

**Projekt "Deutsche Digitale Bibliothek"**  
**Rationale zur IAIS CORTEX Konzeption, den  
Datenmodellen und Mappings**

Dr. Kai Stalman

Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS  
Schloss Birlinghoven  
53757 Sankt Augustin

kai.stalman@iais.fraunhofer.de

(Stand: 21. Dezember 2011)

## **1 Einleitung**

Das Papier begründet die in IAIS CORTEX als technische Plattform der Deutschen Digitalen Bibliothek (DDB) zugrunde liegenden wesentlichen Konzepte. Das vom Beauftragten für Kultur und Medien der Bundesrepublik Deutschland geförderte System wird zum Winter 2011 termingerecht in einer betriebsbereiten ersten Ausbaustufe fertig gestellt. Der Name der Deutschen Digitalen Bibliothek wird derzeit (Dezember 2011) im Kompetenznetzwerk der DDB (KNW) erneut diskutiert und ist daher wieder als vorläufig anzusehen.

IAIS CORTEX sowie die auf CORTEX zugreifende Präsentationsschicht der ersten Aufbaustufe der DDB wurde von Fraunhofer IAIS und Technologie-Partnern (ECM, Arlanis, Fraunhofer IESE, FIZ Karlsruhe, SHI, Neofonie) in der Zeit zwischen Mai 2010 und November 2011 konzipiert, entwickelt, auf einem Rechnercluster bei IAIS implementiert und mehrfach testweise mit den seit Herbst 2011 verfügbaren stehenden ca. 1.7 Millionen Objekten aus Bibliotheken, Mediatheken, Museen, Archiven gespeist. Parallel wird das System im Rechenzentrum des FIZ Karlsruhe implementiert, wo es Anfang 2012 mit zunächst bis zu 6 Mio. Objekten in den Pilotbetrieb gehen könnte, wenn bis dahin im KNW eine Einigung zur Datenüberlassung erzielt worden sein sollte.

Von konkreten Technologiedetails, die in der IAIS CORTEX Dokumentation und in den Javadocs beschrieben sind, wird hier abstrahiert. Ebenso bleiben alle Aspekte der Contenterschließung und der Organisation der Ingests und der weiteren Mappings außer Betracht, die seitens des KNW bisher noch nicht geklärt wurden.

Ziel des Papiers ist es lediglich, die technische Plattform in ihrer Funktionsweise zu erklären, und die Grundannahmen, die zu dem vorliegenden Design der Plattform führten, zu vermitteln.

## 2 Grundannahmen

IAIS CORTEX ist als webbasierte Knowledge Management Plattform für in beliebigen Metadatenformaten ausgezeichnete Objekte praktisch beliebiger Provenienz gedacht. Ein nach IAIS CORTEX ingestierbares Objekt muss, um später zugreifbar zu sein, lediglich einen Identifier besitzen. Ein Objekt der DDB muss dafür eine Herkunftskennung (Halter, Lieferant) und eine beliebige, providerseitig eindeutige Objekt-Kennung (z.B. Inventurnummer oder Signatur) haben. Um im Index findbar zu sein, muss ein Objekt zudem über mindestens ein beliebiges, belegtes deskriptives Metadatenfeld oder einen deskriptiven Text verfügen. Zusätzlich zu den genannten minimalen, deskriptiven Informationen über ein Objekt kann IAIS CORTEX weitere Informationen auswerten und nutzen. Hierzu zählen in erster Linie weitere (tiefere, reichere) deskriptive Metadaten, des Weiteren die oft als administrativ bezeichneten Metadaten (beispielsweise zu Rechten und Lizenzen), sowie einerseits strukturelle Informationen zu möglicherweise enthaltenen Binärdaten und binäre Daten selbst, also z.B. digitalisierte Surrogate von physischen Objekten wie Büchern, Skulpturen, Bildern usw., die entweder ebenfalls in die Plattform ingestiert werden können, oder über Links aus entfernten Quellen im Web angebunden sind.

Wir sind von einigen weiteren Prämissen ausgegangen, von denen wir die Wesentlichen hier aufführen:

1. Die Daten der DDB stammen aus verschiedenen Sparten und von sehr unterschiedlichen Einrichtungen. Auch innerhalb einer Sparte unterscheiden sich die Einrichtungen sehr, z.B. hinsichtlich ihrer Größe, ihrer technischen Möglichkeiten, ihrer Ziele und Aufgaben. Die DDB soll nicht so ausgelegt werden, dass einzelne Einrichtungen oder Sparten auf Kosten anderer bevorzugt werden.
2. Die Plattform muss daher vor allem mit äußerst heterogenen Metadatenformaten umgehen können. Die Unterschiede zwischen den Sparten sind immens. Einfache Übernahmen von spartenspezifischen Datenmodellen oder spartenspezifischen best practises verbieten sich daher für ein spartenübergreifendes Projekt wie die DDB: Was für eine Sparte ein plausibles Mittel der Wahl sein kann, ist für andere Sparten unpraktikabel oder sinnlos.

3. Manche Objekte, bzw. Objekte mancher Einrichtungen können Normdatenbezüge enthalten, andere können binäre Inhalte enthalten, manche stammen aus Datenhaltungssystemen, die nur für den Zweck der Speicherung der jeweiligen Daten entworfen wurden, andere liegen in Austauschformaten vor, die milliardenfach verwendet werden.
4. Die Datenmodelle der spartenspezifischen Daten, die von den Providern geliefert werden, müssen innerhalb der DDB vereinheitlicht werden, allerdings nicht so, dass eine vollständige Replikation der schon in den Originaldaten abgebildeten Semantik in einem weiteren Modell erfolgt, sondern zweckorientiert. (Die Originalmetadaten bleiben innerhalb der DDB stets unverändert erhalten und können prinzipiell ebenso abgefragt werden wie andere Repräsentationen der Objekte innerhalb der DDB auch.)
5. Die Zwecke der Abbildung der Originaldaten in Modelle der DDB sind folgende:
  - a. Retrieval der Objekte über die Suchmaschine innerhalb der DDB
  - b. Navigation zwischen Objekten innerhalb der DDB über Spartengrenzen hinweg
  - c. Visuelle Repräsentation der Objekte innerhalb der Präsentationsschicht der DDB, bzw. Auslieferung geeigneter Views der Objekte auf anderen Plattformen.
  - d. Repräsentation der Semantik der Objekte in einer für menschliche Wahrnehmung geeigneten Form (grafisch oder textuell)
  - e. Repräsentation der Semantik der Objekte in einer für maschinelle Verarbeitung geeigneten Form (z.B. als RDF)
  - f. Export der Objekte, einer Repräsentation der Objekte oder von Ausschnitten der Objekte bzw. ihrer Repräsentation
6. Die Datenmenge kann sehr groß werden, daher muss die Plattform gut skalieren.
7. Das System muss so ausgelegt werden können, dass es ausfallsicher ist.
8. Die ingestierten Daten sollen zusätzlich gesichert werden können, so dass bei einem Totalverlust der Plattforminstanzen eine Rekonstruktion durch erneuten Ingest aller Objekte möglich ist.

9. Datenmodelle, Mappings, Indexierung, Facetten, Views sollen als Konfigurationen des Kernsystems und nicht als konstitutive Elemente des Systems verstanden werden können. Mit jeweils unterschiedlichem Aufwand sind alle diese Aspekte des Systems austauschbar, womit sich die Plattform grundsätzlich von solchen Systemen unterscheidet, in denen die Geschäftsmodelle in einem Datenschema festgelegt sind und Änderungen am Schema Änderungen in allen Schichten nachziehen. Für einige der Komponenten existieren bereits verschiedene alternative Bindungen.
10. Präsentationsschicht und Fremdsysteme sollen leicht über das API angebunden werden können.
11. Eine Personalisierbarkeit der Plattform ist ausdrücklich erwünscht, personenbezogene Daten sollen jedoch nicht im Kernsystem vorgehalten werden.
12. Objektbestandteile (im Allgemeinen binaries) sollen flexibel gegen Zugriff geschützt werden können (regelbasierter Zugriffskontrollmechanismus).
13. Das System soll ein möglichst detailliertes Monitoring erlauben.
14. Da sich mit der Zeit technische Möglichkeiten und Anforderungen ändern werden, sollte die Architektur dem Austausch von Komponenten nicht im Wege stehen.
15. Wie bei der Beauftragung ausdrücklich eingefordert, sollte die technische Plattform der DDB von IAIS für andere Projekte und Zwecke nachgenutzt werden können.
16. IAIS CORTEX soll einerseits als Open Source / Free Software einer breiten Entwickler-Community zugänglich gemacht werden können (es wurden als Kernkomponenten daher nur echte Open Source Frameworks verwendet), andererseits soll die Software bei IAIS zentral weiter gewartet und ausgebaut werden.

### **3 Funktionale Beschreibung und konzeptionelle Architektur des Systems "Deutsche Digitale Bibliothek"**

Das System DDB im weiteren Sinne schließt Datenhaltungssysteme der Einrichtungen ein. Diese üblicherweise lokal in den Einrichtungen laufenden Systeme werden gegenwärtig aber nicht direkt an die DDB angebunden. Statt dessen sollen die Daten zunächst exportiert werden z.B. als records, XML container, in Einzelfällen auch als Datenbank-Dumps bzw. CSV Dateien. Die Übertragung der Daten in die DDB im engeren Sinne kann über eine standardisierte Harvesting-Schnittstelle oder auf irgendeinem anderen Weg erfolgen (FTP, Datenträger).

Die exportierten Daten können in praktisch beliebigen Formaten vorliegen. IAIS stellt ein Datenaufbereitungswerkzeug (den Augmented SIP Creator, ASC) in verschiedenen Ausprägungen bereit. Der ASC ist Teil der IAIS CORTEX Plattform und hat die Aufgabe, die heterogenen Originaldaten für die DDB aufzubereiten. Die Aufbereitung findet in einem mehrteiligem Prozess statt, der im Wesentlichen eine Reihe von Transformatoren (XSLT Skripte) aufruft, die ihrerseits in einer Library organisiert sind. Diese Library besteht aus metadatenspezifischen und allgemeinen Komponenten und ist leicht erweiterbar. Das Ergebnis des Transformationsprozesses eines Objekts ist ein Container (das SIP oder Submission Information Package), das die originalen und DDB-spezifischen Datenmodelle sowie statische Views und andere Informationen enthält. Ein SIP entspricht also einem Objekt und ist weitestgehend self-contained. Bei manchen Originalmetadaten-Files geht der Transformation ein Splitting voraus. Bei Lido oder Marc sind in einem File typischerweise eine Menge von Objekten (bzw. records) enthalten. Bei EAD splitten wir die Files ebenfalls, um sinnvoll handhabbare Einheiten zu erzeugen: einerseits auf Unit/File-Ebene, andererseits auf Collection-Ebene.

Eine Direktanbindung an die lokalen Datenhaltungssysteme ist allerdings durchaus möglich und sinnvoll, und wurde von einem der beteiligten Partner (Arlanis) prototypisch auf Grundlage einer alternativen Implementierung des ASC vorbereitet.

Binäre Daten können vom ASC ebenfalls in das SIP gepackt werden. Dazu müssen die Binärdaten in einer technisch vorgegebenen (aber leicht an andere Bedürfnisse anpassbaren) Weise lokal abgelegt werden.

Der Betrieb des ASC erfolgt nach unseren Vorstellungen zweckmäßiger Weise beim Betreiber, dem FIZ Karlsruhe. Ein Betrieb beim Provider ist technisch möglich, unserer Ansicht nach aber zu riskant. Um dem Provider dennoch eine Möglichkeit an die Hand zu geben, seine Daten testweise zu transformieren und ingestieren, kann die DDB Plattform inklusive des ASC auf einen normalen Desktop PC installiert werden (minimaler Installationsaufwand, keine Abhängigkeiten zu Datenbanken oder Fremdsystemen, lauffähig auf allen üblichen Plattformen wie Mac, Windows, Linux). Die lokal beim Provider lauffähige IAIS CORTEX Variante trägt den Namen MyCortex. Der Provider kann damit ohne Weiteres seine hauseigenen Daten probeweise in eine lokale DDB ingestieren, er kann gefahrlos mit Mapping-Alternativen experimentieren, und seine eigenen Daten vor einer Freigabe zum Ingest beim FIZ qualitätssichern.

Die idealerweise beim FIZ betriebenen ASC Instanzen, die Daten für das Produktivsystem aufbereiten, übertragen die SIPs über ein Messaging-System an das Kernsystem. Normalerweise wird ein größeres Set von Objekten auf einmal übertragen. Diese Sammelübertragung wird logisch von einer Kennung (Zeitstempel) zusammengehalten. Die Übertragung erfolgt dann einzeln, SIP-weise und ausfallssicher, und kann parallel auf mehrere Queues, sowie auf mehrere Ingestoren erfolgen. Die Übertragung eines SIP leitet den Ingest des SIPs in die Plattform ein. Übertragung und Ingest-Vorgang können auf der Plattform bezüglich Erfolg und Performanz von einem zentralen Monitoring-System überwacht werden. Bei Unregelmäßigkeiten werden Nachrichten an konfigurierbare Empfänger verwendet. Zu jedem Ingest kann außerdem jederzeit ein Report gezogen werden, der den Stand des Ingests zusammenfasst. Zudem werden die auf der Plattform im Ingest-Service ankommenden SIPs per Multiplexer auf Wunsch an andere Silos geleitet. Nach aktueller Planung will das FIZ die Messages erstens auf zwei gleichartig ausgestattete, ingestierende Systeme leiten und zudem über einen weiteren Kanal auf ein Sicherungslaufwerk leiten.

Beim Ingest werden die einzelnen SIPs geprüft, analysiert und verarbeitet. Nach der Verarbeitung liegt das SIP (als Archival Information Package, AIP) erstens im Datenrepository der Plattform. Von dort können Bestandteile des Objekts (views, model, Originalmetadaten, binaries) über den beim Ingest vergebenen Identifier des Objekts und eine Pfadkomponente gezogen werden. Zweitens sind alle in den Metadaten enthaltenen Terme (sowie Aliasse und Expansionen der Terme) in einem Volltextindex indexiert und mit dem Objekt (dem Identifier) verbunden. Drittens werden aus einem Teil eines jeden internen Objektmodells ein Facettensatz generiert, der ebenfalls mit dem Objekt assoziiert ist, und viertens wird das Objekt bei Ingest in ein sich ausbildendes Gesamt-Objekt-Netz eingebunden.

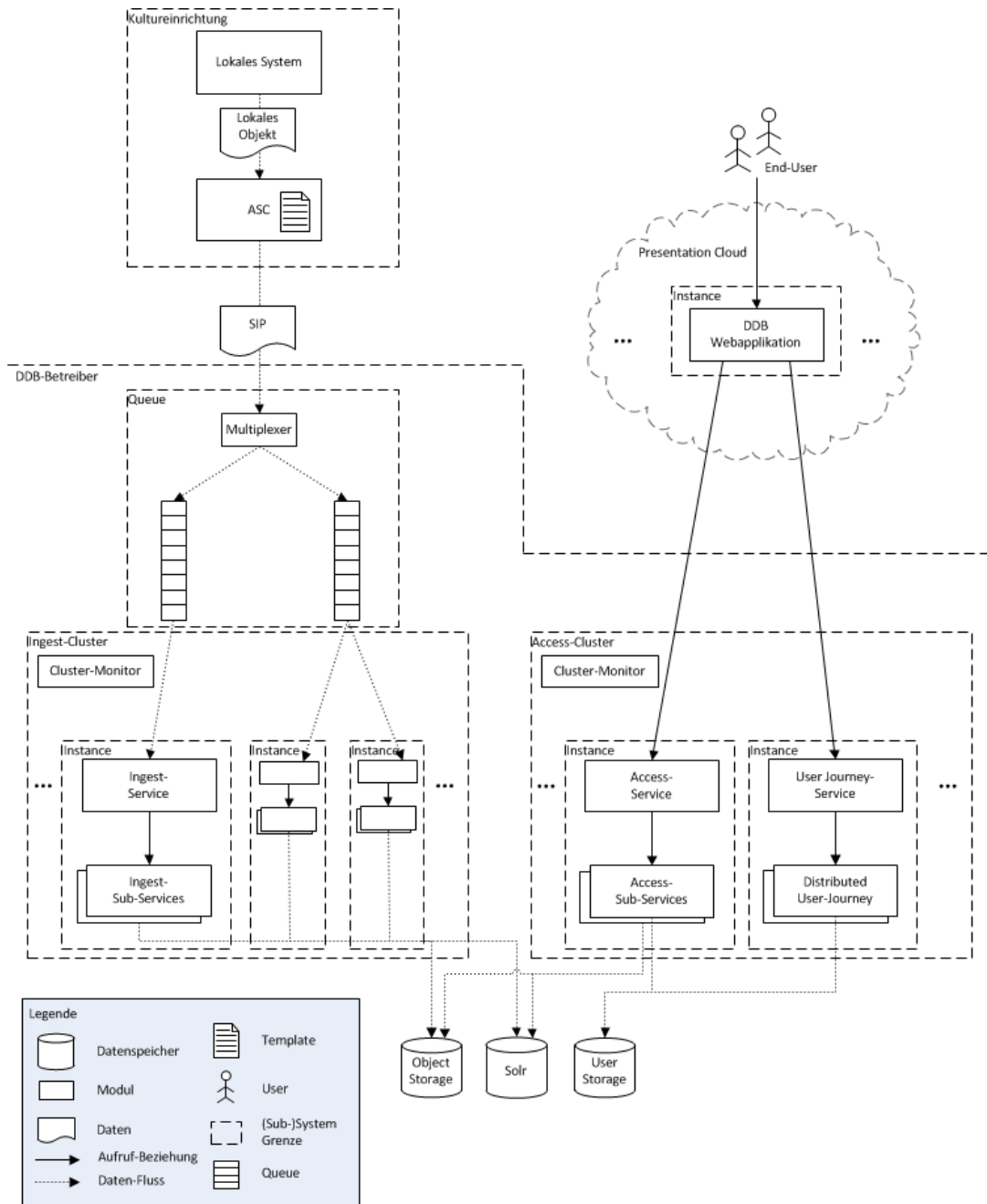


Abb. 1: Konzeptionelle Architektur, aus [IESE].

Der Zugriff des Nutzers auf ein Objekt verläuft üblicherweise so, dass zunächst die Suchmaschine (der Index) befragt wird (Eingabe von Suchtermen), dann über die Facetten die Treffermenge reduziert wird. Erst wenn der Nutzer ein einzelnes Objekt selektiert, wird über das API das Repositorium befragt und ein Datenfragment des AIP geholt (XML, Stream, HTML, RDF etc).

Da die Objekte untereinander im sogenannten Nodestore in verschiedener Weise assoziiert sind, kann man außerdem von einem Objekt zu anderen Objekten navigieren. Diese Navigationsfunktionen sind wie alle anderen Zugriffsoperationen Teil des API.

Ist der Nutzer angemeldet, werden alle das Suchen, Navigieren und Zugreifen betreffenden Benutzeraktionen in einer 'Journey' gespeichert. Die Objektzugriffe können wahlweise außerdem zu Kollektionen zusammengefasst werden.

#### **4 Grundsätzliche Überlegungen zu Fragen der Datenmodelle, der Mappings und zur Bildung der Facetten**

Der verbreitete Ansatz zur datentechnischen Abbildung von heterogenen Metadaten beruht auf der Annahme, dass es möglich sei, einen gemeinsamen Nenner an Zielfeldern zu definieren, auf die dann die Felder der Eingangsmodelle gemappt werden.

Dieser ebenso einfache wie naive Ansatz scheitert seit vielen Jahren in der Praxis immer wieder. Er muss zu entweder erheblicher semantischer Unschärfe oder einer absurd hohen Anzahl von Zielfeldern führen. Das prominenteste, mit diesem Ansatz gestartete Projekt ist Europeana. Europeana hat ursprünglich ein flaches, um einige Felder erweitertes Dublin Core Format etabliert (ESE) und versucht nun, nachdem unzählige Mappings für inzwischen 20 Millionen Objekte auf ESE realisiert wurden, mit EDM ein graphbasiertes Modell, dem als wesentlicher Kern das CIDOC CRM zugrunde liegt, auszurollen. Der Wechsel von ESE zu EDM kommt praktisch einem Neubau gleich, er betrifft sämtliche Schichten der Delving Plattform und erfordert, dass sämtliche ESE Mappings durch die erheblich aufwändigeren EDM Mappings ersetzt werden müssen.

Die Problematik des ursprünglichen Ansatzes kann leicht anhand der Facetten (Filter) verdeutlicht werden, die Europeana dem Nutzer gegenwärtig anbietet.

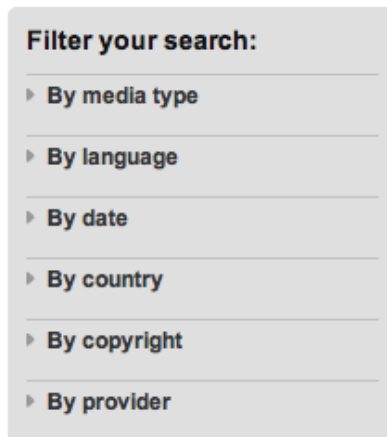


Abb. 2: Kompletter Satz der Europeana Facetten auf Basis von ESE [EURO]

*Media Type* hat die Ausprägungen IMAGE, TEXT, SOUND, VIDEO, wobei IMAGE auf mehr als 13 Mio. Objekte matcht, VIDEO (als kleinste Treffermenge) noch auf mehr als 500.000 Objekte. Somit fragt sich, ob die Facette als Filter zur Eingrenzung einer großen Treffermenge überhaupt noch Sinn macht. Dazu kommt, dass das Selektionskriterium *Media Type* ohnehin sehr schwach ist, da der *Media Type* keinesfalls unbedingt den eigentlichen Objekttyp kennzeichnet, sondern beispielsweise ebenso das IMAGE einer Skulptur wie das IMAGE eines Bildes oder das Standbild eines Films bezeichnen kann.

Noch problematischer erscheint die Facette *Date*, die ein Datum enthält, das in *irgendeiner* und damit semantisch völlig unklaren Beziehung zum Objekt steht. Das ESE Modell und die Mappings unterscheiden nicht, ob eine Jahreszahl die Erzeugung, das Finden, Erfinden, den Verlust, die Akquise, die Restauration oder den Tod eines Objekts bedeutet, oder ob umgekehrt vielleicht das Objekt eine Aussage über die Zeit macht.

Obwohl Europeana inzwischen mit dem EDM einen sinnvollen (graphbasierten) Weg eingeschlagen hat, verstummen die Stimmen nicht, die für die DDB fordern, den Fehler zu wiederholen und ein flaches Indexingprofil mit 'wenigstens einigen wenigen obligatorischen Feldern' zu definieren, auf das man dann 'doch erst mal ganz einfach mappen könnte'.

Um es in aller Klarheit zu formulieren: es völlig irrelevant, wie dieses Profil aussähe, da das Problem prinzipieller Natur ist und vollständig zum Scheitern verurteilt ist. Um ein Beispiel zu nennen: Für Bibliotheken wäre kein Problem, eine Mindestanforderung zu erfüllen, die für jedes Objekt einen Autor und einen Titel verlangt (im Notfall mappt man dann den Herausgeber als Autor). Für Museen, Denkmalsämter und Archive geben Autor und Titel eher fragwürdige Kategorien der Wissensorganisation ab.

Ebenso wenig zielführend wäre es auch, die komplette Semantik eines LIDO oder EAD Objekts in ein Modell der DDB zu übertragen, weil viele Fakten zwar hohen dokumentarischen Wert haben, aber keine Bedeutung für das Retrieval, also die Bildung der Facetten.

Wir haben daher einen anderen Weg eingeschlagen. In der DDB beruhen Erzeugung der Facetten, Indexierung und die verschiedenen Repräsentation der Objekte auf verschiedenen Mappings und Modellen.

Jedes Objekt erhält aus dem Transformationsprozess statische (X)HTML Repräsentationen, die von Endgeräten leicht angezeigt werden können. Die Erzeugung dieser Views beruht auf Mappings, die von den Originalmetadaten ausgehen, aber beispielsweise bei EAD nur einen Bruchteil eines EAD Files (das mehrere Hundert MB groß sein kann) erfassen und sich auf unterschiedliche Ebenen innerhalb der EAD Hierarchie beziehen können. Auf eine dieser so erzeugten Views wird zugegriffen, wenn die Ergebnislisten der Suchmaschine gebildet werden, auf eine andere, wenn eine Detailansicht des Objekts dargestellt wird. Die Mapping-Library ist so aufgebaut, dass die View-Transformer (wie alle übrigen Transformer auch) nicht nur spezifisch für ein Metadatenformat, sondern vielmehr auch spezifisch für einen Provider oder eine Sammlung ausgelegt und versioniert werden kann. Auf diese Weise können Views auf spezielle Möglichkeiten und Bedürfnisse detailliert angepasst werden. Selbstverständlich können Links auf binäre Inhalte enthalten sein.

Programmiertechnisch erheblich aufwändiger sind die Transformer, die aus den Originalmetadaten in der DDB eine Repräsentation erzeugen, die der DDB als Zwischenformat für jene Datenmodelle dient, auf denen das Retrieval (Suche, Filter) der Objekte, die Navigation (Vernetzung) und die Visualisierung der Semantik beruht, und aus denen ggf. exportierbare Dritt-Formate erzeugt werden können (die Originaldaten können jederzeit ebenfalls exportiert werden).

Als Sprachraum des genannten Zwischenformats, auf das bei der Transformation im ASC alle Originalmetadaten normalisiert werden, haben wir einen ISO Standard gewählt: das CIDOC CRM, das seit vielen Jahren in zahlreichen Projekten und äußerst unterschiedlichen Wissensdomänen erfolgreich eingesetzt wird.

Das CRM verwirklicht einen äußerst eleganten gedanklichen Ansatz, da es im Kern nicht Begriffszusammenhänge darstellt, sondern Zustandsänderungen beschreibbar macht. Es ist verglichen mit anderen Modellen äußerst kompakt, weil es auf den (kaum praktikablen) Versuch verzichtet, jeder Entität einen mehr oder weniger vollständigen Satz von möglichen Attributen zuzuordnen, die dann ihrerseits zu allen möglichen Entitäten in Beziehung stehen können. Statt dessen fokussiert das CRM auf Aussagen, die retrieval-relevante Zusammenhänge konstruieren.

Die folgende Übersicht zeigt, welche fundamentalen Zusammenhänge auf Grundlage eines Mappings auf CRM konstruiert und in der Folge abfragbar gemacht werden können.

Domain (select)	Range(query parameter)				
	Thing	Actor	Place	Event	Time
Thing	2.is part of 3.is similar or the same with 4. has met 5. from 6. is origin of 8. refers to 9.is referred by	4.has met 5.from 8.refers to 9.is referred by	4.from 8.refers to 9.is referred to at	4.from 8.refers to	4.from
Actor	4.has met 6.is creator or provider of 8. refers to 9.is referred by	2.is member of 4. has met 5.has parent or founder 6.is parent or founder of 8.refers to 9.is referred by	4.has met 5.from 8.refers to 9.is referred to at	4.has met 8.refers to 6.has met	8.refers to 6.has met 4.from
Place	5.is origin of 8.refers to or is about 9.isreferred by	5.is origin of 8.refers to or is about 9.is referred by	2.is part of 5.is origin of	9.is referred by 5.is origin of	7.at
Event	5.is origin of 9.is referred by 8.refers to or is about	4.from 9.is referred by 8.refers to or is about 6.has met	8.refers to or is about 7.at	8.refers to or is about 2.is part of	8.refers to or is about 7.at
Time	5.is origin of	5.is origin of	5.is origin of	5.is origin of	2.is part of

Abb. 3: "Fundamental Categories and Fundamental Relationships" auf Grundlage des CRM, aus [FORTH]

Welche Aussagen tatsächlich im Modell erscheinen, hängt einerseits natürlich von den Daten ab, die providerseitig zur Verfügung stehen, andererseits von den Mappings.

Für einige der gängigen Metadatensprachen (DC, EAD, LIDO, MARC21, ESE, DIF, METS/MODS) haben wir uns an vorliegenden konzeptionellen Mappings orientiert und diese weiter ausgearbeitet.

Während es aus unserer Sicht empfehlenswert erscheint, das CRM zur Normalisierung der Originaldaten für das Retrieval zu verwenden, stellt sich dennoch die Frage, wie die Informationen für den Endnutzer aufbereitet werden sollten.

CRM lässt sich leicht in Form der in Mode gekommenen RDF Triple serialisieren. Sparql Queries und CRM Aussagen als RDF stellen jedoch so noch kein geeignetes Benutzerinterface dar. Das British Museum, das ebenfalls CRM einsetzt, ist dennoch diesen Weg gegangen, wie die Abbildung zeigt.



Home | Sparql | Help | Licensing | About

**Description**  
**guid:\_43d4bdd4-e491-49e9-84b0-2372597e25dc**

**Results**

Result	Binding	Value
1	Subject	guid:_43d4bdd4-e491-49e9-84b0-2372597e25dc >
	Predicate	http://collection.britishmuseum.org/id/crm/P2F.has_type >
	Object	http://collection.britishmuseum.org/id/dimension/L >
2	Subject	guid:_43d4bdd4-e491-49e9-84b0-2372597e25dc >
	Predicate	http://collection.britishmuseum.org/id/crm/P3F.has_note >
	Object	Dimension :: : a
3	Subject	guid:_43d4bdd4-e491-49e9-84b0-2372597e25dc >
	Predicate	http://collection.britishmuseum.org/id/crm/P91F.has_unit >
	Object	http://collection.britishmuseum.org/id/units/cm >

Abb. 4: British Museum, Semantic Web Collection Online: User Interface [BMU]

Für die DDB, die für die breite Masse der Internetnutzer gedacht ist, haben wir einen konventionelleren und vermutlich deutlich benutzerfreundlicheren Ansatz vorgezogen, der die in CRM abgebildeten Transitionen im modellierten Graph auf einen 'künstlichen' Namensraum projiziert, der dem Benutzer einerseits als Facetten und in gleicher Form als benannte, zu anderen Objekten führende Relationen angeboten wird.

Die Plattform schreibt beim Ingest alle in den CRM Modellen tatsächlich gefundenen Transitionen mit. Dies ist äußerst hilfreich, da es praktisch nicht vorhersehbar ist, welche der gemappten Ausprägungen des Graphen dann in den Daten in welcher Form tatsächlich anzutreffen sind. Das Logbuch der wirklich gefundenen Transitionen dient als Vorlage der Queries, die zur Ausbildung der Facetten beim Ingest ausgeführt werden.

Der folgende Ausschnitt aus der Konfigurationsdatei, die diese Queries beinhaltet, zeigt anhand zweier Beispiele (*Erzeugungszeitraum* und *Person*) wie Facetten aus verschiedenen Pfaden im Graphen gebildet werden:

```
Erzeugungszeitraum
  P108B.was_produced_by,P4F.has_time-span
Erzeugungszeitraum
  P94B.was_created_by,P4F.has_time-span
Erzeugungszeitraum
  P94B.was_created_by,P4F.has_time-span,P79F.beginning_is_qualified_by

Person
  P94B.was_created_by,P14F.carried_out_by
Person
  P94B.was_created_by,P14F.carried_out_by,P131F.is_identified_by
Person
  P108B.was_produced_by,P14F.carried_out_by
Person
  P12F.occurred_in_the_presence_of, P14F.carried_out_by
```

Abb. 5: Ausschnitt aus Konfiguration der Facettenqueries (zwecks besserer Lesbarkeit gekürzt und ohne Namespaces)

Die Entkoppelung von Originalmetadaten, CRM Modellierung und Facettenerzeugung erlaubt es, die Facetten äußerst flexibel zu gestalten. So können beispielsweise unterschiedlich konkrete Pfade problemlos unter ein und demselben Namen zusammengefasst werden. Wir haben so beispielsweise die Wahl, ob wir rollenspezifische Personenfacetten (creator, publisher, contributor, ...) ausbilden oder alle involvierten Personen unter einer allgemeinen Personenfacette zusammenfassen wollen. (Wir können auch beides machen). Wenn alle Actors als Personen zusammengefasst werden, sollten wir zusätzlich eine Rollenfacette anbieten. Auf diese Weise kann der Benutzer zunächst bspw. "Brecht" als Person wählen und nachher entscheiden, dass ihn Brecht nur in der Rolle des Regisseurs und nicht des Autors interessiert.

Eine weitere Besonderheit unseres Mappingsansatzes ist, dass wir für jede Entität, die in unserer Nomenklatur als Node bezeichnet wird, weil wir diese Knoten potentiell als Verknüpfungspunkte und Sprungziele ansehen, immer die Existenz der zwei Attribute *label* und *type* sicher stellen, damit ein Client oder ein Widget den Knoten problemlos rendern kann. Zusätzlich zu den beiden genannten Eigenschaften kann ein Node beliebige weitere Eigenschaften speichern. Die Nodes stellen, obwohl sie ohne das CRM nicht gebildet werden könnten, ein eigenständiges Modell dar, das die Grundlage der Vernetzung der Objekte bildet.

## **5 Vernetzung der Objekte, verwendete Smart-Matching Technologien, Identifizierung, Integration bestehender Ontologien bzw. Normdaten**

Die Objekte der DDB erhalten beim Ingest eine technische Kennung, unter der die Objekte auf der Plattform verwaltet werden. Die Kennung wird aus einer Providerkennung und der providerseitigen Kennung des Objekts (z.B. Signatur oder Inventurnummer) gebildet (die ihrerseits beliebig gebildet sein können). Beim Re-Ingest desselben Objekts (gleicher Provider und gleicher providerseitige Kennung) erhält das Objekt dieselbe DDB-seitige Kennung. Eine Versionierung von Objekten seitens der DDB findet nicht statt und ist gegenwärtig auch nicht vorgesehen.

Jedes Objekt des kulturellen Erbes der DDB existiert im Archiv (als AIP). Das AIP ist datentechnisch mehr als vollständig.

Die ID eines Objekts ist in der Suchmaschine der DDB mit sämtlichen Termen der Objektbeschreibung (den Metadaten) assoziiert.

Im sogenannten Nodestore existiert ein gesondertes Modell eines jeden im Archiv gespeicherten Objekts. Dieses wird über den Umweg der CRM Modellierung durch die Reduktion von Transitionen im CRM Graph auf die auch als Facetten verwendeten Attribute erzeugt. Jedes dieser Attribute entspricht also einem Facettennamen und es verweist seinerseits immer auf Nodes. Jeder Node hat eine ID, einen Typ (type) und eine Bezeichnung (label). Diejenigen Nodes, die als Kulturobjekte ingestiert worden sind, enthalten typischerweise eine Reihe von Attributen, die auf solche Nodes verweisen, die typischerweise nicht ingestiert sondern konstruiert wurden. Konstruierte Nodes bezeichnen beispielsweise Orte, Personen, Ereignisse etc. Der Nodestore ist der geeignete Ort, um bestehende Normdaten einzubinden. Gegenwärtig importieren wir initial die Ortsdaten aus einer von Neofonie speziell aufbereiteten Version der Alexandria Ontologie aus dem Theseus Projekt in das System. Ortsbezüge der Kulturobjekte sollen dann automatisch mit den importierten Ortsobjekten verbunden werden (Smartmatchings, siehe unten).

id	i -	P2EMPROSUSAE7NFSW7X5W2SLBWLJF4EP
label_s	i -	Deutsche Nationalbibliothek
ndx.Inhaltstyp_s	i -	7ZAZP7QXR7RCI74DCFLFZTJJBKENFJ42
ndx.crm.P2F.has_type_s	i -	7ZAZP7QXR7RCI74DCFLFZTJJBKENFJ42
ndx.crm.P48F.has_preferred_identifier_s	i -	LCVUH32CB5YRZMAAJ5VCYODBPOUJXGHY
ndx.crm.P76F.has_contact_point_s	i -	UR3WPVPBFBVK5RPOOPAXRXCMMKIWFDKW
node_s	i -	{"id": "P2EMPROSUSAE7NFSW7X5W2SLBWLJF4EP", "label": "Deutsche Nationalbibliothek"}
source_s	i -	construct
type_s	i -	Bibliothek

Abb. 6: Beispiel eines konstruierten Nodes. Das Feld *id* entspricht dem, was in einem relationen Modell ein Primärschlüssel wäre. Die Felder mit dem Präfix *ndx* entsprechen Sekundärschlüsseln. Das Feld *node\_s* enthält eine kondensierte Kopie des Nodes als Json String und wird nur zur Visualisierung benötigt.

id	i -	FCBX7B6IUMGBFWWXJDWU7L3E54K7FO2T
label_s	i -	Kamera
ndx.Datenlieferant_s	i -	TLUSOX63ZMPLG65GW56TMD3Y2C4WDM7K
ndx.Halter_s	i -	ATPJYKHDPTUCIMG3OV2BXPMCUQ6BJ2R
ndx.Inhaltstyp_s	i -	UQFCRTKZ767N6RIXSDW44KDTUFNEPWWI
ndx.Sammlung_s	i -	AF2YS3WFG5WFOLUWGOZMCEZXI6WZRLLOU
ndx.Sparte_s	i -	FCWQKDQRYODCMKTFGXZQ3DJKUMPM2H6U
ndx.Titel_s	i -	EUMMSLVHAXSXYGPBN73HZTXAGFWBSXFF
ndx.crm.P1F.is_identified_by_s	i -	EUMMSLVHAXSXYGPBN73HZTXAGFWBSXFF
ndx.crm.P2F.has_type_s	i -	G2ZUIGJAAQHGGKEC23TKHTDYEBE6Z6JU
ndx.crm.P46B.forms_part_of_s	i -	AF2YS3WFG5WFOLUWGOZMCEZXI6WZRLLOU
ndx.crm.P48F.has_preferred_identifier_s	i -	4KOQPMCIWNOG3ZCHPIAC4BVUTSLTZFMW
ndx.crm.P50F.has_current_keeper_s	i -	TLUSOX63ZMPLG65GW56TMD3Y2C4WDM7K
ndx.crm.P52F.has_current_owner_s	i -	ATPJYKHDPTUCIMG3OV2BXPMCUQ6BJ2R
node_s	i -	{"id":"FCBX7B6IUMGBFWWXJDWU7L3E54K7FO2T","label"
source_s	i -	ingest
type_s	i -	Technisches Gerät

Abb. 7: Beispiel eines ingestierten Nodes.

Das Konzept der Nodes spielt eine Rolle bei der Vernetzung der Objekte. Die Mappings projizieren wie erwähnt die Originaldaten auf das CRM Modell und bilden dabei Strukturen aus, die während des Ingests als Nodes im Nodestore persistiert werden.

Wenn die Originalmetadaten im ASC von den Transformern verarbeitet werden, besteht keine Möglichkeit zu erkennen, ob eine Person oder ein Ort in der Plattform bereits bekannt ist. Diese Objekte werden bei der Transformation mit temporären und nur im Kontext des jeweiligen Objekts eindeutigen Identifier versehen, so dass die CRM Aussagen, die als RDF Triple im SIP gespeichert sind, beim Ingest von einem Service (IdHandler) analysiert und die temporären Identifier der konstruierten Nodes in persistente Identifier übersetzt werden müssen.

Dieser Vorgang soll zumindest in Grundzügen erläutert werden.

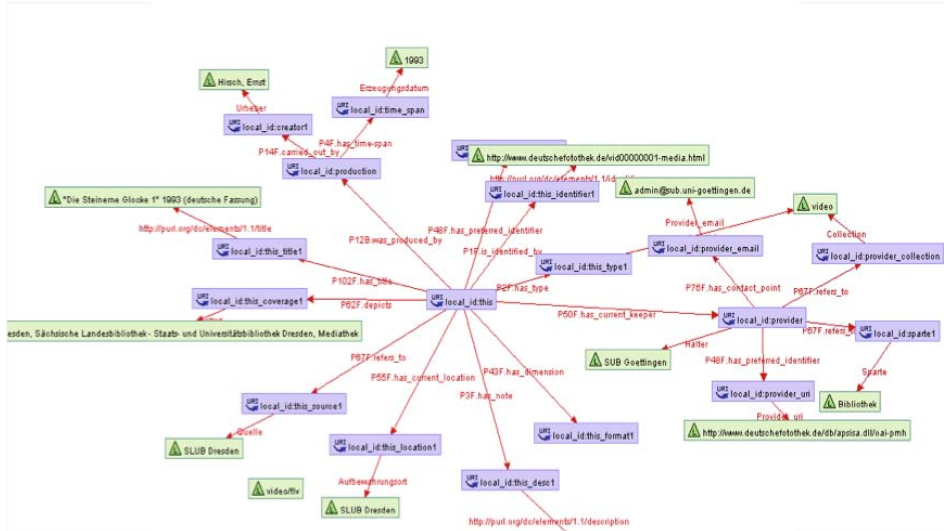


Abb. 8: Beispiel eines Objekts mit den temporären, bei der Transformation zugewiesenen Identifiern, also vor Zuweisung der endgültigen Identifier.

Bei der Vernetzung eines neuen Objekts mit seinen (konstruierten) Subnodes wird zunächst ein Dependency Graph des Modells gebildet, so dass der zu ingestierende Graph von den Rändern her nach innen durchlaufen werden kann.

Für jeden Subknoten werden dann alle Properties typisiert ausgewertet. Ziel der nachfolgenden Operationen ist festzustellen, ob ein entsprechender Subknoten im Nodestore bereits vorhanden ist. Die Typinformation wird auf Grundlage der CRM Domain und Range Informationen gewonnen und dient dazu, typspezifische Distanzmaße für die Ähnlichkeitsbestimmung von Nodes zu selektieren. Dies bedeutet, dass Informationen wie Normdatenbezüge, die z.B. als URNs vorliegen können, anders verglichen werden (Binärvergleich) als Literale, die (Personen-)Namen enthalten, und damit wiederum anders als numerische oder Datumswerte. Die Analyse der Properties eines Sub-Knotens hat neben der typspezifischen Auswahl des Vergleichsverfahrens den Zweck, zuverlässigere Attribute höher zu wichten als weniger zuverlässige. Die Gesamtheit der gewichteten lokalen Ähnlichkeitsmaße der einzelnen Attribute eines Subknotens wird dann zu einem Ähnlichkeitsmaß des gesamten Subknotens zusammengefasst. Ein Schwellwert entscheidet, ob zwei Subknoten als ähnlich gewertet werden. Wird ein ähnlicher Subknoten erkannt, dann wird der Identifier des gefundenen Subknotens zugeordnet, andernfalls wird ein neuer Subknoten erzeugt.

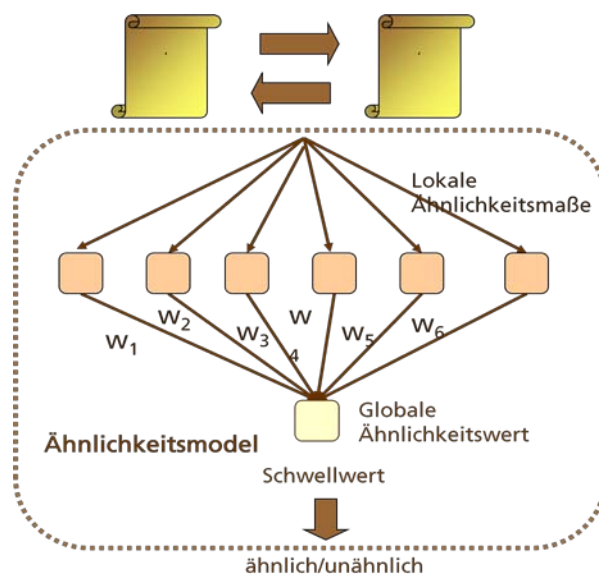
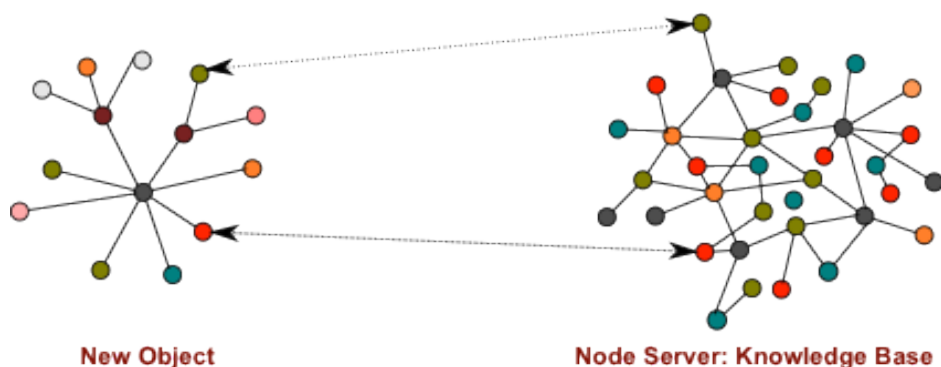


Abb. 9: Ähnlichkeitsmodell [FRIE]

Es versteht sich, dass die Schwellwerte trainiert werden müssen, und wir gehen davon aus, dass man den Bestand der DDB regelmäßig reingestiert, um Performanz- und Verhaltensoptimierung wirksam werden zu lassen.

Das aufwändig erscheinende Verfahren läuft vertretbar schnell, da native approximative Suchmechanismen der Suchmaschine (Solr) verwendet werden, um eine begrenzte Menge von Kandidaten für den paarweisen Vergleich zu identifizieren.

Bei den zur Zeit relativ populären semantischen Lösungsansätzen, die zur Queryzeit einen Triplestore per Sparql-Queries befragen, um Beziehungen zwischen Objekten zu analysieren, erzielt man eine sehr hohe Flexibilität der Anfragen um den Preis der Nutzbarkeit des Systems. Zum einen, weil Sparql-Queries nur von hoch spezialisierten Nutzern ausgeführt werden können, zum anderen, weil die Engines schon bei wenigen Millionen Objekten spürbar langsam werden, sowohl was Inserts als auch was die Query-Seite angeht. Komplexere Queries auf großen Datenbeständen können heute nicht sinnvoll synchron durchgeführt werden. Dennoch können solche Abfragen für Forschungszwecke sinnvoll sein. Die IAIS CORTEX Plattform implementiert für diesen Zweck einen weiteren Store (den Graphstore), der ebenfalls als Index realisiert ist, gegenwärtig noch nicht befüllt wird, und im Rahmen einer späteren Ausbaustufe entwickelt werden soll.



Bei den der DDB heute zugrunde liegenden Verfahren werden die rechenaufwändigen Prozesse, bei denen Objekte (teils mittels Smart-Matching Verfahren) in ein sich ausbildendes Wissensnetz eingebettet werden, für jedes Objekt nur einmal und zwar während des Ingests vorberechnet. Diese teils 'geratenen' Relationen werden dem Nutzer angeboten, wenn er sich navigierend im Objektnetz bewegen will. Für spätere Ausbaustufen könnte *die Crowd* zu Bewertung der geratenen Relationen herangezogen werden.

Die semantisch präzisen Relationen, die als Facetten oder Filter zur Selektion verwendet werden, sind ebenfalls vorberechnet und können in nutzerfreundlicher Weise und performant angefragt werden. Für die Berechnung der Filter haben die Smartmatchings jedoch keine Relevanz, da hierfür nicht die Nodes, sondern die literalen Werte benötigt werden.

## Verweise

- [IESE] "Architektur-Bewertung: Deutsche Digitale Bibliothek" von Balthasar Weitzel und Glib Kutepov, Fraunhofer IESE, November 2011.
- [EURO] Screenshot nach Query auf "[http://europeana.eu/portal/search.html?query=\\*](http://europeana.eu/portal/search.html?query=*)" vom 4. Dezember 2011.
- [FORTH] A New Framework for Querying Semantic Networks, von Katerina Tzompanaki und Martin Doerr, [www.ics.forth.gr/tech-reports/2011/2011.TR419\\_Querying\\_Semantic\\_Networks.pdf](http://www.ics.forth.gr/tech-reports/2011/2011.TR419_Querying_Semantic_Networks.pdf)
- [BMU] British Museum Semantic Web Collection Online, <http://collection.britishmuseum.org/>
- [FRIE] Erkennung von ähnlichen Objekten beim Ingest, Natalja Friesen, IAIS