Pascal Christoph

Datenanreicherung auf LOD-Basis

1 Einleitung

> „[A]ccess to information is not so much of an issue anymore, but rather aggregation and contextualisation of data and information and thus knowledge enabling... And with knowledge on their agenda libraries are back in the role they once had, before the advent of printing."[1]

Immer weniger Metadaten werden in Bibliotheken erstellt: Metadaten kommen zunehmend von Verlagen, aus großen Systemen wie z. B. Ex Libris' Alma oder OCLCs Worldshare, und potentiell aus Crowdsourcing-Projekten wie Open Library[2], oder zukünftig Wikidata[3]. Bibliothekare werden daher in Zukunft eher „Information Broker" denn „Information Provider" sein, d. h. die Aufgabe wird eher darin bestehen, *Kontexte* zu (Titel- und Norm-)Daten herzustellen als diese gänzlich neu zu erfassen. Die Herstellung von Kontext kann als Kataloganreicherung einen Mehrwert bei der Recherche mit sich bringen: Ein Buch, das häufig in den 1990er Jahren von Linguisten zitiert wurde, hat eben eine wichtige Bedeutung für die Linguistik der 90er Jahre und ist eben deshalb interessant für jemanden, der sich einen Einblick in die Linguistik des ausgehenden 20. Jahrhunderts verschaffen möchte.

Doch wozu ist Kataloganreicherung überhaupt wichtig?

Die originäre Aufgabe bibliographischer Daten besteht in der Unterstützung des Menschen, relevante bibliographische Ressourcen aufzufinden. Wenn bibliographische Daten mit zusätzlichen Daten angereichert werden, dann kann diese Aufgabe unter Umständen besser erfüllt werden. Zusätzliche Schlagwörter oder Tags machen ein Dokument besser recherchierbar – sei es durch Erweiterung des Suchindexes oder der Möglichkeit mittels Facetten durch die Ressourcen zu „browsen". Deshalb ist ein Desiderat der Bibliothekswelt die Kataloganreicherung. Die Anreicherung von Katalogdaten findet statt durch Daten*verknüpfungen*, z. B. zu Klassifikationen, Inhaltsverzeichnissen, Cover-Bildern und zu anderen

1 Stefan Gradmann in „From Containers to Content to Context: the Changing Role of Libraries in eScience and eResearch", siehe http://conference.ub.uni-bielefeld.de/programme/abstracts/gradmann.htm

2 http://openlibrary.org/

3 http://meta.wikimedia.org/wiki/Wikidata/de

Kontexten (z.B. nach Domänen geordnete Referenzierungen[4], Ausleihhäufigkeit usw.). Das schon im Begriff von Linked Open Data (LOD) steckende Wort „linked" (also: „*verknüpft*") legt nahe, dass die Vorhaltung bibliographischer Metadaten in Form von Linked Data prädestiniert ist, um Kataloganreicherung durchzuführen.

Durch Linked Open Data ist es aber nicht nur möglich, Daten zu *konsumieren* (und damit den eigenen Katalog anzureichern), vielmehr ist die Basis von LOD das *Publizieren* der Daten. Ein Seiteneffekt dieser Art der Arbeit an den eigenen Daten ist die Steigerung ihres Wertes, da sie dadurch in größeren und in ganz anderen Zusammenhängen als der Bibliothekswelt gebraucht werden können. Als LOD exponierte Daten können wiederum von anderen konsumiert werden und sich mit deren Datentöpfen verbinden.[5] Geschieht dies wiederum auf Basis von Linked (Open) Data, dann lassen sich diese *von anderen erzeugten* Verknüpfungen prinzipiell wieder *re*konsumieren: Es kann sich also (bis zu einem gewissen Sättigungsgrad) ein sich selbst verstärkender Kreislauf der wechselseitigen Datenanreicherung entwickeln.

Dieser Beitrag beleuchtet einführend theoretische Aspekte, die im Zusammenhang mit Kataloganreicherung auf Basis von Linked (Open) Data wichtig sind. Danach wird anhand des vom Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (hbz) betriebenen LOD-Services lobid.org beispielhaft gezeigt, wie bibliographische Daten auf Basis anderer LOD-Quellen angereichert werden können. Abschließend werden zukünftige Chancen und Möglichkeiten der Kataloganreicherung betrachtet.

2 Definition von Kataloganreicherung

Die Wikipedia definiert: "Mit **Kataloganreicherung** (englisch *catalog enrichment*) werden Einträge eines Bibliothekskatalogs um weiterführende Informationen ergänzt, die über die reguläre Formal- und Sacherschließung hinausgehen."[6]

---

4 Im Linked-Data-Netz kann gezählt werden, wie viele Ressourcen die eigene Ressource referenzieren. Die Anzahl der Referenzierungen ist alleine schon eine neue Information. Werden jetzt diese Referenzierungen auch noch nach ihrer Ursprungsdomäne aufgeschlüsselt – handelt es sich z. B. um Universitätswebseiten oder Fernsehwebseiten - dann können weitere erkenntnisfördernde Aussagen hinzugefügt werden.

5 Siehe auch den Beitrag in diesem Sammelband „Open Data und Linked Data in einem Informationssystem für die Archäologie" von Maike Lins und Hans-Georg Becker.

6 Seite „Kataloganreicherung". In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 5. Juli 2012, 12:41 UTC. URL: http://de.wikipedia.org/w/index.php?title=Kataloganreicherung&oldid=105215379 (Abgerufen: 5. Oktober 2012, 13:31 UTC)

Dazu gehören also beispielsweise Inhaltsverzeichnisse und -angaben, Rezensionen, Volltexte, Coverabbildungen und zusätzliche Schlagwörter. Dabei spielt es im Prinzip keine Rolle, ob die Daten maschinell oder intellektuell, von einer Bibliothekarin oder durch Crowdsourcing von einem Studenten angefertigt wurden.[7]

Die sogenannte „Query Expansion", bei der die Sucheingaben des Benutzers erweitert werden durch Hinzuziehen etwa eines Thesaurus oder Synonymwörterbuchs, zählt an dieser Stelle nicht als ein Mittel zur dynamischen Kataloganreicherung[8]: „Katalog" wird hier Katalogdatenzentriert verstanden, d. h. ein Kopieren der *Katalogdaten* (also Texte oder Links) muss genügen, um die Daten präsentieren zu können. Da die vom Benutzer eingegebenen Suchbegriffe nicht Teil dieser Katalogdaten sind, reichern sie auch nicht den Katalog an – sie reichern lediglich die Suchanfrage an.

3 Berücksichtigung von Lizenzen

Nicht immer ist die Entscheidung für oder wider eine der drei im folgenden Kapitel beschriebenen Anreicherungsarten freiwillig – die Auswahl einer Nachnutzungsform kann neben technischen insbesondere rechtlichen Beschränkungen unterliegen. Ist – wie z. B. in LibraryThing – das Verwenden von Teilen der Daten nur in einem nicht-kommerziellen Rahmen erlaubt[9], so ist die Datenübernahme in einen LOD-Dienst nicht möglich[10], lediglich der Link zu einer Ressource darf gespeichert werden. Sind die Daten hingegen frei verwendbar, wie z. B. die Public-Domain-Daten der Open Library[11], dann dürfen die Daten in jedes Anwendungsszenario integriert werden. Alleine aus lizenzrechtlichen Gründen wird es wahrscheinlich in einem komplexeren LOD-Portal eine Mischung aller drei im folgenden Kapitel beschriebenen Szenarien geben.

---

7 Das soll nicht heißen, dass es nicht wichtig wäre, die Information über die Herkunft der Daten oder die Benennung des Algorithmus vorzuhalten – siehe dazu Abschnitt „Provenienz".

8 Im Gegensatz zu der genannten Wikipediadefinition (ebd.), wo es heißt: „Die so genannte Query Expansion, [...] die Erweiterung der Benutzeranfragen durch zusätzliche semantische Ressourcen (Thesaurus, Ontologie, [...]) ist eine weitere Option."

9 Siehe https://www.librarything.com/wiki/index.php/LibraryThing_APIs.

10 Schließlich bedeutet das „Open" in „Linked Open Data", dass die Daten auch einer kommerziellen Nachnutzung offenstehen. Vgl. Pohl/Danowski in diesem Sammelband, Abschnitt 4.

11 http://Open Library.org/about

4 Drei Formen der Kataloganreicherung

Werden Daten angereichert, kann dies entweder lediglich durch Speichern von Referenzen geschehen (z. B. durch einen Link zu einem Inhaltsverzeichnis) oder/und durch direkte Verspeicherung der durch diese Referenzen erreichbaren Neudaten (z. B. Inhaltsverzeichnisse in maschinenlesbarer Textform). Im Zusammenhang mit der Lizenzbestimmung folgen drei technisch grundlegend verschiedene Ansätze der Anreicherung, mit jeweils spezifischen Vor- und Nachteilen.

4.1 Bloße Verlinkung

Die primitivste Form der Kataloganreicherung funktioniert auch mit restriktiv lizenzierten Daten. In den bibliographischen Daten werden dabei lediglich URLs zu den Anreicherungsdaten hinterlegt. Diese Links lassen sich im Portal darstellen, die Benutzerin kann diese Links anklicken und somit durch die Daten browsen. Dadurch verlässt die Benutzerin das eigene Portal, wobei sie in eine andere Anwendung gelangt.

4.1.1 Nachteil

Der Bruch der Anwendungsoberfläche beim Browsen wird oft als verwirrend empfunden, u. a. weil die neue Oberfläche meist ein ganz anderes (Navigations-)Design haben dürfte. Es bedeutet einen Mehraufwand für die Benutzerin, um in der neuen Umgebung erfolgreich recherchieren zu können. Wenn zudem in diesem neuen Portal ebenfalls Links zu anderen Portalen führen, steigt die Gefahr eines „Verirrens“ stetig, und es ist nicht immer leicht, zum eigentlichen Ausgangspunkt zurückzukehren.

Dazu ein Beispiel: folgt die Benutzerin den Links in das fremde Portal und kann sie dort vielleicht tatsächlich eine passende Ressource finden, so wird wahrscheinlich dort eine Verfügbarkeitsrecherche nicht für die Bibliotheken ihrer Wahl durchführbar sein. In diesem Fall muss durch das händische Kopieren von Identifizierungsmerkmalen in das Ausgangsportal eine Verfügbarkeitsprüfung angestoßen werden.

Der größte Nachteil besteht sicherlich darin, dass über Daten, die lediglich verlinkt sind, keine integrierte Suche stattfinden kann.[12]

12 Handelt es sich bei den Links um Daten, die über einen SPARQL-Endpoint angeschlossen sind, so lassen sich zwar sog. „federated searches“ konstruieren (siehe hierzu auch den Ab-

Die genannten Nachteile zeigen, dass diese Form der Anreicherung eine umständliche Nutzererfahrung bewirkt, weshalb bloße Verlinkung oft suboptimal ist.

4.1.2 Vorteil

Die Einfachheit dieser Anreicherungsart ist ihr Vorteil. Handelt es sich bei den Anreicherungsdaten z.B. um Volltexte oder Inhaltsverzeichnisse oder Ähnliches, so kann diese Art der Anreicherung durchaus sinnvoll sein. Die Verlinkung ist schnell und ohne großen technischen Aufwand durchgeführt, wobei es nicht notwendig ist, Daten lokal zu speichern, für ihre Synchronisierung zu sorgen oder ähnliches.

Ein weiterer Vorteil ist, dass Verlinkungen auch zu restriktiv lizensierten Datenquellen erfolgen können.

Der Link auf die Fremdressource sollte zudem auf jeden Fall - auch bei Umsetzung der anderen Anreicherungsformen - immer auch mit verspeichert werden, da er grundlegend für die Rekonstruktion der Provenienz (also der Herkunft) der Daten ist.[13]

4.2 Dynamische Kataloganreicherung

Bei der dynamischen Kataloganreicherung werden, wie im Szenario „Verlinkung", in den bibliographischen Daten lediglich HTTP-URIs zu den neuen Daten hinterlegt. Liegt hinter diesen URIs *direkt* ein konsumierbares Objekt (z.B. ein Textdokument, ein PDF oder ein Bild) oder eine strukturierte Quelle (wie etwa RDF)[14], dann lassen sich diese externen Datenquellen dynamisch, also zur Laufzeit generiert, konsumieren, indem sie etwa durch ein Javascript Mashup in das eigene Portal eingeblendet werden.[15]

---

schnitt „API"), aber aus Performanzgründen sind diese Abfragen eher nicht zu empfehlen, weil nur eingeschränkt möglich (z.B. ist eine Suche über kontrolliertes Vokabular vielleicht möglich; eine Freitextsuche mit Wildcards auf ein Datenset mit 100 Millionen Tripeln ist aber nicht in annehmbarer Zeit durchführbar).

13 Siehe auch den Abschnitt „Provenienz" und den Beitrag von Kai Eckert in diesem Band.

14 Im Gegensatz zu einer HTML-„Landing-Page", die lediglich - und dies gilt leider für die meisten Dokumentenserver – wiederum nur das gewünschte Objekt maschinenunlesbar verlinkt.

15 Ein Beispiel für die Javascript-basierte Integration von DBpedia in Primo bietet die Suchmaschine des Österreichischen Bibliotheksverbunds (OBVSG): http://search.obvsg.at/primo_library/libweb/action/search.do (dort z.B. nach „Harry Potter" suchen). Ein anderes Beispiel ist

4.2.1 Nachteil

Da die Anreicherungsdaten bei der dynamischen Kataloganreicherung extern gespeichert sind, können sie nicht in eigene Suchindexe eingespielt werden. Viele Dienste, die ein Kopieren und Verspeichern ihrer Daten verbieten, erlauben aber dynamische (Such-)Abfragen durch Bereitstellung einer API. Ist z.B. ein SPARQL[16] Endpoint[17] verfügbar, so lassen sich sog. Federated Queries über diese HTTP-URIs absetzen – dabei werden mehrere Endpoints in einer einzigen Suchanfrage gleichzeitig abgefragt. Die Ergebnisse lassen sich dann ebenfalls dynamisch in die Benutzersicht einblenden. Auf diese Weise kann über Daten gesucht werden, obwohl sie nicht in einen lokalen Index eingespielt werden dürfen.[18] Bei Anwendung dieser Technik sollte man sich allerdings auf das Suchen über URIs beschränken[19], weil bei anderen Anfragen die Performanz leidet.[20]

Aus dem gleichen Grund, nämlich dem Fehlen der Daten in einem lokalen Suchindex, gibt es normalerweise keine Facettierung ohne Portalgrenzenbruch. Diesem Umstand kann wiederum mit einer API begegnet werden.[21]

Da die Daten immer aktuell bei Anfrage abgeholt werden, entfällt der Aufwand von Updates. Es besteht jedoch die Gefahr bei einer eventuellen Umstel-

---

lobid.org, wo die Schlagworte aus den GND-Links mit einer SPARQL-Abfrage dynamisch aufgelöst werden, um dem Benutzer auf der Webseite einen menschenlesbaren Namen der URI zu präsentieren, siehe z.B. http://lobid.org/resource/HT002948556 .

16 SPARQL ist eine graph-basierte Abfragesprache für RDF. Der Name ist ein rekursives Akronym für **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage, siehe : https://en.wikipedia.org/wiki/SPARQL

17 Endpoints sind die Schnittstellen zu den Tripel Stores, in denen die RDF Daten liegen.

18 Z.B LibraryThings JSON-APIs: „License: Must be run as Javascript on user's browser, not fetched by a server; cannot be stored, except for browser caching.", siehe: https://www.librarything.com/wiki/index.php/LibraryThing_APIs

19 Z.B. „Suche Ressourcen, deren Autoren unter „Personen zu Mathematik" (http://d-nb.info/standards/vocab/gnd/gnd-sc#28p) subsumiert sind".

20 So sind Suchen mit Wildcards über 100 Millionen Tripel in einem Tripel Store nicht performant zu haben. Für Suchen wird deshalb oft auf Suchmaschinen oder suchmaschinenähnliche Indexe zurückgegriffen, siehe z.B. „Virtuoso": „Virtuoso has an optional full-text index on RDF literals. Searching for text matches using the SPARQL regex feature is very inefficient in the best of cases." (http://virtuoso.openlinksw.com/virt_faq/"Do you support full-text search?")

21 Siehe z. B. die Webservices von LibraryThing: ein Browsen durch Tags ist möglich, *ohne* die eigenen Portalgrenzen zu verlassen. Doch dies bedingt, dass Bestandsangaben der eigenen Bibliothek an LibraryThing weitergegeben werden, damit eine API auf Seiten von LibraryThing die Auflösung von Tag zu Ressource ermöglichen kann. Damit ist es notwendig, anders als in den anderen Anreicherungsformen, dass auch auf Seiten des verlinkten Fremdanbieters Datenintegrationsaufwand betrieben wird. Die Datenhoheit liegt bei LibraryThing. Für ein Beispiel siehe den Katalog der Dortmunder Stadt- und Landesbibliothek: http://katalog.dortmund.de:8080/webpac-bin/wgbroker.exe?+new+-access+top.do_intern_ger+search+open+ISBN+3423071516

lung der Datenstruktur oder gar bei einem Ausfall des Services des Fremdanbieters, dass zeitgleich die Daten im eigenen Portal nicht mehr ordentlich dargestellt werden. Zudem kann die dynamische Integration zu spürbaren Performanzeinbußen führen.

4.2.2 Vorteil

Da die Daten immer aktuell bei Anfrage abgeholt werden, entfällt der Aufwand von Updates.

Die dynamische Kataloganreicherung funktioniert auch mit restriktiver lizenzierten Daten, die z.B. für die im nächsten Abschnitt beschriebene Kataloganreicherungsform „Datenübernahme" nicht verwendet werden dürfen.

Der Datenintegrationsaufwand beschränkt sich auf das Portal – in den eigenen Datenspeichern muss nichts angepasst werden (siehe dazu das Kapitel „Datenintegration").

4.3 Datenübernahme

Der durch die Verknüpfung erreichte Anschluss der Fremddaten an die eigenen Daten wird durch eine teilweise oder vollständige Integration der Fremddaten in den eigenen lokalen Index vollendet.

4.3.1 Nachteil

Die zur Datenintegration notwendigen Arbeiten sind mindestens ebenso hoch wie bei der dynamischen Anreicherung. Höher wird der Aufwand, wenn eine einheitliche Feldersuche oder Facettierung über alle Daten aus diesen verschiedenen Quellen möglich sein soll: Hierzu bedarf es, je nach Art der Vokabulare, verschiedener Mappings, um zu einer für Facettierung oder Feldersuche notwendigen Vereinheitlichung zu gelangen.[22]

Da die Daten vom Datenanbieter abgeholt werden müssen, fällt die Aktualität der Daten mit der Periodizität der Datenabholung zusammen. Wenn der Datenanbieter keine inkrementellen Updates bietet, sondern lediglich Vollab-

22 Z.B. kann der Zeitpunkt der Erscheinung einer Ressource mit dct:issued oder isbd:P1021 angegeben sein.

züge, dann ist bei größeren Datenmengen eine zeitnahe Aktualisierung unmöglich oder zumindest erschwert.

4.3.2 Vorteil

Im Gegensatz zur dynamischen Anreicherung sind mit einer Datenübernahme auch Freifeldsuchen, Einzelfeldsuchen und Facetten zum explorativen Suchen über die Fremddaten möglich, je nach dem Grad des betriebenen Datenintegrationsaufwandes.

Ein weiterer Vorteil liegt in der Unabhängigkeit von der technischen Infrastruktur des Fremdanbieters: Die Daten können performant aus dem eigenen Backend geholt werden und einer eventuellen Datenstrukturumstellung des Datenanbieters kann kontrolliert begegnet werden, da die neuen Daten aktiv abgeholt werden und somit vor einer Übernahme validiert werden können. Sollte sich dabei herausstellen, dass für die neuen Daten z.B. ein neues Mapping benötigt wird, kann die automatische Integration der neuen Daten zurückgestellt werden, um vorerst mit den alten, aber in die Portallogik integrierten Daten weiterzuarbeiten.

4.4 Resümee

Vor einer Daten*integration*, egal ob dynamisch oder per Übernahme, muss zuerst immer ein Blick auf die Lizenzen geworfen werden.[23] Dies ist bei einer **reinen Verlinkung** nicht notwendig. Auch technisch ist die reine Verlinkung am einfachsten umzusetzen. Allerdings bringt sie auch den geringsten Mehrwert für den Benutzer mit sich. Aus lizenzrechtlichen Gründen führt manchmal kein Weg am bloßen Verknüpfen vorbei. Aus der Erfahrung lässt sich aber leider berichten, dass viele Volltextlinks, Links zu Abstracts oder zu Inhaltsverzeichnissen keine *direkten* Links zur Ressource darstellen. Sie führen oft lediglich zu einer Landing Page (oder „Splash"-Seite, also eine Webseite), auf der wiederum der Link zur tatsächlichen Ressource hinterlegt ist.[24] Ein Inhaltsverzeichnis lässt sich dann nicht als Mashup in das Portal einblenden. Daraus folgt, dass auch bei den beiden anderen Anreicherungsformen immer ein Anteil an einfacher Verlinkung

23 Siehe hierzu das Kapitel „Lizenzen"

24 Beispiel für Landing Pages sind: http://dx.doi.org/10.1007/978-1-4419-9443-1 und http://edoc.vifapol.de/opus/volltexte/2012/3730. Für die damit verbundenen Probleme siehe meinen Blog-Post http://www.dr0i.de/lib/2011/03/23/publisher_make_urls_useless.html.

mitsamt Portalanwendungsbruch vorhanden sein wird. Außerdem ist der Link zur Fremddatenressource obligatorisch für die Nachhaltung von Provenienzangaben und sollte deshalb auch bei den beiden anderen Formen nachgehalten werden.

Aus Anwendersicht bietet die **Datenübernahme** am meisten: Selbst wenn die Daten nicht zu 100% zu den eigenen Daten passen, um z. B. nahtlose Facettenintegration zu ermöglichen, so ist doch allein die Möglichkeit, über alle Daten suchen zu können, ein Gewinn. Die komplette Datenübernahme ist außerdem die nachhaltigste Form: Werden die Daten, die durch die Anreicherungen gewonnen wurden (z. B. Social Tags und Rezensionen usw.) als Open Data wieder zurückgegeben, dann sind sie direkt auch außerhalb der eigenen Anwendung wiederverwendbar und tragen mit dazu bei, dass der Pool an freien Daten mit dem ihm inhärenten Potential weiter wächst.[25] Diese Form ist hingegen mit dem größten Entwicklungsaufwand verbunden.

Sowohl was die Lizenzen als auch was die Technik angeht, ist die **dynamische Anreicherung** ein Kompromiss aus einfacher Verlinkung und kompletter Datenübernahme: Das dynamische Einbinden der Daten ist häufiger erlaubt als die Möglichkeit alle Daten zu laden und zu verändern; und wenn es schon keine Suchmaschinenintegration gibt, so ist doch eine Integration der Daten in ein Portal möglich und der Portalanwendungsbruch, wie bei der reinen Verlinkung, entfällt.

Aus den oben detaillierter beschriebenen Vor- und Nachteilen ergibt sich folgende, grob verallgemeinerte Kurzübersicht:[26]

25 Ein willkommener Seiteneffekt der Nachnutzung und Verbreitung von Open Data ist – im Hinblick auf Langzeitverfügbarkeit – die redundante Datenhaltung nach dem LOCKSS-Prinzip („Lots Of Copies Keep Stuff Safe").

26 Die Smilies sind der „Open Icon Library" entnommen, siehe: http://sourceforge.net/projects/openiconlibrary/

Tabelle 1: Kurzübersicht Vor- und Nachteile der Anreicherungsarten.

	nur Verlinkung	dynamisch	Übernahme
Lizenzbeschränkung			
Aktualisierungsaufwand			
Datenaufbereitungsaufwand			
Präsentationsperformanz			
Fremdanbieterunabhängigkeit			
Portalintegration			
Suchmaschinenintegration			

5 Provenienz[27]

Das Nachhalten von Informationen zur Provenienz (also: Herkunft) der Daten ist unter verschiedenen Aspekten wichtig. Sowohl bei der Datenhaltung in den Backends als auch bei der Datenpräsentation im Frontend sollten nachgenutzte Daten aus anderen Quellen von selbst erstellten, „eigenen“[28] Daten unterscheidbar sein: Im Backend lassen sich z. B. besondere Suchfelder verwenden, die eine andere Rankinggewichtung haben als die Originärdaten[29]; im Frontend ließe

27 Siehe auch Kai Eckerts Beitrag zum Thema Provenienz in diesem Band.

28 Was sind eigentlich „eigene“ Daten? Die „Originärdaten“ des Katalogs bestehen ja oft schon aus Übernahmen sog. Fremddaten. Da die Herkunft der Quellen der Daten in den Katalogen bisher oft nicht oder nur unzureichend festgehalten wurde, kann für einen solchen Katalogdatensatz nur gelten, dass er der „Originärdatensatz“ ist.

29 Z.B. können bei der Open Library Tags frei vergeben werden, und dies zudem von allen Benutzern und nicht nur von Bibliothekaren. Es ist also nicht unwahrscheinlich, dass auch Ressourcen, die sich nur am Rande z.B. mit dem Thema „Semiotik“ beschäftigen, trotzdem dieses Tag bekommen, wenn eine Benutzerin einen interessanten, aber kleinen Abschnitt dazu in der Ressource gelesen hat. Im Suchindex könnten z.B. die Tags mit einem Faktor von 0.3 gewichtet sein, sodass dieses bibliographische Objekt durchaus gefunden würde bei einer Suche nach „Semiotik“, es würde aber nicht so hoch gerankt wie eine Ressource mit dem vom Bibliothekar vergebenen Schlagwort „Semiotik“.

sich z. B. selektieren, ob (und welche) Fremddatenquellen zur Anzeige gebracht werden sollen. Über An- und Abschalten von Datenquellen können z. B. verschiedene Ebenen von Verlässlichkeit oder auch von Details, oder domänenspezifische Ausschnitte erzeugt werden. Dies kann wichtig sein, da die Daten aus sehr verschiedenen Quellen stammen können, angefangen von anderen Bibliotheken mit einer dementsprechend hohen Authentizität, über teilweise supervisierte Crowdsourcing-Projekte wie Wikipedia und Open Library, bis hin zu privaten Webseiten auf denen Aussagen über Katalogressourcen vorliegen.[30]

6 Vokabular

Sollen Datenquellen angebunden werden, so muss man sich mit dem verwendeten Vokabular auseinandersetzen, um die Daten richtig miteinander in Beziehung bringen zu können. Das Vokabular ist für zweierlei Dinge grundlegend: Zum einen gilt es – um überhaupt Daten miteinander verknüpfen zu können - Anknüpfungspunkte zu finden, im Idealfall sind dies Identifier wie z. B. die ISBN.[31] Zum anderen müssen auch andere Daten bei der Zusammenführung, ob im Backend oder im Frontend, in Beziehung gebracht werden, z. B. für Facettierung. Es ist also ein Mapping von Fremdvokabular zum eigenen Vokabular notwendig.

Die Prädikate definieren die Beziehungen von Subjekt und Objekt. Sie sind also mindestens das, was in MARC/MAB/PICA die Felder und Unterfelder sind. Verwenden beide Datensets die gleichen für die Verlinkung notwendigen Prädikate, z. B. bibo:isbn[32] und dct:issued, dann ist das Mapping eine einfache 1 zu 1 Abbildung. Werden verschiedene Prädikate verwendet, so müssen diese Prädikate erst in Verbindung gebracht werden. Fehlt ein sog. Vocabulary Alignment (also die Beschreibung der Beziehung unterschiedlicher Vokabulare zueinander), so muss die Beziehung durch eigene Arbeit hergestellt werden. Ist aber schon der Gebrauch von z. B. dct:title teilweise unterschiedlich (z. B. mal mit, mal ohne Zusätze zum Hauptsachtitel), so verschärft sich potentiell die Differenz bei Verwendung unterschiedlicher Vokabulare – zumindest wird der zu betreibende Aufwand des Vokabularmappings größer sein. Deshalb ist es vorteilhaft, für das

30 Z. B. eine Kurzrezension über Thomas Bernhards „Auslöschung“ in RDFa: http://www.dr0i.de/lib/2012/06/03/thomas_bernhard_auslschung_ein_zerfall.html

31 Es kommt natürlich darauf an, welche Daten verknüpft werden sollen: für eine Verknüpfung von FRBR-Manifestationen ist eine ISBN alleine nicht ausreichend, hier muss z. B. noch die Auflage berücksichtigt werden.

32 zur Auflösung aller in diesem Beitrag genannten Präfixe, aka Namespaces, siehe http://prefix.cc/.

Datenmapping die zu integrierenden Daten mit einem geläufigen Vokabular zu beschreiben. So ist die Wahrscheinlichkeit größer, dass die beiden Datensets die *gleichen* Vokabulare benutzen und damit überhaupt die Notwendigkeit eines Datenmappings entfällt.[33] Zudem ist der Gebrauch von häufig vorkommenden Vokabularen wahrscheinlich besser dokumentiert.[34] Für die Datenmodellierung sind spezielle, genauere Vokabulare natürlich besser geeignet, um die Informationen zu beschreiben: z. B. ist isbd:P1004 und isbd:P1006 für die Beschreibung des Sachtitels genauer als ein sehr allgemein definiertes Prädikat wie dct:title. Es spricht aber nichts gegen Redundanz: Beispielsweise wird in lobid.org sowohl das Prädikat dct:title verwendet, da es auch in nicht-bibliothekarischen Kreisen bekannt ist, als auch die genaueren isbd-Prädikate.

7 API

Eine API (englisch: *application programming interface*, deutsch: *„Schnittstelle zur Anwendungsprogrammierung"*) ist eine Programmierschnittstelle. APIs können abgefragt werden und liefern eine Antwort. Somit können sie von anderen Programmen benutzt werden. Mit Aufkommen des WWW enstand die Möglichkeit, APIs über das Internet anzubieten, und somit jedem zur Verfügung zu stellen, der an das Internet angebunden ist. Solche APIs werden auch Web-Services genannt. Web-Services können z. B. programmiersprachenspezifisch sein, d. h., dass für ihre Nutzung eine bestimmte Programmiersprache benutzt werden muss.[35] Web-Services können oft einfach über HTTP angesprochen werden. Genügen diese APIs dann vor allem auch noch der Anforderung „[...] dass eine URL genau einen Seiteninhalt als Ergebnis einer serverseitigen Aktion (etwa das Anzeigen einer Trefferliste nach einer Suche) darstellt, wie es der Internetstandard HTTP für statische Inhalte (Permalink) bereits vorsieht [...]",[36] dann heißen sie *RESTful* Web-Services oder auch Web-API. Die URLs lassen sich z.B. einfach im Browser

33 Um eine möglichst einfache Integration der eigenen Daten in fremde Kontexte zu ermöglichen, setzt OCLC deshalb vor allem auf schema.org, siehe dazu auch Ed Summer unter http://inkdroid.org/journal/2012/07/06/straw/

34 Um zu entscheiden, welche Vokabularien geeignet sind, siehe auch den Beitrag in diesem Sammelband „Vokabulare für bibliographische Daten - Zwischen Dublin Core und bibliothekarischem Anspruch" von Carsten Klee.

35 Siehe z. B. http://wiki.ckan.org/API

36 Seite „Representational State Transfer". In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 2. Oktober 2012, 05:36 UTC. URL: http://de.wikipedia.org/w/index.php?title=Representational_State_Transfer&oldid=108774737 (Abgerufen: 9. Oktober 2012, 15:32 UTC).

aufrufen.[37] Das Ergebnis des URL-Aufrufs ist unabhängig vom Zustand des anfragenden Programms (etwa eines Browsers). Der Aufruf ist also „Session-unabhängig". Solche URLs lassen sich z. B. „bookmarken" oder jemand anderem per Mail senden.

LOD *ist* bereits auch eine API, nämlich eine Web-API[38] (was ein Synonym für RESTful Web-Service ist). Kommen neben den basalen Prinzipien von LOD, nämlich Dereferenzierung von HTTP-URIs und Bereitstellung von Daten in RDF, noch die SPARQL-Technik hinzu (heutzutage, abhängig vom Tripel Store, meist ebenso über HTTP anwendbar), dann lassen sich Daten z. B. nicht nur lesen, sondern optional auch schreiben. Die Daten lassen sich vor allem durch elaborierte Weise abfragen und kombinieren, sogar über mehrere SPARQL Endpoints hinweg.[39] Beispiele hierzu liefert der Laborberichts-Abschnitt „Datenintegration".

Es spricht nichts dagegen, die basale API und die SPARQL-API durch andere APIs zu kapseln. Die Reduktion an Komplexität geht zwar Hand in Hand mit einem Verlust an Möglichkeiten, aber oftmals werden nur ein paar einfache Funktionen von Frontend-Entwicklerinnen benötigt und somit deren Arbeit erleichtert.

8 Laborbericht: Kataloganreicherung in lobid-resources

Der hbz LOD Service „lobid.org" existiert seit Mitte 2010.[40] Wie der Ausschnitt aus der LOD-Cloud von 2010[41] in Abbildung 1 zeigt, bestanden bereits zu Anfang Verlinkungen zu zwei anderen Datenquellen: zur GND (Verlinkung zu Personennormdaten und Schlagwörtern) und zum Linked-Data-Index kultureller Institutionen „lobid-organisations"[42], um Titel mit besitzenden Organisationen zu

37 Z. B. liefert http://thedatahub.org/api/rest/dataset/lobid-resources die Informationen über die lobid-Ressourcen in JSON zurück.

38 https://en.wikipedia.org/wiki/Web_API

39 http://www.w3.org/2009/sparql/wiki/Feature:BasicFederatedQuery

40 Zu Hintergrund, Motivation und zugrunde liegender Technik von lobid.org siehe Ostrowski / Pohl (2012).

41 Linking Open Data cloud diagram, erstellt von Richard Cyganiak and Anja Jentzsch, 2010-09-22: http://richard.cyganiak.de/2007/10/lod/lod-datasets_2010-09-22.png

42 http://lobid.org/organisation/. lobid-organisations ist ein internationales Linked-Data-Adressverzeichnis von Bibliotheken und verwandten Organisationen. 15.000 Einträge beschreiben Bibliotheken und Museen aus Deutschland. Das Datenset ist leider (noch) nicht Open Data, doch können die einzelnen Datensätze durchaus genutzt werden. U.a. enthält das Datenset Geo-In-

verbinden. Diese Links basieren auf den originär im hbz-Verbundkatalog vorhandenen Daten. Die Links zu den Organisationen leiten sich aus den Bestandsangaben im Verbundkatalog ab, wozu einem Titel die International Standard Identifier for Libraries and Related Organizations (ISIL) der Institutionen verspeichert ist, die ein Exemplar besitzen.

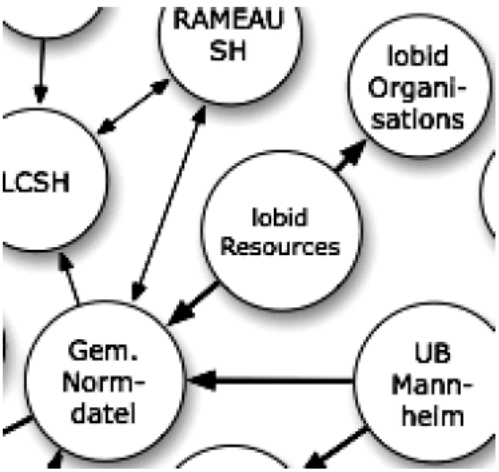

Abbildung 1: Ausschnitt aus der LOD Cloud 2010.

Seit Mitte 2012 ist lobid-resources zu zwölf anderen Datenquellen verlinkt.[43] Teilweise basieren die Links ebenfalls auf den originären hbz-Daten und wurden nur nicht von Anfang an in Form von HTTP-URIs verspeichert. Dazu gehören etwa die Links zur Dewey Decimal Classification und den LOC-Sprachcodes ISO 639-2. Teilweise sind erst nach Mitte 2011 Datensets als Linked Data publiziert worden (wie z.B. der B3Kat[44] und die ZDB[45]) und konnten folglich erst ab diesem Zeitpunkt verlinkt werden. Interessant sind hier die Links zu DBpedia, Projekt Gutenberg und der Open Library, denn diese wurden nachträglich erzeugt und sind nicht originärer Teil des Verbundkatalogs.

Die folgenden Kapitel erläutern Techniken zur Herstellung von Verknüpfungen zu Fremddaten und, anhand eines Beispiels, das in lobid.org tatsächlich

formationen. Damit lassen sich Geo-Suchen durchführen (sogar über SPARQL – siehe https://wiki1.hbz-nrw.de/pages/viewpage.action?pageId=5144604), und es lassen sich die Organisationen in einer Karte, z. B. OpenStreetMap, einblenden. Genutzt wird lobid-organisation momentan vom GBV (http://uri.gbv.de/organization/) und vom LODUM Projekt (http://lodum.de/post/28619267432/linking-bibliographic-resources) .

43 http://thedatahub.org/dataset/lobid-resources

44 Siehe dazu den Beitrag von Ceynowa et. al. in diesem Band.

45 http://www.zeitschriftendatenbank.de/services/schnittstellen/linked-data/

zum Einsatz kommende Verfahren mit der Software *Silk*. Es wird auf Matchingprobleme und mögliche Disambiguierungslösungen eingegangen. Im Anschluss werden Wege für die Portalintegration der lobid-Daten aufgezeigt. Anschließend wird der Gewinn durch – und das Potential von – verknüpften Daten angerissen.

8.1 Software zur Verknüpfung: Silk

Es existieren verschiedene Software-Werkzeuge um Datenverknüpfungen herzustellen, z. B. Google Refine[46], Silk[47] und culturegraph[48]. In lobid.org kam bisher lediglich Silk zum Einsatz. Nachfolgend werden einige Ergebnisse einer Anreicherung mit Daten der deutschen DBpedia dargestellt.[49]

8.1.1 Vorteile von Silk

Neben der Verarbeitung von RDF Dumps lassen sich mit Silk auch SPARQL Endpoints abfragen, d.h. es ist nicht notwendig, einen Vollabzug der nachzunutzenden Daten herunterzuladen. Somit sind auch Linked-Data-Quellen einfach verknüpfbar, die nicht mit der Open Knowledge Definition konform gehen. Ein großer Vorteil von Silk ist die sehr gute Dokumentation und die einfache Bedienung. Matchingalgorithmen werden hauptsächlich durch SPARQL-Abfragen definiert, zudem gibt es diverse „Linkage Rules“[50].

Silk gibt es in verschiedenen Varianten. Für kleinere Anreicherungen kann auf einfache Installationen zurückgegriffen werden. Für Anreicherungen über größere Datensets gibt es komplexere Cluster-Versionen. Zum Einsatz für die lobid Anreicherungen kam die hadoop-Variante.[51]

Die Konfiguration von Silk kann entweder durch Manipulation eines XML Files geschehen oder durch Verwendung einer Weboberfläche, der Silk Workbench. Für lobid.org Anreicherungen wurde nicht die Workbench verwendet, sondern direkt die XML Datei angepasst. Eine Beispieldatei folgt weiter unten.

---

46 https://code.google.com/p/google-refine/

47 http://www4.wiwiss.fu-berlin.de/bizer/silk/

48 http://culturegraph.sourceforge.net/

49 Für weitergehende Informationen wie z. B. das Silk Konfigurationsfile siehe Christoph, 2012. URL:https://wiki1.hbz-nrw.de/display/SEM/2012/05/03/First+results+using+SILK+to+link+to+DBpedia .

50 http://www.assembla.com/spaces/silk/wiki/Linkage_Rule

51 https://www.assembla.com/wiki/show/silk/Silk_MapReduce

8.1.2 Nachteile von Silk

Die Abfrage vom lobid-SPARQL Endpoint von 16 Millionen Ressourcen dauert 40 Stunden. Auch wenn ein einmal erzeugtes Binärfile durch einfaches Kopieren mehrmals benutzt werden kann, um z. B. einmal mit der deutschen DBpedia und danach mit der internationalen DBpedia usw. zu verknüpfen, so ist dies insgesamt sehr langsam und der offensichtliche Flaschenhals bei der Herstellung der Verknüpfungen. Zum Vergleich: Das Matching der Daten dauerte gerade einmal 4 Minuten.[52]

8.1.3 Konfiguration für das Matching

Die triviale Feststellung: „Es kann nur gematcht werden, was auch vorhanden ist," mündet in einen eher primitiven Matchingalgorithmus, dessen Ergebnisse[53] *nachträglich* mit weiteren Heuristiken verbessert werden müssen.[54]

Es folgt die Silk XML Konfigurationsdatei mit der die Verknüpfungen zur deutschen DBpedia erzeugt wurden:[55]

```
<DataSources>
 <DataSource id="lobid" type="sparqlEndpoint">
   <Param name="endpointURI" value="http://lobid.org/sparql/"/>
 </DataSource>
 <DataSource id="DBpedia" type="sparqlEndpoint">
   <Param name="endpointURI" value="http://de.DBpedia.org/sparql"/>
   <Param name="retryCount" value="100"/>
   <Param name="retryPause" value="400"/>
 </DataSource>
</DataSources>
<Interlinks>
```

52 Zum Zeitpunkt der Erstellung der Verknüpfungen gab es einen Bug in Silk, der nun behoben ist. Nun kann auch mit der hadoop Variante ein Filedump verwendet werden. Somit ist es nicht mehr notwendig, die Daten in einen SPARQL Endpoint zu laden. Damit entfällt dieser Flaschenhals. Die hier gemachten Zeitangaben dürften sich demnach bei Verwendung des Filedumps stark unterscheiden, so dass es sich lohnen wird, einen vorhandenen Dump zu nutzen. Der Dump wird allerdings direkt in das RAM geladen. Deswegen ist es zwingend erforderlich, genügend RAM zu haben.

53 Siehe Abschnitt „Verknüpfungsergebnisse".

54 Siehe Abschnitt „Postprozessierung".

55 Diese und weitere Silk-Konfigurationsdateien finden sich auf dem github Account von lobid unter https://github.com/lobid/silk-xml-configs.

```
<Interlink id="workManifested">
 <LinkType>rdrel:workManifested</LinkType>
 <SourceDataset dataSource="lobid" var="b">
  <RestrictTo>
   ?b rdf:type bibo:Book
  </RestrictTo>
 </SourceDataset>
 <TargetDataset dataSource="DBpedia" var="a">
  <RestrictTo>
    ?a dcterms:subject category:Literarisches_Werk
  </RestrictTo>
 </TargetDataset>
 <LinkageRule>
  <Aggregate type="max">
   <Compare metric="equality">
    <TransformInput function="lowerCase">
     <TransformInput function="replace">
      <TransformInput function="regexReplace">
       <Input path="?a/rdfs:label"/>
      </TransformInput>
      <Param name="search" value="_"/>
      <Param name="replace" value=" „/>
     </TransformInput>
    </TransformInput>
    <TransformInput function="lowerCase">
     <Input path="?b/isbd:P1004"/>
    </TransformInput>
   </Compare>
  </Aggregate>
 </LinkageRule>
 <Filter/>
</Interlink>
</Interlinks>
```

Der hier konfigurierte Matchingalgorithmus kann natürlichsprachlich wie folgt ausgedrückt werden:

Nimm alle Titel der deutschen DBpedia Ressourcen der Kategorie „Literarisches Werk", wandle alle Zeichen in Kleinbuchstaben, ersetze alle Unterstriche mit einem Leerzeichen und vergleiche diese Zeichenkette mit dem ebenfalls nach Kleinbuchstaben gewandelten Titel aller Bücher aus lobid. Wenn beide identisch sind, speichere die Beziehung als Tripel mit dem Prädikat rdrel:workManifested.[56]

56 Es ist evident, dass dieser Matchingalgorithmus zu primitiv ist und somit viele falsche Matches produzieren wird. Das anschließende Kapitel „Verknüpfungsergebnisse" geht auf die not-

8.1.4 Verknüpfungsergebnisse von Silk

Mit dem oben beschriebenen Algorithmus wurden ca. 28.000 Verknüpfungen zwischen lobid-Ressourcen und der DBpedia hergestellt. Da der Algorithmus zu primitiv ist, gibt es viele „false positives“: So haben z. B. die folgenden Ressourcen den gleichen Titel (dc:title) „Helden“, zeigen also laut dem oben beschriebenen Algorithmus alle auf denselben DBpedia URI, haben aber unterschiedliche Autoren:

http://lobid.org/resource/HT009535982
http://lobid.org/resource/HT013915133
http://lobid.org/resource/HT002957164
http://lobid.org/resource/HT003564841

Diese verschiedenen Ressourcen verlinken nun zu einem gemeinsamen Identifier (dem DBpedia URI) und könnten deshalb unter einem einzigen URI gebündelt werden. Dies wäre aber falsch, da die einzelnen Ressourcen *nicht* Manifestationen desselben Werks sind. Für eine Bereinigung ist eine Postprozessierung notwendig.

8.2 Postprozessierung

Eine Postprozessierung zur Disambiguierung baut auf einfachen bis komplexen Heuristiken auf. Sie kann bis zu einem gewissen Grad automatisiert erfolgen, doch sollte am Ende auch immer eine intellektuelle Überprüfung anstehen.[57] Eine einfache Heuristik ist: Ein Bündel von Manifestationen, die ein gleiches Werk identifizieren wollen, ist dann zu verwerfen, wenn die Ressourcen auf unterschiedliche Autoren verweisen.[58] Eine komplexere Heuristik wäre: Wenn ein Bündel mit z. B. 10 Manifestationen besteht, wobei 9 Manifestationen den Autor A haben und nur eine Manifestation den Autor B, dann wird nicht das gesamte Bündel verworfen, sondern lediglich die Manifestation mit Autor B, denn es wird angenommen, dass je bekannter ein Werk ist, umso mehr Manifestationen davon im Katalog vorhanden sind. Und ebenso, dass zumindest das bekannteste Werk in der DBpedia beschrieben ist (wenn zusätzlich auch noch die anderen – unbe-

wendige Postprozessierung ein.

57 Dies sollte idealerweise unter Mithilfe der Nutzer geschehen, also durch sog. „Crowdsourcing“.

58 Das Script zu dieser Disambiguierung liegt unter: https://github.com/lobid/linked-data-tools

kannteren - Werke beschrieben sind, ist das in der Wikipedia durch die sogenannte Disambiguierungsseite kenntlich gemacht).[59]

In lobid.org wurde bisher nur der oben beschriebene einfache Postprozessierungsalgorithmus angewendet. Dadurch schrumpfte die Anzahl der Links zur DBpedia von 28.000 auf nunmehr 6.000.

Um eine noch größere Sicherheit bei der Verknüpfung von Datensets zu erreichen, sollten die automatisch hergestellten Beziehungen (oder zumindest ein durch bestimmte Kriterien maschinell bestimmbares, „unsicheres" Subset dieser Beziehungen) intellektuell überprüft werden. Wenn auf intellektuelle Aufarbeitung nicht verzichtet werden sollte, wozu ist dann eine Automatisierung überhaupt sinnvoll? Ganz einfach: Zumindest der Schritt, der die Liste von möglichen Beziehungen herstellt, entfällt. Diese Liste manuell herzustellen, auf Grundlage gleicher Titel und für 16 Millionen Ressourcen, würde ein paar Dutzend Bibliothekare ein paar Jahre beschäftigen.[60]

Für diesen letzten Schritt, der intellektuellen Evaluierung, bietet sich supervisiertes Crowdsourcing an, also die durch z. B. einen Bibliothekar überprüfte Zusammenführung durch die Katalognutzer.

8.3 Speicherung

Im Folgenden wird gezeigt mit welchen Prädikaten die Verknüpfung zur DBpedia und Wikipedia geschieht, welche Arten der Anreicherung verwendet werden, wie die Provenienz vorgehalten wird, und wie die Daten abgefragt werden um sie z. B. in ein Portal integrieren zu können. (Um die Beispiele nachvollziehen zu können, ist es wichtig zu wissen, dass das Webfrontend unter lobid.org mittels Phresnel[61] gerendert wird. Phresnel ist auf Basis der Daten im Tripel Store konfiguriert und gibt nicht notwendigerweise alle im Tripel Store befindlichen Aussagen über eine Ressource wieder. Die Beispiele funktionieren also zumindest mit dem Tripel Store, aber nicht notwendigerweise über das lobid.org-Portal).

---

59 siehe „dbpedia-owl:wikiPageDisambiguates" unter z. B. http://de.dbpedia.org/resource/Helden

60 Gesetzt den Fall, ein Bibliothekar schafft manuell für 1000 Titel am Tag nachzuschlagen, welche Ressourcen den gleichen Titel haben. Daraus folgt: 16 Millionen / (200 Arbeitstage * 12 Bibliothekare * 1000 Titel pro Tag) = 6 Jahre.

61 https://github.com/lobid/Phresnel. Zur Funktionsweise von Phresnel siehe Ostrowski / Pohl (2012).

8.3.1 Vokabular

Es gibt eine reiche Auswahl an Verknüpfungsprädikaten, z. B. owl:sameAs[62] wenn es sich um identische Ressourcen handelt; foaf:isPrimaryTopicOf wenn die Zielressourcen lediglich das primäre Objekt der Katalogressource sind; rdf:seeAlso für eine eher unspezifische Verbindung u.v.m. Im vorliegenden Fall wird rdrel:workManifested verwendet. Zwar lässt sich argumentieren, dass eine Anwendung des WEMI-Modells hier problematisch ist, allerdings haben wir uns nach einiger Diskussion für diesen pragmatischen Ansatz einer FRBRisierung entschieden. Ein Wikipedia-Eintrag zu einem literarischen Werk beschreibt eben in der Regel die Werk-Ebene und nicht bestimmte Expressionen oder konkrete Manifestationen.

Da die DBpedia auf Grundlage der Wikipeda erzeugt wird, bedeutet eine Verknüpfung zu DBpedia immer auch zugleich eine Verknüpfungsmöglichkeit zur Wikipedia. Ein passendes Prädikat haben wir in der Music Ontology gefunden.[63] Auch wenn die Wikipedia nicht maschinenlesbar ist, so kann eine Verknüpfung durchaus Sinn machen, da hier mehr Informationen stehen als in der DBpedia extrahiert wurden und die Benutzerin kann sogar die Grundlage der Katalogan-reicherung, nämlich die DBpedia, durch Veränderung der Wikipediaartikel modifizieren.[64]

Es folgen zwei Tripel – als Beispiele zur Verknüpfung von lobid.org-Ressourcen zur DBpedia und Wikipedia:

<http://lobid.org/resource/HT014797039> rdrel:workManifested <http://de.dbpedia.org/resource/Gödel,_Escher,_Bach> .
<http://lobid.org/resource/HT014797039> mo:wikipedia <http://de.wikipedia.org/wiki/Gödel,_Escher,_Bach> .

62 Die owl:sameAs-Property ist im Semantic Web häufig fehlerhaft gebraucht worden. In Fällen der Verlinkens verschiedener Datenquellen gibt es gute Gründe gegen eine Verwendung von owl:sameAs und für das Erwägen von Alternativen. Siehe dazu Halpin / Hayes (2010), wo es etwa heißt (S.1): „Much of the supposed "crisis" over the proliferation of owl:sameAs in Linked Data can be traced to the fact that these uses of owl:sameAs tend to be mutually incompatible, and almost always violate the rather strict logical semantics of identity demanded by owl:sameAs."
63 http://musicontology.com/#term_wikipedia
64 Ein Phänomen von LOD ist die sog. „positive Rückkopplung": miteinander verknüpfte Daten sorgen potentiell auch für eine Verbesserung der Zielressource.

8.3.2 Arten der Anreicherung

Neben der Verspeicherung des Links ist es darüber hinaus auch lizenzrechtlich gestattet, alle Daten aller direkt in lobid-resources verlinkten Quellen in das eigene Backend zu integrieren. So könnte es z. B. zu einem Merge mit den Daten des B3Kat kommen, denn trotz Datenaustauschs der Verbünde untereinander („Ringtausch") und der beide Katalogeinträge als identisch definierenden Erstkatalogisierungs-ID (EKI), fehlt im hbz-Katalog teilweise die Verschlagwortung, die der B3Kat vorhält.

In lobid.org wurden die Daten der DBpedia-Matches bisher lediglich in den lobid.org Tripel Store eingespielt. Die Anreicherung des Suchmaschinenindex soll noch in 2012 geschehen.

8.3.3 Provenienz

In lobid.org lässt sich die Herkunft jedes Datums (aka Tripel) zurückverfolgen.[65] Um diese Provenienzinformationen zu liefern, werden vier Arten von Einheiten unterschieden:

1. die tatsächliche Ressource, das Objekt, sei es eine Organisation, ein Buch, eine Zeitschrift etc., z. B. http://lobid.org/resource/HT014797039.
2. der Datensatz, also das eine Ressource beschreibende RDF-Dokument, z. B. http://lobid.org/resource/HT014797039/about[66]
3. das Datenset, also z. B. die gesamten bibliographischen Daten des lobid-resources Datasets oder das im selben Tripel Store liegende Verzeichnis kultureller Institutionen „lobid-organisations"[67] (z. B. http://lobid.org/dataset/resource) und
4. der (Herkunfts-)Graph als Named Graph, dem Tripel aufgrund ihrer Herkunft (Verbundkatalog, DBpedia-Matching etc.) zugewiesen werden.

Das nebenstehende Bild veranschaulicht die Zusammenhänge zwischen den verschiedenen RDF-Graphen 2. bis 4.

---

65 Für mehr Hintergrund siehe https://wiki1.hbz-nrw.de/display/SEM/Provenienzinformationen.

66 Da das Objekt selbst (1.), für die der URI steht (z. B. ein Buch), meist nicht direkt über das Netz geliefert werden kann, wird per Content Negotiation eine Beschreibungsseite dieses Objektes (2.) geliefert: die Metadaten. Zu dieser Nutzung des 303-Redirect im Semantic Web siehe auch http://www.w3.org/TR/cooluris/#r303gendocument.

67 http://lobid.org/organisation

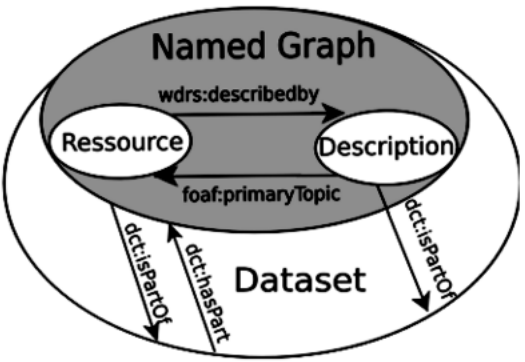

Abbildung 2: Zusammenhang Datensatz, Datenset und Named Graph

Anhand eines Beispiels wird im Folgenden gezeigt, wie sich die Provenienz ermitteln lässt.[68] Die Beschreibung beispielsweise einer bibliographischen Ressource enthält u.a. das folgende Tripel:

<http://lobid.org/resource/HT014797039> powder:describedby <http://lobid.org/resource/HT014797039/about> .

Der Subjekt-URI steht für die bibliographische Ressource, die durch die Metadaten – identifiziert durch den Objekt-URI – beschrieben wird. Da es bei den Provenienzinformationen ja um Informationen zu den Metadaten geht, müssen darüber entsprechende Aussagen getroffen werden, wie etwa:

<http://lobid.org/resource/HT014797039/about> dct:isPartOf <http://lobid.org/dataset/resource> .

Die Beschreibung ist also Teil des Datasets lobid-resources, das wiederum selbst genauer beschrieben wird (Lizenz, letzte Änderungen, benutzte Vokabulare etc.). Auch werden Aussagen über die Named Graphs gemacht, die Teil eines Datasets sind, z. B.:

<http://lobid.org/dataset/resource> dct:hasPart <http://lobid.org/graph/dbpedia-de>

Der Objekt-URI des vorhergehenden RDF-Tripels identifiziert einen Named Graph[69]. Für das Kernset von RDF-Tripeln, die aus dem Verbundkatalog generiert wurden, wie für die Ergebnisse einer jeden Datenanreicherung, existiert ein

68 http://www.w3.org/TR/cooluris/#r303gendocument
69 http://patterns.dataincubator.org/book/named-graphs.html

Named Graph. Die Metadaten zu einem Named Graph enthalten die Informationen über das „Wann – Wie - von Wem" usw., also die Provenienzinformationen über die Erstellung der Anreicherung, z. B.:

<http://lobid.org/graph/dbpedia-de/about> opmv:wasGeneratedBy _:process1 .[70]
_:process1 opmv:wasControlledBy <http://lobid.org/person/pc> .

Ein Named Graph kann wie ein Container verwendet werden: In dem DBpedia-Graph sind alle Tripel eingespielt worden, die die Titelkatalogressourcen mit der DBpedia verbinden. So kann auf diesen Daten via SPARQL unabhängig vom Rest der Daten im Tripel Store[71] gearbeitet werden, als handele es sich um eine separate Datenbank.[72] Somit lassen sich auch Aussagen über die Provenienz machen. Folgende Anfrage liefert alle Tripel der Beispielressource, die explizit aus dem deutschen DBpedia-Graphen kommen:

```
SELECT * FROM <http://lobid.org/graph/dbpedia-de> WHERE {
        <http://lobid.org/resource/HT014797039> ?p ?o .
}
```

Der hier gezeigte Weg Provenienzinformationen zugänglich zu machen, ist sicher nur einer von mehreren, zeigt aber die prinzipielle Möglichkeit, Provenienzangaben bis auf Tripelebene abzubilden und deutet die Wichtigkeit sowie die Möglichkeiten an, die sich damit eröffnen.[73]

8.3.4 Datenintegration via API

Prinzipiell ist das LOD-Netz schon die API, alle Daten lassen sich RESTful abholen. Das Antwortformat wird per Content Negotiation bestimmt und kann z. B. RDF-XML, Turtle, RDFa in HTML oder auch JSON sein. Hier ein Beispiel, um

70 Der Objekt-URI ist eine sog. „Blank Node", also eine Referenz, die es nur innerhalb des Tripel Stores gibt bzw. nur dort gültig ist.

71 Zum Einsatz kommt der Tripel Store „4Store", der, wie der Name schon sagt, auch ein sog. Quad-Store ist, also mit Named Graphs arbeiten kann.

72 Für ein komplexeres SPARQL Beispiel siehe den folgenden Abschnitt „Datenintegration".

73 Diese Form der Provenienzinformationen macht sich das Konzept der Named Graphs zunutze, das noch keinen W3C-Standard darstellt. Da der in lobid.org benutzte Tripel Store – 4store – eben mit Named Graphs umgehen kann, lag der Rückgriff darauf nahe. Detaillierte Erläuterungen und weitere Ansätze zur Provenienzinformation finden sich in Kai Eckerts Beitrag in diesem Band.

die Daten – unter Verwendung des Kommandozeilen-Programms cURL– so zu bekommen, wie sie das Phresnel Webfrontend präsentiert:

```
curl -L http://lobid.org/resource/HT016508950
```

Dasselbe als RDF-XML:

```
curl -H „Accept: application/rdf+xml“ -L
http://lobid.org/resource/HT016508950
```

usw. Um elaboriertere Abfragen zu gestalten, z. B. um alle Bücher zu erhalten, die mit dem Hugo-Award ausgezeichnet wurden, lässt sich folgende RESTFul Anfrage an den SPARQL Endpoint stellen:

```
curl -H “Accept: application/json” --data-urlencode ‘query=
SELECT ?slo WHERE {
    ?sdb <http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:Hugo_Award_for_Best_Novel_winning_works>.
    ?slo <http://rdvocab.info/RDARelationshipsWEMI/workManifested> ?sdb .
}’ http://lobid.org/sparql/
```

Das Antwortformat ist JSON und sollte sich einfach z. B. mittels Java-Script oder serverseitige Programme in eigene Portalanwendungen einbinden lassen.

Die hier durch curl gezeigten Beispiele lassen sich auch als Links für den Browser beschreiben, doch ist die Antwort dann immer in XML[74]. Für einen Anwendungsfall: „Ich schreibe eine E-Mail und möchte dort einen Link mitteilen der alle Ressourcen auflistet, die von Philip K. Dick geschrieben wurden“, sieht der Link wie folgt aus (es lassen sich natürlich beliebig komplexe, durch SPARQL beschreibbare Abfragen formulieren):

```
http://lobid.org/sparql/?query=SELECT%20*%20WHERE%20%7B?s%20%3Chttp://purl.org/
dc/elements/1.1/creator%3E%20%3Chttp://d-nb.info/gnd/174660774%3E%7D
```

Durch die Named Graphs lassen sich die abzufragenden Daten einschränken, z. B. kann über alle Graphen *außer* dem DBpedia Graphen gesucht werden um z. B. die DBpedia Daten aus einer Anwendung auszublenden. Ebenso kann über alle Daten des gesamten Tripel Stores gearbeitet werden, als ob es keine Container

74 http://www.w3.org/TR/rdf-sparql-XMLres/

gäbe. Beispiel für eine SPARQL Abfrage, die Ressourcen sucht, deren frbr:Work Ebene durch einen bestimmten Wikipedia-Eintrag beschrieben wird:

```
SELECT * FROM <http://lobid.org/graph/dbpedia> WHERE {
?s <http://rdvocab.info/RDARelationshipsWEMI/workManifested> <http://dbpedia.org/resource/Do_Androids_Dream_of_Electric_Sheep%3F> .
}
```

Trotz der relativen Einfachheit von SPARQL neigen Frontendprogrammierer zum Konsum von in der Mächtigkeit stark reduzierten Schnittstellen der folgenden Art:

```
http://lobid.org/ISBN-lookup/$isbn
```

Diese Schnittstellen geben als Antwort eine Aggregation in einem einfachen JSON-Format zurück. Solche Schnittstellen herzustellen ist allerdings nicht schwer: sie sind nichts weiter als kleine Proxies für die oben genannten RESTful SPARQL Abfragen mit einer zusätzlichen Wandlung in ein einfaches JSON.[75,76] Die Proxies könnten aber durchaus an Komplexität gewinnen, wenn über diese Zwischenschicht z. B. eine sich verändernde Datenbasis, etwa durch Umbenennung der Prädikate,[77] durch ein Mapping von Feldern auf die vom Konsumenten erwarteten Feldnamen begegnet werden würde. Dies würde zudem sicherlich Arbeit ersparen, wenn mehr als nur ein Konsument diese API verwendet, da sich der Mappingaufwand auf den zentralen Proxy beschränkt, statt in zahlreichen Anwendungen nachimplementiert werden zu müssen.

Da wir im konkreten Fall Linked *Open* Data vorliegen haben, kann prinzipiell jede Programmiererin diese einfachen Schnittstellen implementieren.

Dank Open Data lässt sich die Datenbasis ebenso gut auf die eigenen Server spiegeln und beliebig aufbereiten, z. B. für Suchmaschinen, und somit idealerweise in das eigene Datenökosystem integrieren, ganz ohne die Notwendigkeit

---

75 Siehe z. B. die Web Services der ZBW unter http://zbw.eu/beta/econ-ws/, die zum Teil auf SPARQL-Abfragen basieren.

76 Wünschenswert wären sicherlich kleine Web-Bausteine, die sich einfach in eine Webseite einbauen lassen, wie die sog. „Widgets" von LibraryThing.

77 Wegen der Offenheit und Flexibilität der Datenbeschreibung mittels RDF änderten neu entstandene LOD-Services öfter ihr Vokabular (auch wenn schon anfangs darauf geachtet wurde, Vokabulare zu nutzen, die möglichst weit verbreitet sind (wie etwa Dublin Core oder bibo)). Mittlerweile gibt es die Gruppe „DINI AG KIM Titeldaten" in Deutschland, um eine Art Kernelementeset zu beschreiben und damit eine stabile RDF Repräsentation für diese Kerndaten zu erreichen, siehe https://wiki.d-nb.de/display/DINIAGKIM/Titeldaten+Gruppe .

der Schaffung einfacher Schnittstellen. Nur weil die Daten durch LOD erzeugt wurden und als LOD zur Verfügung gestellt wurden, heißt das nicht, dass sie nicht in eine nicht-LOD konforme Form zu bringen und zu verwenden wären. LOD schließt das keinesfalls aus und eröffnet im Gegenteil eine breite Palette an Nachnutzungsszenarien.

Egal, ob LOD nun über LOD-Schnittstellen konsumiert wird, ob die Daten über einfachere Schnittstellen angeboten und konsumiert werden,[78] oder ob Daten direkt in eigene Datensysteme übertragen werden sollen: Da das Ermöglichen des Konsums der Daten der eigentliche Zweck von LOD ist, ist LOD ideal für Menschen, die Daten integrieren wollen.

9 Ausblick

> If Content is King, then Context is Queen.

Der Katalog ist für Bibliotheken wichtiger denn je. Das, was Jakob Voss über e-Ressourcen schreibt, gilt auch für Metadaten:

> Libraries that license eResources to be accessed from publisher sites, limit their role to temporary, intermediary retailers. Advice: Data that cannot be copied and modifed is lost. Libraries must actually collect and process digital documents (or won't be in the document business anymore).[79]

Ohne die Möglichkeit, fremde Metadaten frei zu kopieren und den eigenen Bedürfnissen nach anzupassen, bleiben diese Metadaten temporäre Artefakte. Zwar können Kataloganreicherungen, wie sie beispielsweise von LibraryThing und Amazon angeboten werden, einen momentanen Nutzwert bieten; die Form der Nutzung ist jedoch in der Regel von den Anbietern vorgegeben und damit beschränkt. Zudem steigt so die Abhängigkeit der Bibliotheken von Fremdanbietern. Bleibt der eigene Katalog ein „Rumpfkatalog", der lediglich mit geschlossenen Datentöpfen verbunden ist, führt das langfristig dazu, dass es letztendlich gar keines eigenen Katalogs mehr bedarf.[80] Dies kann durchaus gewünscht sein,

78 Siehe hierzu auch das einführende Kapitel „API" in diesem Beitrag.

79 Voss (2012). Online: http://de.slideshare.net/nichtich/libraries-in-a-datacentered-environment

80 Die Mindestvorraussetzung eines „Katalogs" ließe sich beschränken auf eine Liste von Identifiern, um auf die Fremddaten verweisen zu können, die dann dynamisch eingeblendet werden.

um, zumindest kurz- oder mittelfristig, Kosten zu sparen. Eine Konsequenz ist die Aufgabe von eigenen, selbst anpassbaren Suchmaschinen und letztendlich von eigenen, speziellen Portalfunktionalitäten. Bibliotheken wären dann nur noch Wieder"verkäufer" der Daten anderer: aggregierte Daten werden von lizenzierbaren Indexen eingekauft und durch ein Portal den Kunden zugänglich gemacht. Die Chance, selber als „Information Broker" aufzutreten oder *jedem* anderen die Möglichkeit dazu zu geben, wäre vertan. Ein unwahrscheinliches, aber nicht ganz unmögliches Szenario, ist: *Alle* Daten gelangen in geschlossene Datentöpfe, es entwickeln sich zentrale Gewaltstrukturen (also: Monopole), mit den daraus folgenden Kontrollmöglichkeiten (z. B. Präferierung im Suchmaschinenranking aufgrund von Verlagszugehörigkeit oder politischer Ausrichtung).[81] Dieser pessimistischen Möglichkeit der Entwicklung der Zukunft kann mit (Linked) Open Data, dem Teilen und der Anreicherung der Daten, begegnet werden.

Neben den direkten Verbesserungen von Katalogen durch Datenanreicherung, gibt es ein gewaltiges *indirektes* Potential. Dazu gehören z. B. die Schaffung einer frbr:Work-Ebene über eine Bündelung von Ressourcen, egal ob dies z. B. über die Verknüpfung mit der in Open Library vorhandenen frbr:Work-Ebene geschieht,[82] oder durch die interne Zusammenführung im eigenen Katalog (sog. Deduplizierung)[83]. Durch diese maschinelle Bündelung können prinzipiell die Titeldaten der verschiedenen Manifestationen voneinander partizipieren, um z. B. Schlagworte zu (ver)erben.

Werden, wie in der Einleitung angeklungen, die eigenen Katalogressourcen von anderen Benutzern konsumiert, z. B. in Literaturlisten oder Handapparaten (per RDFa in HTML[84] oder sogar in PDF via XMP[85]), dann lassen sich diese Daten rekonsumieren und an die eigenen Ressourcen anhängen, mit dem Effekt z. B. einen Impact Factor[86], Ähnlichkeitsbeziehungen, Empfehlungen und einen Social Graph[87] von Autoren berechnen zu können.

---

81 Auch diese Sicht, nämlich die Betrachtung gesamtgesellschaftlicher Auswirkungen von (Daten-)Lizenzierung, verbindet die Open-Data- mit der Open-Source-Bewegung, und so ist es auch kein Zufall, dass Richard Stallmann (Gründer und Präsident von http://gnu.org/ und http://fsf.org/) auf der Open Knowledge Conference 2011(http://okcon.org/2011) als Gastredner auftrat.

82 Siehe dazu auch http://www.slideshare.net/h_jansen/dynamische-kataloganreicherung-auf-basis-von-linked-open-data, Seite 17 ff.

83 Siehe dazu auch http://www.culturegraph.org/

84 Eine mit RDFa angereicherte Version dieses Artikels wird unter http://www.dr0i.de/lib/pages/Datenanreicherung_auf_LOD_Basis.html erscheinen.

85 https://www.adobe.com/devnet/xmp.html

86 https://de.wikipedia.org/wiki/Impact_Factor

87 https://en.wikipedia.org/wiki/Social_graph

Durch die LOD-Technik können Benutzer dem Datengraphen Informationen zufügen, ohne die Datenbank, auf die sie sich beziehen, direkt verändern zu müssen.[88] Zukünftig könnten deshalb Bücher aus der dem Internetbenutzer nächsten Bibliothek in Online-Fernsehzeitungen als Programmbegleitung angezeigt und die Bücher im Gegenzug mit dem Hinweis auf den sie erwähnenden Film versehen werden.

Es ließen sich auch noch weitere Beispiele aufführen. Ihnen gemeinsam ist die dem LOD inhärente „Win-Win-Situation".

Der Begriff *Wissensallmende*[89] bringt das Ziel von LOD auf den Punkt: Es wird davon ausgegangen, dass das Wissen, anders als natürliche Ressourcen, durch Gebrauch aufgewertet wird. Der Gebrauch von Daten wird sich erhöhen, wenn sich deren Nutzbarkeit und Nützlichkeit erhöht, wenn also die Daten maschinenlesbar, offen und mit anderen Daten verknüpft sind. Aus diesem Grund ist LOD ideal, um die Idee der *Wissensallmende* für Daten umzusetzen. Und da vor allem die Bibliotheken der Idee der Wissensallmende schon immer verpflichtet waren (oder sein sollten), sind sie intrinsisch motiviert, diese Technik zu nutzen.

10 Quellen

Alle verwendeten Links dieses Beitrages sind zuletzt geprüft worden am 11.9.2012.

Berners-Lee, Tim (2006): Linked Data. Online: http://www.w3.org/DesignIssues/LinkedData.html

Brenner, Simon (2012): LibraryThing for Libraries. Web Widgets für die Anreicherung von Bibliothekskatalogen mit Community-generierten Daten einer Social Cataloging-Plattform . Online: http://*www.b-i-t-online.de/heft/2012-03/fachbeitrag-brenner.pdf*

Christoph, Pascal (2012): First results using SILK to link to DBpedia. Online: https://wiki1.hbz-nrw.de/display/SEM/2012/05/03/First+results+using+SILK+to+link+to+DBpedia

Christoph, Pascal (2012): 1.2 M links to Open Library. Online: https://wiki1.hbz-nrw.de/display/SEM/2012/05/23/1.2+M+links+to+Open+Library

Dodds, Leigh; Davis, Ian (2012): Linked Data Patterns. A pattern catalogue for modelling, publishing, and consuming Linked Data. Online: http://patterns.dataincubator.org/book/

88 Ein existierendes Beispiel ist „LODUM". Dort werden Veröffentlichungen von Mitarbeitern der eigenen Universität (https://www.uni-muenster.de/forschungaz/) mit lobid-Ressourcen verknüpft. Dadurch gewinnen die LODUM-Ressourcen Bestandsangaben zu den Exemplaren in besitzenden Bibliotheken. Diese werden auf einer Landkarte visualisiert (siehe z.B. http://data.uni-muenster.de/context/cris/publication/41562). Im Gegenzug können die lobid-Ressourcen leicht mit den durch LODUM bereitgestellten Abstracts verknüpft werden.

89 https://de.wikipedia.org/wiki/Wissensallmende

Halpin Harry; Hayes, Patrick J. (2010): When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web. Online: http://events.linkeddata.org/ldow2010/papers/ldow2010_paper09.pdf.

Heath, Tom; Bizer, Christian (2011): Linked Data: Evolving the Web into a Global Data Space. Online: http://linkeddatabook.com/editions/1.0/

Jansen, Heiko; Christoph, Pascal (2012): Dynamische Kataloganreicherung auf Basis von Linked Open Data. Online: http://www.slideshare.net/h_jansen/dynamische-kataloganreicherung-auf-basis-von-linked-open-data

Koster, Lukas (2012): Discovery tools: a rearguard action? Online: http://de.slideshare.net/lukask/discovery-tools-a-rearguard-action

Kreutzer, Till (2011): Open Data – Freigabe von Daten aus Bibliothekskatalogen. Leitfaden von Dr. Till Kreutzer, i.e. - Büro für informationsrechtliche Expertise Berlin. Hbz. Online: http://www.hbz-nrw.de/dokumentencenter/veroeffentlichungen/open-data-leitfaden.pdf

Ostrowski, Felix; Pohl, Adrian (2012): Zur Entwicklung eines Linked-Open-Data-Dienstes für Bibliotheksdaten. Erscheint im Tagungsband zur WissKom 2012.

Pohl, Adrian (2012): Provenienzinformationen. Online: https://wiki1.hbz-nrw.de/display/SEM/Provenienzinformationen

Voss, Jakob (2007) : LibraryThing: Web 2.0 für Literaturfreunde und Bibliotheken, Mitteilungsblatt der Bibliotheken in Niedersachsen und Sachsen-Anhalt(137), Seite 12-13. Online: http://hdl.handle.net/10760/11077

Voss, Jakob (2012): Libraries in a data-centered environment. Online: http://de.slideshare.net/nichtich/libraries-in-a-datacentered-environment