weiteren Spezifizierung des Ressourcentyps besteht jedoch in der Bestimmung von weiteren LIDO Elementen, u.a. <lido:classification> und <lido:objectWorkType>. Nach LIDO Definition enthält das Element <lido:classification> "*information about the type of the object*".[140] Sein Attribut *europeana:type* weist auf eins von den vier Europeana Typen – *Text*, *Image*, *Video* und *Sound*.

```
<lido:classification lido:type="europeana:type">
    <lido:term>IMAGE</lido:term>
</lido:classification>
```

Basierend auf einem Hinweis der Entwickler des EDM lässt sich folgern, dass Europeana Typen ihre Entsprechung in den DC Typen finden:

---

[138] European Union: Definition of the Europeana Data Model elements, Version 5.2.1, v. 07.03.2011, S. 50, <http://www.version1.europeana.eu/c/document_library/get_file?uuid=aff89c92-b6ff-4373-a279-fc47b9af3af2&groupId=10605> (29.07.2011).

[139] Siehe Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: DC.type mapping to CIDOC/CRM, Technical Report, DELOS WP5-Task 5.5, Ontology Driven Interoperability, January 2007, <http://www.cidoc-crm.org/crm_mappings.html> (13.07.2011).

[140] Siehe Coburn, E., Light, R., McKenna, G., Stein, R., & Vitzthum, A.: LIDO – Lightweight Information Describing Objects Version 1.0, S. 4.

> Providers are recommended to map the values entered in this element [dc:type, M.A.] to the four material types used in Europeana: TEXT, IMAGE, SOUND and VIDEO.[141]

Da es bei dem <dc:type> um "*original analog or born digital object*" handelt, könnte dem Ressourcentyp in LIDO eine der Konstanten: Text, Image, Video oder Sound zugeordnet werden. Im Folgenden wird diese Vermutung am Beispiel von LIDO Metadaten getestet.

Bei dem hierzu ausgewählten Beispielobjekt handelt es sich um ein Foto des Gemäldes "*Perseus befreit Andromeda"*, das von Jacopo Palma im Jahre 1560 vollendet wurde.[142] Der Europeana Typ wird hier als IMAGE kennzeichnet. Jedoch geht daraus nicht eindeutig hervor, auf was genau dieser Europeana Typ sich bezieht: auf das Foto des Gemäldes oder auf das Gemälde selbst, das sich in einem der Staatlichen Museen Kassel befindet.

Title: Perseus befreit Andromeda
Urheber: Jacopo Palma
Erzeugungszeitraum:1560

Die Quellmetadaten (der Urheber, das Erzeugungsdatum und Erzeugungsort) betreffen jedoch das Gemälde und nicht das Foto, deswegen sollte die Primärressource hier das Gemälde sein. Dennoch stellte sich bei der genaueren Untersuchung von mehreren LIDO Dokumenten heraus, dass der Eintrag des Europeana Typs des Öfteren fehlt und in der Regel auf die digitale Ressource hin bestimmt wird und nicht auf den ursprünglichen Objekttyp. Diese Gegebenheit könnte – nebenbei bemerkt – die Abwesenheit des

[141] European Union: Definition of the Europeana Data Model elements, Version 5.2.1, v. 07.03.2011, S. 50.

[142] Die originalen Metadaten stammen aus dem Bildarchiv Foto Marburg: <http://www.fotomarburg.de/> (13.07.2011).

passenden Europeana Typs für 3D Objekte wie bspw. Statuen erklären, die in DC dem Typ "Physical Object" zugeordnet werden.

In der LIDO Struktur kommen jedoch andere Begriffe vor, die dazu verhelfen können, den Typ der Informationsressource genauer zu bestimmen. Das Element <lido:objectWorkType> bietet bspw. eine Definitionsmöglichkeit des ursprünglichen Typs der Ressource. Dies erfordert einen zusätzlichen Mapping-Schritt, indem die Begriffe der LIDO objectWorkType-Terminologie den entsprechenden Clustern zugeordnet werden:

4.4.3 EAD

Die EAD Typbestimmung gestaltet sich wesentlich einfacher als bei LIDO. In Deutschland handelt es sich bei EAD-Beschreibung meistens um eine Sammlung von Archivalien. Demgemäß entsprechen EAD-Ressourcen meistens dem Collection Cluster. Es ist schwer, den Typ von Archivalien, aus denen eine Sammlung besteht, exakt zu bestimmen. In der Regel kann eine Archivaliensammlung auch aus mehreren Objekten von verschiedenen Typen bestehen. Man transformiert daher die Items einer Sammlung, wie in der Definition des CRM vorgegeben ist, in die Klasse *E19 Physical Object*; diese entspricht in unserem Mapping dem physical_object Cluster.

5 Schlusswort

Das Projekt Deutsche Digitale Bibliothek hat sich zum Ziel gesetzt, ein Online-Portal zu erstellen, auf dem die Informationen zu den Ressourcen des deutschen Kulturerbes aller Art zusammen getragen werden. In diesem Zusammenhang spielt das Metadatenmapping eine Rolle von hoher Relevanz. Sein Anspruch liegt darin es zu ermöglichen, die unterschiedlichen, institutionsabhängigen Metadatenformate in ein einheitliches institutions- und metadatensprachen- unabhängiges Datenmodell zu transformieren.

Die besondere Herausforderung des Mappings, das im praktischen Teil dieser Arbeit erarbeitet wurde, bestand vor allem in dem unterschiedlichen Charakter der Quell- und Zielmetadaten. Als Quellformat liegen die deskriptiven Metadaten DC, LIDO und EAD vor. Im Gegensatz dazu stellt die objektorientierte Ontologiesprache CIDOC CRM das Zielformat des Mappings dar. Bei diesem Mapping handelt es sich nicht um eine bloße Transformation eines Elementes aus der Quelldatei in ein ihm äquivalentes Element in der Zieldatei (one-to-one Mapping). Vielmehr wird das betreffende Element einer semantischen Interpretation ins CIDOC CRM unterzogen, in Folge dessen das Element einer CRM-Klasse zugeordnet und in seinen Beziehungen zu den anderen Objekten bestimmt wird. Erschwerend dazu kommt die Tatsache, dass die Transformationssprache (XSLT) selbst nicht objektorientiert ist.

Darüber hinaus mussten während der Ausarbeitung der Mappings grundlegende Probleme der Quellmetadaten berücksichtigt werden, auf die im Kapitel 2.7.2 eingegangen wurde. Darunter befand sich auch eines der Hauptprobleme im Bereich der digitalen Bibliotheken, nämlich das Problem der Datenqualität. Dieses wurzelt in fehlerhaften oder nicht ausreichend vorhandenen Metadaten, was die Auffindung von Bibliotheksobjekten verkompliziert und diese bisweilen unmöglich macht. Qualitätssichernde Normalisierungsprozesse finden vor dem Mapping nicht statt und somit musste eine Reihe an speziellen Lösungen für die Probleme entwickelt werden. Für das Problem der „deplatzierten“ Informationsinhalte (siehe Kapitel 2.7.2) wird als Lösung das Mapping an den betroffenen Instanzen abstrakter definiert. Die unterschiedlichen Versionen eines Metadatenformats mussten unter der Berücksichtigung ihrer Version einzeln behandelt

werden; hinzu kamen die Fälle des institutionsabhängigen Gebrauchs der Quellmetadatenstandarte.

Die grundlegende Idee, auf der das Mapping-Verfahren basiert, arbeitet mit dem Konzept von Clustern. Es wurde ein Set an konzeptuellen Mappings von gängigsten Typen von Objekten und Events des Kulturbereichs definiert und beim Mapping diverser Ressourcentypen entsprechend angewendet. Ihre praktische Realisierung finden die Cluster in XSLT Templates, die aufgerufen werden, sobald eine vorausgesetzte Bedingung erfüllt wird. Grundsätzlich berücksichtigt das hiesige Cluster- Konzept eine Unterscheidung zwischen den "konkreten" und den "allgemeinen Clustern". Die "konkreten Cluster" betreffen die Quellmetadaten und ihre verschiedenen Ausprägungen und bieten die Flexibilität, von ihren zukünftigen Anwendern nach eigenen Mustern erstellt werden zu können. Den "konkreten Clustern" entsprechen in der objektorientierten Programmierung die benutzerdefinierten Applikationsklassen. Ein Beispiel eines "konkreten Clusters" zeigt Abbildung 16.

Die "allgemeinen Cluster" betreffen die Semantik des CIDOC CRM und entsprechen in der objektorientierten Programmierung dem Konzept der Bibliotheksklassen. Mit ihrem Aufruf wird ein leichter Anschein von Objektorientierung erweckt. Diesem Anschein nach kann, je nach Cluster, die betreffende Ressource einem Typ zugeordnet werden. Die Parameter, mit denen ein allgemeines Cluster aufgerufen werden kann, entsprechen den Eigenschaften dieses Ressourcentyps. Die belegten Parameter werden in den Clustern ins DDB-Metadatenmodell interpretiert (Kapitel 4.1). Die Benutzung von "allgemeinen Clustern" sorgt dafür, dass die Mappings von gleichen Ressourcentypen identisch ausfallen. Die Simulation einer objektorientierten Sprache wird zusätzlich durch den Vererbungseffekt verstärkt, der erreicht wird, indem in den "allgemeinen Clustern" die ihnen entsprechenden "Basiscluster" intern aufgerufen werden. Die Benennung der "allgemeinen" und "Basiscluster" richtet sich im Wesentlichen nach der CRM Klassenhierarchie.

Der relevante Vorteil der Differenzierung zwischen den o.g. Arten von Clustern besteht in der Möglichkeit der unabhängigen Verarbeitung der Quell- und der Zielmetadaten, sodass bei möglichen Modifizierungen die nötigen Anpassungen nur in den entsprechenden Clustern vorgenommenen werden. Ein weiterer gewinnbringender Aspekt dieser

Unterteilung ist seine, auch Nicht-CRM-Experten zugängliche, Handhabung. Grundkenntnisse von XSLT und gute Kenntnisse der Quellmetadatenstandards stellen sich als zureichend aus, um bestimmen zu können, welche Informationen der Quelldatei beim Aufruf der Cluster als Parameter übergeben werden können. Die Information darüber, welcher CRM Klasse der Typ der Ressource entspricht, die in Quellmetadaten beschrieben wird, genügt, um den Aufruf des gleichnamigen Templates zu betätigen.

Ferner ist es in dem DDB Projekt vorgesehen, ein grafisches Tool zur Erzeugung der DDB-Metadaten zu erstellen. Das Werkzeug soll den Benutzern die Möglichkeit bieten, das Mapping-Verfahren selbst zu gestalten. Der ganze Mapping-Prozess wird somit intuitiver gestaltet und wird dadurch vereinfacht, dass die Cluster im Hintergrund aufgerufen werden und der Benutzer sich nicht mit der XSLT auskennen muss. Das Programm soll u.a. eine geeignete Visualisierung für das Mapping-Ergebnis gewährleisten.

6 Literaturverzeichnis

[1] Berners-Lee, T.: Uniform Resource Locators (URL), December 1994.
URL: <http://tools.ietf.org/pdf/rfc1738.pdf> (10.06.2011)

[2] Berners-Lee, T., Hendler, J. & Lassila, O.: The Semantic Web, *Scientific American*, Mai 2001. URL: <http://old.hki.uni-koeln.de/temp/SemWebSeminal.pdf> (10.06.2011)

[3] Berners-Lee, T.; Bizer, C. & Heath, T.: Linked Data – The Story So Far, to appear in Special Issue on Linked Data.
URL: <http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf> (29.07.2011).

[4] Bundesregierung, Presse- und Informationsamt: Deutsche Digitale Bibliothek, 04.12.2009.
URL:<http://www.bundesregierung.de/nn_774/Content/DE/StatischeSeiten/Breg/BKM/2008-02-26-deutsche-digitale-bibliothek.html> (10. 06.2011)

[5] Bund-Länder-Fachgruppe DDB: Fachkonzept zum Aufbau und Betrieb einer Deutschen Digitalen Bibliothek, 16.02.2008.
URL: <http://www.deutsche-digitale-bibliothek.de/dokumente.htm> (10. 06.2011)

[6] Byrne, D. J.: *MARC manual: understanding and using MARC records.* Englewood, Colo.: Libraries Unlimited, 1998.

[7] Coburn, E., Light, R., McKenna, G., Stein, R., & Vitzthum, A.: LIDO – Lightweight Information Describing Objects Version 1.0, November 2010.
URL: <http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf>
(10. 06 2011)

[8] Christen, M.: Machbarkeitsstudie zum Aufbau und Betrieb einer "Deutschen Digitalen Bibliothek". Die IT-Architektur der DDB auf der Basis des Fachkonzeptes der Bund-Länder-Fachgruppe, Frankfurt am Main, den 23.07.2008.
URL: <http://www.deutsche-digitale-bibliothek.de/pdf/machbarkeitsstudie_20080723.pdf> (11.08.2011)

[9] Deutscher Museumsbund: Das CIDOC Conceptual Reference Model: Eine Hilfe für den Datenaustausch? Oktober 2004.
URL: <http://www.cidoc-crm.org/docs/cidoc_paper_german.pdf> (10. 06 2011)

[10] Dickinson, I.: Jena Ontology API, 20.05.2011.
URL: <http://jena.sourceforge.net/ontology/> (06.06.2011)

[11] Doerr, M.: The CIDOC CRM – an Ontological Approach to Semantic Interoperability of Metadata, 2002.
URL: <http://www.cidoc-crm.org/docs/ontological_approach.pdf > (10. 06 2011)

[12] Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: EAD mapping to CIDOC/CRM, Technical Report, DELOS WP5-Task 5.5., Version 0.2, Ontology Driven Interoperability, 02.03.2007. URL: <http://www.cidoc-crm.org/crm_mappings.html> (10. 06 2011)

[13] Doerr, M., Stasinopoulou, T., Bountouri, L., Kakali, C., Lourdi, I., Papatheodorou C. & Gergatsoulis, M.: Ontology-Based Metadata Integration in the Cultural Heritage Domain, in: *Asian digital libraries: looking back 10 years and forging new frontiers*: *10th International Conference on Asian Digital Libraries, ICADL, Hanoi, Vietnam.* Berlin, New York: Springer, 2007.

[14] Doerr, M., Hunter, J. & Lagoze, C.: Towards a Core Ontology for Information Integration, *Journal of Digital Information*, Vol 4, No 1. 2003.
URL: <http://journals.tdl.org/jodi/article/view/92/91> (10.06.2011)

[15] Doerr, M., Plexousakis, D., Kopaka, K. & Bekiari, C.: Supporting Chronological Reasoning in Archaeology, 2004.
URL: <http://www.cidoc-crm.org/docs/caa2004_supporting_chronological_reasoning.pdf> (10.06.2011)

[16] Doerr, M., Gradmann, S., Hennicke, S., Isaac, A., Meghini, C. & Van de Sompel, H.: The Europeana Data Model (EDM). IFLA 2010, World Library and Information Congress: 76th IFLA General Conference and Assembly, 10-15 August, 2010, Gothenburg, Sweden.
URL: <http://www.ifla.org/files/hq/papers/ifla76/149-doerr-en.pdf> (10.06.2011)

[17] Doerr M.: Semantic Interoperability, Epistemic Networks and the CIDOC CRM
Foundation for Research and Technology – Hellas Institute of Computer Science Hamburg, December 3, 2007. URL: <http://www.hdh.uni-hamburg.de/webfm_send/8> (16.07.2011)

[18] Doerr, M., Kakali, K., Papatheodorou, C. & Stasinopoulou, T.: DC.type mapping to CIDOC/CRM, Technical Report, DELOS WP5-Task 5.5, Ontology Driven Interoperability, January 2007.
URL: <http://www.cidoc-crm.org/crm_mappings.html> (10. 06 2011)

[19] Doerr, M., Kondylakis, H., Plexousakis, D.: Mapping Language for Information Integration. Technical Report 385, December 2006.
URL: <http://www.cidoc-crm.org/crm_mappings.html> (10. 06 2011)

[20] Doerr M., Crofts, N., Gill, T., Stead S. & Stiff M.: Definition of the CIDOC Conceptual Reference Model, Version 5.0.2, Januar 2010.
URL: <http://www.cidoc-crm.org/docs/cidoc_crm_version_5.0.2.pdf> (10. 06 2011)

[21] Doerr M., Lampe, K. & Krause S. (hrg. & übers.): Definition des CIDOC Conceptual Reference Model, Version 5.0.1, ICOM Deutschland, 2010.
URL: <http://www.icom-deutschland.de/client/media/380/cidoccrm_end.pdf>
(10. 06 2011)

[22] Eckstein, R., Eckstein, S.: *XML und Datenmodellierung*. Dpunkt-Verl., 2004.

[23] Ehrig, M. & Studer, R.: Wissensvernetzung durch Ontologien in: Pellegrini, T.: *Semantic Web Wege zur vernetzten Wissensgesellschaft*, mit 4 Tabellen, Berlin, Heidelberg, New York: Springer, 2006, S. 472-472.

[24] European Union: Definition of the Europeana Data Model elements, Version 5.2.1, 07.03. 2011. URL: <http://www.version1.europeana.eu/c/document_library/get_file?
uuid=aff89c92-b6ff-4373-a279-fc47b9af3af2&groupId=10605> (10. 06 2011)

[25] Fox, M. J.: EAD Cook Book, 2002.
URL: <http://jefferson.village.virginia.edu/ead/cookbookhelp.html> (10. 06 2011)

[26] Fraunhofer Institut für Intelligente Analyse- und Informationssysteme: Auf dem Weg zur DDB, 04.03.2008.
URL: <http://www.deutsche-digitale-bibliothek.de/pdf/auf_dem_weg_studie.pdf> (10.06.2011)

[27] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, Vol. 5, 1993, 199-220.

[28] Guarino, N.: Formal ontology in information systems, in: Guarino, N. (hrg.): *Formal ontology in information systems.* Proceedings of the first international conference (FOIS'98), Trento, Italy, 6-8 June, Amsterdam; Washington, DC: IOS Press; Tokyo: Omsha, 1998.

[29] Hesse, W.: Aktuelles Schlagwort Ontologie(n), in: *Informatik-Spektrum*, Vol. 25, N. 6, 2007, S. 477-480.

[30] Hesse,W., Krzensk, B.: Ontologien in der Softwaretechnik, 2004.
URL:<http://www.uni-marburg.de/fb12/informatik/homepages/hesse/publikationen/dateien/h_k_04.pdf> (10.06.2011)

[31] Heuer, A., Saake, G., & Sattler, K.-U.: *Datenbanken: Konzepte und Sprachen.* Heidelberg: Mitp bei Redline, 2008.

[32] Kay, M.: *XSLT 2nd Edition: programmer's reference*. Canada, Wrox Press, 2004.

[33] Kommission der Europäischen Gemeinschaften: Europas kulturelles Erbe per Mausklick erfahrbar machen, Stand der Digitalisierung und Online-Verfügbarkeit kulturellen Materials und seiner digitalen Bewahrung in der EU, 2003.
URL:<http://ec.europa.eu/information_society/activities/digital_libraries/doc/communications/progress/communication_de.pdf> (10. 06 2011)

[34] Kuhnt, M.: Ontologiesprachen im Kontext des Semantic Web, Hauptseminarausarbeitung, Abt. KI, Universität Ulm, 2003.
URL: <http://www.informatik.uni-ulm.de/ki/Edu/Seminare/Semantic.Web/WS0203/4-Kuhnt-Ontologiesprachen.pdf> (10. 06 2011)

[35] Lackes, R., & Siepermann, M.: Datenmodellierung, 01. 10 2010.
URL: <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Datenmanagement/Daten-/Datenmodellierung-/> (10. 06 2011)

[36] Lindenthal, J.: Kontrollierte Vokabulare – Grundlagen, Hochschule für Angewandte Wissenschaften Hamburg, 02.07.2009.
URL: <http://www.bui.haw-hamburg.de/pers/ulrike.spree/remind/vokabulare.htm#4> (10. 06 2011)

[37] Lucke, J., Geiger, C. P.: Open Data Government – Frei verfügbare Daten des öffentlichen Sektors – Gutachten zur T-City Friedrichshafen, 03.12.2010.
URL:<http://www.zeppelin-university.de/deutsch/lehrstuehle/ticc/TICC-101203-OpenGovernmentData-V1.pdf> (10. 06 2011)

[38] Lülfing, D., Benkert, H., & Siebert, S.: *95. Deutscher Bibliothekartag in Dresden 2006: Netzwerk Bibliothek*. Frankfurt am Main: Vittorio Klostermann, 2007.

[39] Mackie, J. L.: *The Cement of the Universe*. Oxford: Clarendon Press, 1980.

[40] Magliano, J. P., Pillow, B.H.: Causal Reasoning, in: Guthrie, J. W.: *Encyclopedia of education*, Vol. 1, New York [u.a.]: Macmillan Reference USA [u.a.], 2003, S. 1425-1427. URL: <http://groups.psych.northwestern.edu/gentner/papers/GentnerLoewenstein02a.pdf> (10. 06 2011)

[41] Menne-Haritz, A.: EAD in Germany. Bundesarchiv, Berlin, 29.08.2008. URL: <http://www.archivists.org/publications/proceedings/EAD@10/MenneHaritz-EAD@10.pdf> (10.06.2011)

[42] Menne-Haritz, A. & Löbnitz, A. (übers.): Encoded Archival Description Tag-Library, Version 2002, Berlin, 2006. URL: <http://www.bundesarchiv.de/imperia/md/content/daofind/ead_tag_library.pdf> (10.06. 2011)

[43] Pellegrini, T.: *Semantic Web Wege zur vernetzten Wissensgesellschaft, mit 4 Tabellen*. Berlin, Heidelberg, New York: Springer, 2006.

[44] Rowley, J. E. & Hartley, R. J.: Organizing knowledge: an introduction to managing access to information, Aldershot, Hants, England; Burlington, VT: Ashgate, 2007.

[45] Schweibenz, W., & Sieglerschmidt, J. K.: Aktuelle Entwicklungen bei Kultur-Portalen: BAM-Portal, Deutsche Digitale Bibliothek und Europeana, 2010. URL:<http://opus.bsz-bw.de/swop/volltexte/2010/834/pdf/Schweibenz_Aktuelle_Entwicklungen_bei_Kultur_Portalen.pdf> (10. 06 2011)

[46] Stalmann K. & Budde R.: Projekt Deutsche Digitale Bibliothek (DDB): Grobkonzept für das Portal der Deutschen Digitalen Bibliothek, Version 2.0, 12.08.2010. URL: <http://www.iais.fraunhofer.de/uploads/media/DDBGrobkonzeptFinal.pdf> (16.07.2011)

[47] Stock W. G. & Stock, M.: *Wissensrepräsentation: Informationen auswerten und bereitstellen.* München: Oldenbourg, 2008.

[48] Stuckenschmidt, H.: *Ontologien : Konzepte, Technologien und Anwendungen*. Berlin: Springer, 2010.

[49] Thaller, M.: Retrospektive Digitalisierung von Bibliotheksbeständen – Evaluierungsbericht über einen Förderschwerpunkt der DFG, 2005. URL:<http://www.dfg.de/forschungsfoerderung/wissenschaftliche_infrastruktur/lis/download/retro_digitalisierung_eval_050406.pdf> (10. 06 2011)

[50] Vossen, G.: *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme.* München, Wien: Oldenbourg Wissenschaftsverlag GmbH, 2008.

[51] Wiborny, W.: *Datenmodellierung, CASE, Datenmanagement.* Bonn, München: Reading, Mass. [u.a.]: Addison-Wesley, 1991.

[52] Will, M. O.: *Aufbau und Nutzung einer digitalen Bibliothek in einer universitären Ausbildungsumgebung*. Münster [u.a.] : Waxmann: Univ., Diss--Freiburg (Breisgau), 1991.

---

[53] The CIDOC Conceptual Reference Model: <http://www.cidoc-crm.org/> (10.06.2011)

[54] Dublin Core Metadata Initiative: <http://dublincore.org/> (10.06.2011)

[55] EAD: Encoded Archival Description Version 2002 Official Site:
< http://www.loc.gov/ead/> (19.06.2011)

[56] Extensible Markup Language (XML), W3C Recommendation vom 26.11.2008:
<http://www.w3.org/TR/2008/REC-xml-20081126> (10.06.2011)

[57] LIDO Terminology: <http://lido.vocnet.org/lidoTerminologySearch.php> (19.06.2011)

[58] RDF Primer, W3C Recommendation vom 10.02.2004:
<http://www.w3.org/TR/rdf-syntax/> (10.06.2011)

[59] Semantic Web Technologies: <http://www.w3.org/standards/semanticweb/> (10.06.2011)

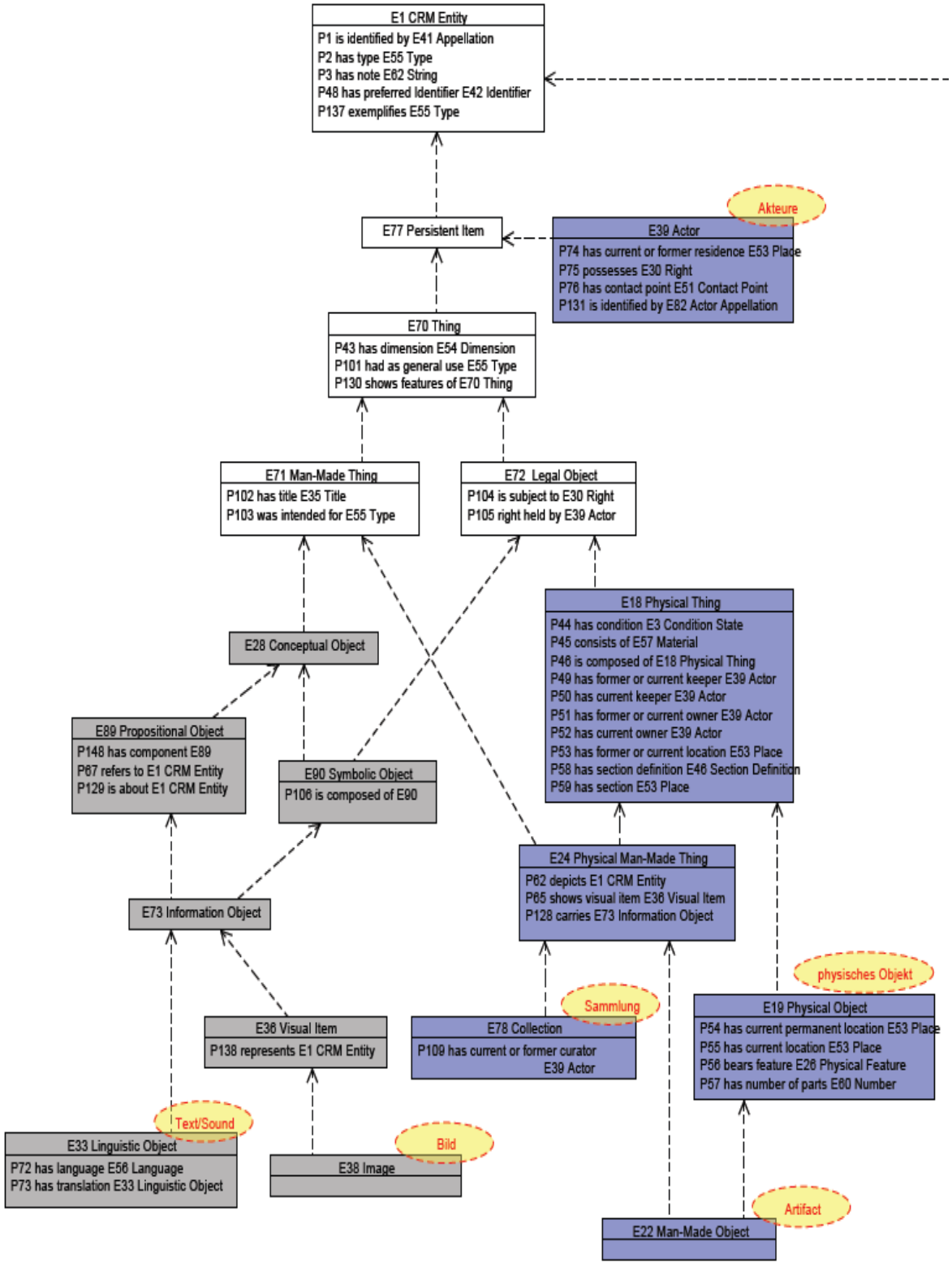
[60] SPARQL Query Language for RDF, W3C Recommendation vom 15.01.2008:
<http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/> (10.06.2011)

[61] Web Ontology Language (OWL), W3C Recommendation vom 10.02.2004:
<http://www.w3.org/TR/owl-features/> (10.06.2011)

[62] XSL Transformations (XSLT) Version 2.0, W3C Recommendation vom 23.01.2007:
<http://www.w3.org/TR/2007/REC-xslt20-20070123/> (10.06.2011)

[63] Vocabularies: <http://www.w3.org/standards/semanticweb/ontology> (19.06.2011)

7 Anhang A: Visualisierung der für das Mapping relevante CRM-Klassen und Properties (ausgenommen von Events)



Anmerkung: Grau= immaterielle Objekte, Blau= materielle Objekte

8 Anhang B: Mapping-Beispiele

<dc:title>La Pesca Lungo Le Coste Orientali Dell'Adria</dc:title>
<dc:creator>De Marchesetti, Carlo</dc:creator>
<dc:subject>Zoologica</dc:subject>
<dc:publisher>Hermanstorfer</dc:publisher>
<dc:date>1882</dc:date>
<dc:type>Monograph</dc:type>
<dc:type>Text</dc:type>
<dc:format>image/jpeg</dc:format>
<dc:format>application/pdf</dc:format>
<dc:language>deutsch</dc:language>
<dc:identifier>http://resolver.sub.uni-goettingen.de/purl?PPN64048204X</dc:identifier>
<dc:rights>Open Access</dc:rights>

Abbildung B1: Dublin Core Metadatenbeispiel

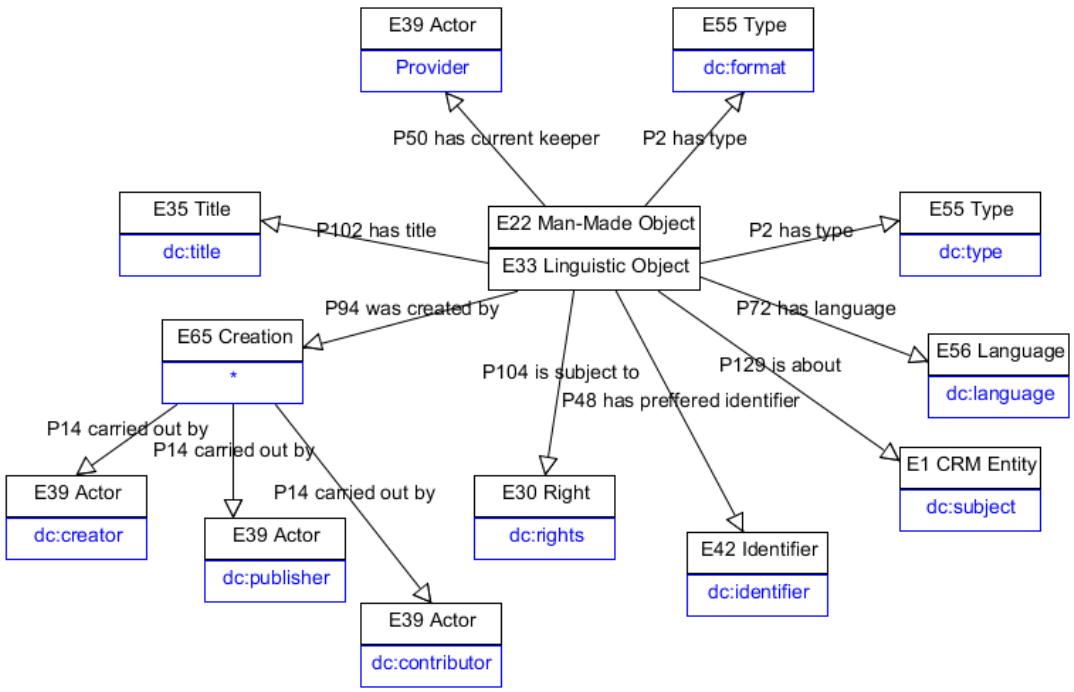


Abbildung B2: Mapping-Schema von Dublin Core Elementen

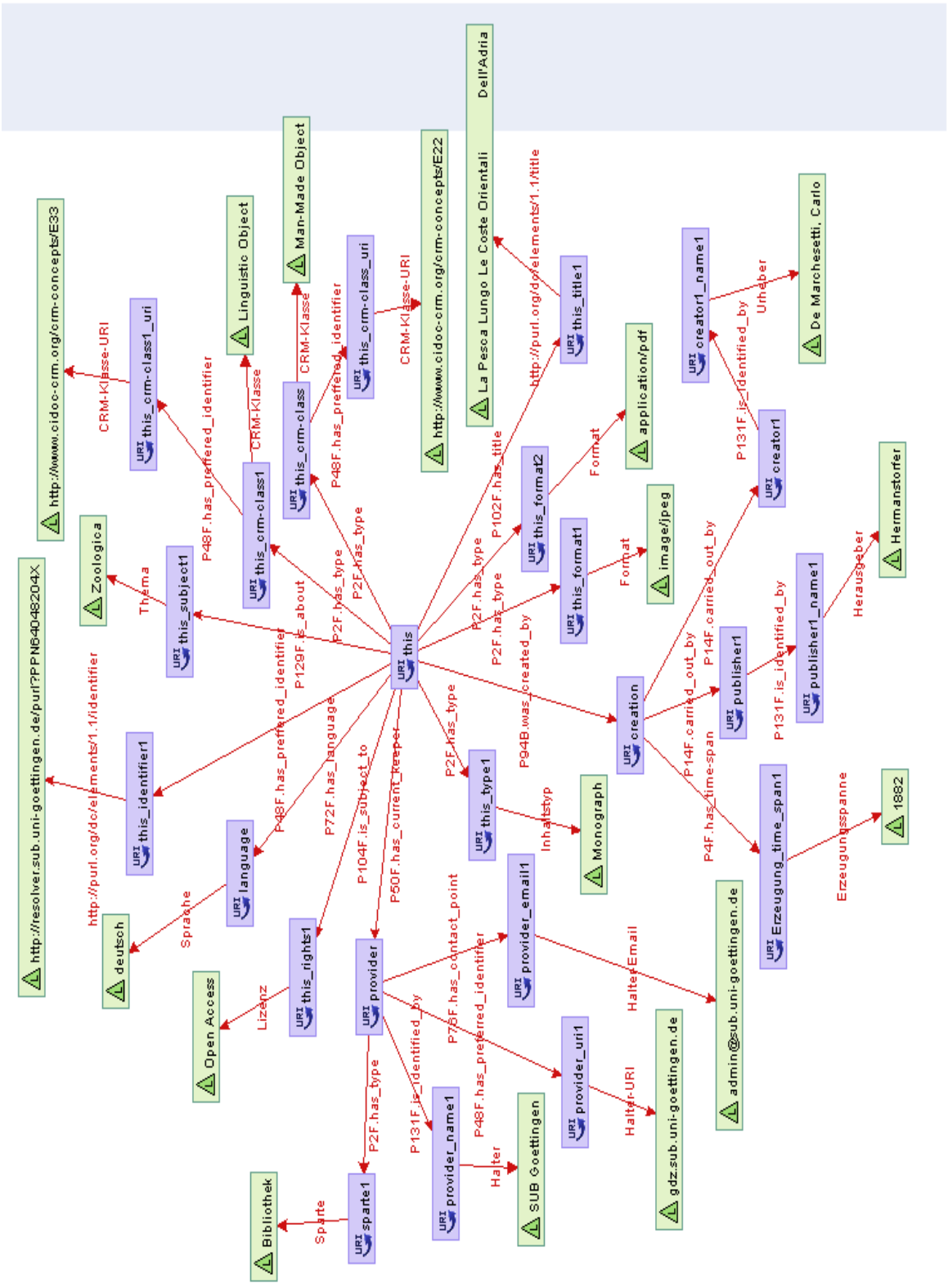


Abbildung B3: Das Mapping-Ergebnis

9 Anhang C: Clusterhierarchie

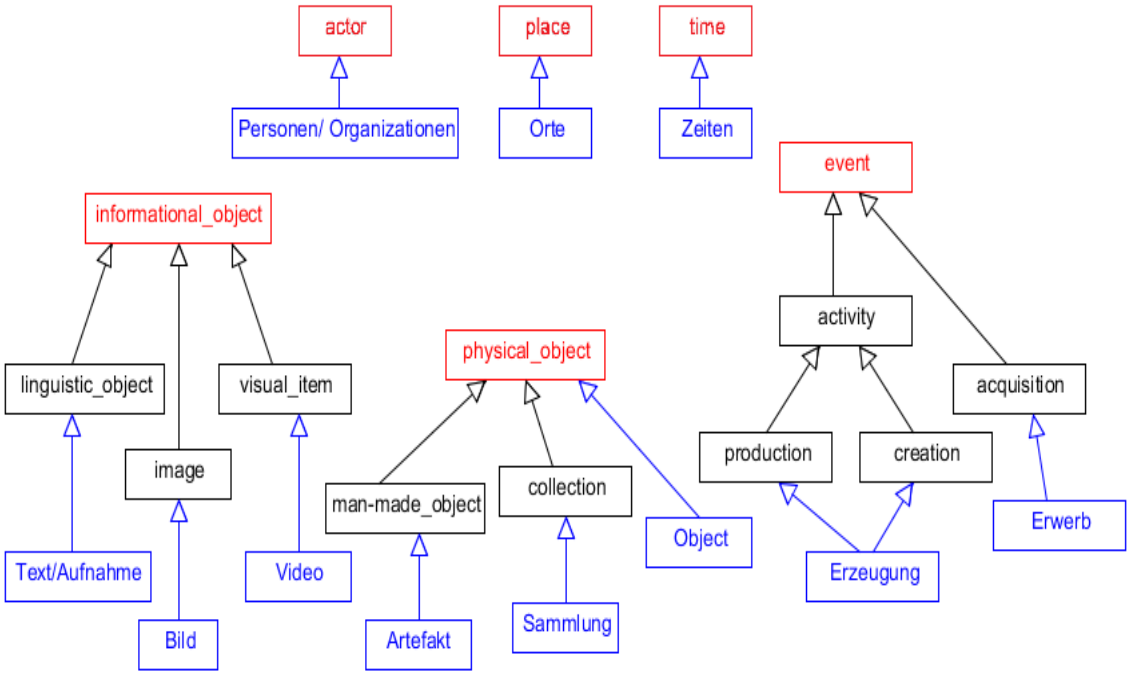


Abbildung C1: Clusterhierarchie

Anmerkung: Blau = konkrete Objekte, die von unterschiedlichen Metadaten beschrieben werden
Schwarz = allgemeine Templates
Rot = Basistemplates

10 Anhang D: Templates

```
<-- actor -->
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
                xmlns:crm="http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.2_english_label.rdfs#"
                xmlns:dc="http://purl.org/dc/elements/1.1/">


        <xsl:template name="actor">
                        <xsl:param name="id" required="yes"/>
                        <xsl:param name="address"/>
                        <xsl:param name="role" required="yes"/>
                        <xsl:param name="uri"/>
                        <xsl:param name="swd"/>
                        <xsl:param name="pnd"/>
                        <xsl:param name="partner_identifier"/>
                        <xsl:param name="name"/>
                        <xsl:param name="alternative_name"/>
                        <xsl:param name="nationality"/>
                        <xsl:param name="birthDate"/>
                        <xsl:param name="deathDate"/>
                        <xsl:param name="url"/>
                        <xsl:param name="email"/>



<!-- Partner_id of the actor-->
        <xsl:if test="not($uri='')">
            <xsl:for-each select="$uri">
                <xsl:call-template name="rdf_serializer">
                        <xsl:with-param name="domain_id" select="$id"/>
                  <xsl:with-param name="crm_property"  select="'crm:P48F.has_preferred_identifier'"/>
                  <xsl:with-param name="nexus_id" select="concat($id,'_uri', position())" />
                  <xsl:with-param name="property" select="concat($role,'-URI' )" />
                  <xsl:with-param name="value" select="."/>
                  <xsl:with-param name="datatype" select="'URI'" />
                 </xsl:call-template>
           </xsl:for-each>
        </xsl:if>


<!-- SWD of the actor-->
        <xsl:if test="not($swd='')">
           <xsl:for-each select="$swd">
                <xsl:call-template name="rdf_serializer">
                  <xsl:with-param name="domain_id" select="$id"/>
                   <xsl:with-param name="crm_property" select="'crm:P48F.has_preferred_identifier'"/>
                   <xsl:with-param name="nexus_id" select="concat($id,'_swd', position())" />
                   <xsl:with-param name="property" select="concat($role,'-SWD' )" />
                   <xsl:with-param name="value" select="."/>
                   <xsl:with-param name="datatype" select="'URI'" />
                </xsl:call-template>
             </xsl:for-each>
           </xsl:if>



<!-- PND of the actor-->
        <xsl:if test="not($pnd='')">
           <xsl:for-each select="$pnd">
                <xsl:call-template name="rdf_serializer">
                  <xsl:with-param name="domain_id" select="$id"/>
                  <xsl:with-param name="crm_property" select="'crm:P48F.has_preferred_identifier'"/>
                  <xsl:with-param name="nexus_id" select="concat($id,'_pnd', position())" />
                  <xsl:with-param name="property" select="concat($role,'-PND' )" />
```

```
                    <xsl:with-param name="value" select "."/>
                    <xsl:with-param name="datatype" select="'URI'" />
                    </xsl:call-template>
              </xsl:for-each>
            </xsl:if>


<!-- Partner_id of the actor-->
            <xsl:if test="not($partner_identifier='')">
                <xsl:for-each select="$partner_identifier">
                  <xsl:call-template name="rdf_serializer_ns">
                      <xsl:with-param name="domain_id" select="$id"/>
                    <xsl:with-param name="crm_property" select="'crm:P48F.has_preferred_identifier'"/>
                    <xsl:with-param name="nexus_id" select="concat($id,'identifier', position())" />
                    <xsl:with-param name="property" select="'dc:identifier'" />
                    <xsl:with-param name="value" select "."/>
                    <xsl:with-param name="datatype" select="'URI'" />
                    </xsl:call-template>
                </xsl:for-each>
            </xsl:if>


<!-- email of the actor-->
            <xsl:if test="not($email='')">
                 <xsl:for-each select="$email">
                   <xsl:call-template name="rdf_serializer">
                      <xsl:with-param name="domain_id" select="$id"/>
                      <xsl:with-param name="crm_property" select="'crm:P76F.has_contact_point'"/>
                      <xsl:with-param name="nexus_id" select="concat($id,'_email', position())" />
                      <xsl:with-param name="property" select="concat($role, '-Email')" />
                      <xsl:with-param name="value" select "."/>
                      <xsl:with-param name="datatype" select="'String'" />
                     </xsl:call-template>
                  </xsl:for-each>
            </xsl:if>


<!-- URL of the actor-->
            <xsl:if test="not($url='')">
                <xsl:call-template name="rdf_serializer">
                      <xsl:with-param name="domain_id" select="$id"/>
                      <xsl:with-param name="crm_property" select="'crm:P76F.has_contact_point'"/>
                      <xsl:with-param name="nexus_id" select="concat($id,'_homepage')" />
                      <xsl:with-param name="property" select="concat($role, '-Homepage')"/>
                      <xsl:with-param name="datatype" select="'URL'"/>
                      <xsl:with-param name="value" select ="$url"/>
                      <xsl:with-param name="datatype" select="'URL'" />
                </xsl:call-template>
            </xsl:if>


<!-- Birth of the actor-->

<xsl:if test="not($birthDate='')">
            <xsl:call-template name="rdf_serializer">
                      <xsl:with-param name="domain_id" select="$id"/>
                      <xsl:with-param name="crm_property" select="'crm:P98B.was_born'"/>
                      <xsl:with-param name="nexus_id" select="concat($id,'_birth')" />
                      <xsl:with-param name="property" select="concat($role, '-Geburtsdatum')"/>
                      <xsl:with-param name="datatype" select="'Date'"/>
                      <xsl:with-param name="value" select ="$birthDate"/>
                   </xsl:call-template>
                  </xsl:if>


<!-- Death of the actor-->
<xsl:if test="not($deathDate='')">
            <xsl:call-template name="rdf_serializer">
```

```
                    <xsl:with-param name="domain_id" select="$id"/>
                    <xsl:with-param name="crm_property" select="'crm:P100F.died_in'"/>
                    <xsl:with-param name="nexus_id" select="concat($id,'_death')" />
                    <xsl:with-param name="property" select="concat($role, '-Todesdatum')"/>
                    <xsl:with-param name="datatype" select="'Date'"/>
                    <xsl:with-param name="value" select ="$deathDate"/>
                </xsl:call-template>
            </xsl:if>


<!-- Address of the actor-->
<xsl:if test="not($address='')">
        <xsl:call-template name="rdf_serializer">
                    <xsl:with-param name="domain_id" select="$id"/>
                    <xsl:with-param name="crm_property" select="'crm:P74F.has_current_or_former_residence'"/>
                    <xsl:with-param name="nexus_id" select="concat($id,'-address')" />
                    <xsl:with-param name="property" select="concat($role, '-Adresse')"/>
                    <xsl:with-param name="datatype" select="'String'"/>
                    <xsl:with-param name="value" select ="$address"/>
            </xsl:call-template>
        </xsl:if>


<!-- Name of the actor-->
<xsl:if test="not($name='')">
        <xsl:for-each select="$name">
            <xsl:call-template name="rdf_serializer">
                <xsl:with-param name="domain_id" select="$id"/>
                <xsl:with-param name="crm_property" select="'crm:P131F.is_identified_by'"/>
                <xsl:with-param name="nexus_id" select="concat($id,'_name', position())" />
                <xsl:with-param name="property" select="$role"/>
                <xsl:with-param name="value" select ="."/>
                <xsl:with-param name="datatype" select="'String'" />
              </xsl:call-template>
            </xsl:for-each>
          </xsl:if>


 <!--alternative name of the actor-->
<xsl:if test="not($alternative_name='')">
        <xsl:for-each select="$alternative_name">
          <xsl:call-template name="rdf_serializer">
                    <xsl:with-param name="domain_id" select="$id"/>
                    <xsl:with-param name="crm_property" select="'crm:P131F.is_identified_by'"/>
                    <xsl:with-param name="nexus_id" select="concat($id,'_alternative_name', position())" />
                    <xsl:with-param name="property" select="concat($role, '-Alias')"/>
                    <xsl:with-param name="value" select ="."/>
                    <xsl:with-param name="datatype" select="'String'" />
            </xsl:call-template>
           </xsl:for-each>
        </xsl:if>


<!--nationality of the actor-->
<xsl:if test="not($nationality='')">
        <xsl:for-each select="$nationality">
            <xsl:call-template name="rdf_serializer">
                <xsl:with-param name="domain_id" select="$id"/>
                <xsl:with-param name="crm_property"  select="'crm:P107B.is_current_or_former_member_of'"/>
                <xsl:with-param name="nexus_id" select="concat($id,'_nationality', position())" />
                <xsl:with-param name="property" select="concat($role, '-Nationality')"/>
                <xsl:with-param name="value" select ="."/>
                <xsl:with-param name="datatype" select="'String'" />
            </xsl:call-template>
          </xsl:for-each>
        </xsl:if>
 </xsl:template>
</xsl:stylesheet>
```

11 Erklärung

Hiermit versichere ich, dass ich diese Magisterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht.

Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen.

_____

Unterschrift