X.media.press

Tassilo Pellegrini
Harald Sack
Sören Auer (Hrsg.)

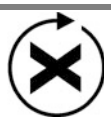
X.media.press ist eine praxisorientierte Reihe zur Gestaltung und Produktion von Multimedia-Projekten sowie von Digital- und Printmedien.

Linked Enterprise Data

Management und Bewirtschaftung vernetzter Unternehmensdaten mit Semantic Web Technologien

Springer Vieweg

X.media.press



X.media.press ist eine praxisorientierte Reihe zur Gestaltung und Produktion von Multimedia-Projekten sowie von Digital- und Printmedien.

Tassilo Pellegrini · Harald Sack · Sören Auer
Herausgeber

Linked Enterprise Data

Management und Bewirtschaftung vernetzter Unternehmensdaten mit Semantic Web Technologien

Springer

Herausgeber

Tassilo Pellegrini
Institut für Medienwirtschaft
Fachhochschule St. Pölten
St. Pölten, Österreich

Sören Auer
Institut für Informatik III
Rheinische Friedrich-Wilhelms-Univ. Bonn
Bonn, Deutschland

Harald Sack
Hasso-Plattner-Institut für
Softwaresystemtechnik GmbH
Universität Potsdam
Potsdam, Deutschland

ISSN 1439-3107
ISBN 978-3-642-30273-2 ISBN 978-3-642-30274-9 (eBook)
DOI 10.1007/978-3-642-30274-9

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media
www.springer.com

Vorwort der Herausgeber

Das World Wide Web ist im Begriff sich von einer weltweiten Sammlung vernetzter Dokumente hin zu einem Netzwerk verknüpfter Daten zu entwickeln. Dieses „Web of Data“ erlebt seit einigen Jahren ein immenses Wachstum. Es speist sich aus unzähligen Datenquellen unterschiedlichster Größe, Qualität und Themen, die entweder offen oder geschlossen zur Verfügung stehen und bereits jetzt vielerorts eine wichtige Komponente in der betrieblichen Datenverarbeitung darstellen. In Kombination mit dem weithin bekannten „Web of Documents“ bildet das „Web of Data“ ein neues Öko-System und eine grundlegende Infrastruktur für Software-Anwendungen und Dienste der Zukunft. Prominente Entwicklungen wie etwa „Big Data“, Open (Government) Data, Cloud Computing und Service-Orientierung sind Teilaspekte eines weitreichenden Wandels im Enterprise Data Management, in dessen Zentrum die Nutzung und Bewirtschaftung verteilter Daten steht.

Doch die stetig wachsende Verfügbarkeit qualitativ hochwertiger und strukturierter Daten sowohl innerhalb als auch außerhalb von Unternehmen veranlasst die Frage nach neuen Methoden und Technologien des Enterprise Data Managements. Konventionelle Datenbereitstellungsstrategien in Form von (semi-)strukturierten Dokumenten (z. B. HTML, CSV-Dateien) oder proprietären Programmierschnittstellen werden nur bedingt den Ansprüchen hoch vernetzter und dynamischer Daten-Ökosysteme gerecht. Mit jeder zusätzlichen Quelle steigen die Integrationsaufwände, Veränderungen in der Datenbankstruktur gehen oftmals zu Lasten der Systemintegrität und Aktualisierungen der Datenbasis sind nur mit hohem Aufwand in Echtzeit verfügbar.

Hier setzt der „Linked Data“ Ansatz an, der eine höchst-mögliche Flexibilität und technische Interoperabilität in der unternehmerischen Datenhaltung anstrebt und so die kosteneffiziente und zeitkritische Integrierbarkeit, eindeutige Interpretierbarkeit und Wiederverwendbarkeit von Daten ermöglicht. Linked Data bedient sich dazu sogenannter Semantic Web Standards, um existierende Datenbestände hoch strukturiert aufzubereiten und plattformunabhängig für die weitere Integration und Syndizierung bereitzustellen.

Mit diesem Band thematisieren die Herausgeber die Bedeutung neuer Formen des technologisch gestützten (Meta-)Daten-Managements für die voranschreitende Vernetzung und Integration verteilter, heterogener Datenbestände zur Unterstützung des betrieblichen Informations- und Wissensmanagements. Hierbei spielt vor allem der Einsatz von Se-

mantic Web Technologien, sowohl als Produktions- als auch als Distributionsinfrastruktur für umfassende Datensammlungen, eine zentrale Rolle. Denn durch Semantic Web Technologien werden Daten zu Netzgütern und erlauben neue Formen der Datenhaltung und Bewirtschaftung. Hierbei stellt sich die Frage, inwieweit die föderalen, selbstorganisierenden und kollaborativen Mechanismen des Webs in den kontrollierten Umgebungen einer Organisation sinnvoll zum Einsatz gebracht werden können, um neue Ressourcen aus bestehenden Informationsbeständen zu generieren und um von der Fülle an verfügbaren, qualitativ hochwertigen (offenen) Datenquellen im Web zu profitieren. Dazu diskutieren die einzelnen Beiträge technologische und methodische Aspekte des semantisch gestützten Datenmanagements und zeigen mittels Fallstudien existierender Implementierungen, wie eine gesteigerte Konnektivität und Interoperabilität von Datenquellen das Enterprise Data Management beeinflussen (werden).

Der Band „Linked Enterprise Data“ richtet sich an eine technisch versierte Leserschaft, die sich mit den Grundlagen und Anwendungsmöglichkeiten des Linked Data Prinzips für das betriebliche Informations- und Wissensmanagement vertraut machen möchte. Insbesondere adressiert der Band existierende und in Ausbildung befindliche Professionalisten, die als CTOs, CIOs, Enterprise Architects, Projektmanager und Applikationsentwickler in Unternehmen, Non-Profit Organisationen oder öffentlichen Einrichtungen arbeiten und mit Fragen der Skalierbarkeit, Flexibilität, Robustheit und Nachhaltigkeit von Informationssystemen befasst sind.

Der Band gliedert sich in drei Abschnitte. Der erste Abschnitt gliedert sich in Kap. 1 bis 3 und erläutert die technologischen Grundlagen und die geschäftsmodellrelevanten Aspekte des betrieblichen Einsatzes von Linked Data. Der zweite Abschnitt mit den Kap. 4 bis 9 diskutiert methodische Aspekte von Linked Data Technologien zur Lösung konkreter Probleme sowie technologische Entwicklungsmöglichkeiten im betrieblichen Datenmanagement. Der dritte Abschnitt mit den Kap. 10 bis 14 rundet die theoretischen Erläuterungen mit Fallstudien von konkreten Implementierungen von Linked Data Technologien ab.

Im Folgenden werden die Beiträge kurz vorgestellt.

In Kap. 1 erläutert Andreas Blumauer die Prinzipien von Linked Data in der Aggregation, Verwaltung und Bewirtschaftung von geschäftsrelevanten Datenquellen. Er zeigt, dass Linked Data basierte Datenmodelle weniger abstrakt sind als XML-Schemata oder relationale Datenbankmodelle und deshalb besser geeignet sind, Informationen für Mensch und Maschine verfügbar zu machen. Sogenannte „semantische Wissensgraphen“ oder Ontologien können hierbei modular und inkrementell entwickelt werden und mit den Geschäftsanforderungen flexibel mitwachsen. Linked Data Graphen tragen direkt zur Verbesserung der User-Experience bei und generieren Netzeffekte rund um interoperable Datenbestände. Der Beitrag führt diese Aspekte weiter aus und diskutiert vier Anwendungsfälle für den praktischen Einsatz von Linked Data im Unternehmen.

Kapitel 2 von Harald Sack bietet einen grundlegenden Überblick über das Thema Linked Data und führt in die dazugehörigen Basistechnologien ein. Nach der detaillierten

Erläuterung der Bedeutung und Funktion der eindeutigen Identifikation von Ressourcen in Wissensbasen wird in das Resource Description Framework (RDF) zur einfachen Modellierung von Fakten eingeführt. Linked Data lebt von der Verknüpfung der Fakten untereinander sowie mit zugrundeliegenden Wissensrepräsentationen in Form von Ontologien. In diesem Zusammenhang werden auch RDF(S) und OWL als formale Ontologiebeschreibungssprachen vorgestellt, um Möglichkeiten und Grenzen des Ansatzes aufzuzeigen. Weiterführend werden Möglichkeiten zur Nutzung von Linked Data in unternehmerischen Anwendungen vorgestellt sowie auf die Veröffentlichung eigener Datensätze als Linked Data eingegangen.

In Kap. 3 diskutiert Tassilo Pellegrini vor allem rechtliche Aspekte der Bewirtschaftung von vernetzten Daten entlang der Content Value Chain. Dies umfasst zum einen die Integration und Verwendung externer Daten im Zuge der Content-Verarbeitung, zum anderen die Wahl des richtigen Lizenzmodells für die Veröffentlichung eigener Daten als Linked Open Data. Ausgehend von unterschiedlichen Asset-Typen, die bei der Generierung von Linked Data anfallen, zeigt der Beitrag, welche Asset-Typen durch welches Rechtsinstrument geschützt werden können. Ein besonderes Augenmerk liegt auf der Kombination offener und geschlossener Lizenzinstrumente zu Zwecken der Diversifikation von Geschäftsmodellen.

In Kap. 4 gehen Sören Auer, Jörg Unbehauen und Rene Pietsch auf methodische Probleme der Integration verteilt vorliegender Unternehmensdaten ein. Sie argumentieren, dass Daten-Intranets auf Basis von Linked Data Technologien die existierenden Intranet- und SOA-Landschaften in großen Unternehmen erweitern und flexibilisieren. Hierbei bietet Linked Data die Möglichkeit der Nutzung von Daten aus der inzwischen auf über 50 Mrd. Fakten angewachsenen Linked Open Data (LOD) Cloud. Im Ergebnis kann ein unternehmensinternes Daten-Intranet, das sowohl interne als auch externe Quellen integriert, dazu beitragen die Brücke zwischen strukturiertem Datenmanagement (in ERP, CRM, SCM Systemen) sowie semi- und unstrukturierten Informationen (Dokumente, Wikis, Portale) der Intranet-Suche zu schlagen.

Diese Ausführungen werden durch Robert Isele in Kap. 5 vertieft. Sein Beitrag behandelt die notwendigen Prozesse, um eine globale Sicht auf mehrere Datenquellen herzustellen, sodass diese für eine gemeinsame Abfrage zur Verfügung stehen. Im Kern steht das Problem, dass Linked Data Publisher eine Vielzahl verschiedener Vokabulare verwenden um Informationen zu repräsentieren. Es gilt zunächst die Datensets in ein konsistentes Zielvokabular überzuführen und in einem zweiten Schritt, Ressourcen in unterschiedlichen Datensets, welche dasselbe Realwelt-Objekt repräsentieren, zu identifizieren und zu verknüpfen.

In Kap. 6 illustrieren Philipp Frischmuth, Michael Martin, Sebastian Tramp und Sören Auer am Beispiel der Anwendung OntoWiki aktuelle Ansätze in der Linked Data-Visualisierung und diskutieren deren Bedeutung im Enterprise Information Management. Die visuelle Aufbereitung von Linked Data sowohl für Zwecke der Prozessverarbeitung als auch zur Konsumierung durch Endanwender ist ein wichtiges Designelement in der

unternehmerischen Aneignung von Semantic Web Technologien, insbesondere im Zuge der Kuratierung und Qualitätssicherung verteilter Daten.

Philipp Cimiano und Christina Unger diskutieren in Kap. 7 das Problem der Multilingualität in verteilten Wissensbasen, die über Länder- und Sprachgrenzen hinweg erzeugt und genutzt werden. Die Autoren besprechen Verfahren, mit denen Datenschemata, die für verschiedene Länder entwickelt wurden, synchronisiert werden können, um die Aggregation und Integration von Daten über Länder und Sprachgrenzen hinweg zu ermöglichen. Darüber hinaus erläutern sie, wie Linked Data mit linguistischen Informationen angereichert werden kann, und betrachten einige Anwendungen, die zeigen, wie solche Informationen für die Generierung und die Interpretation natürlicher Sprache verwendet werden können.

In Kap. 8 beschäftigen sich Sebastian Bayerl und Michael Granitzer mit dem Einsatz von Linked Data Technologien im Data-Warehousing. Data-Warehousing bezeichnet die technologische Realisierung analytischer Datenbestände sowie entsprechender Schnittstellen zu deren Exploration und Analyse. Linked Data bietet vor allem mit der vor Kurzem begonnenen Entwicklung des RDF Data Cube Vokabulars neue Entwicklungsmöglichkeiten für Data-Warehousing Technologien und deren Einsatzspektrum. Der Beitrag stellt die Grundlagen zu Data-Warehouses vor und führt in das RDF Data Cube Vokabular als Linked Data Äquivalent ein. Beide Grundlagen dienen der Diskussion sowohl der Anwendung von RDF Data Cubes im Data-Warehousing als auch der Erweiterung traditioneller Data-Warehousing Ansätze, z. B. durch Integration offener Daten in Data-Warehousing Prozessen.

Kapitel 9 bietet einen kompakten Einstieg in das Thema Reasoning auf Basis strukturierter Daten zu Zwecken der automatischen Erschließung neuen Wissens aus oder der Qualitätssicherung von Datenbeständen. Dazu beschreiben Jens Lehmann und Lorenz Bühmann die Grundlagen des Reasonings in RDF/OWL-Wissensbasen und besprechen unterschiedliche Methoden des Reasonings. Weiters gehen sie auf Herausforderungen und Grenzen des Einsatzes von Reasoning-Technologien im Kontext von Linked Data ein.

In Kap. 10 beschreibt Anja Jentzsch die Linking Open Data Cloud, eine umfangreiche Sammlung von offen lizenzierten, vernetzen Daten und Kristallisationspunkt des Web of Data. Diese Data Cloud besteht aus mittlerweile 82 Milliarden RDF-Tripeln verteilt auf fast 1000 Datensätze, die vielfältige thematische Domänen abdecken. Der Beitrag analysiert ausgewählte Datensätze, welche im gemeinschaftlich gepflegten LOD Cloud Data Catalog eingetragen sind, und illustriert, wie diese Datensätze und deren Verlinkungen über die Linking Open Data Cloud visualisiert werden.

In Kap. 11 erläutern Natalja Friesen und Christoph Lange die Implementierung von Linked Data im Kontext Digitaler Bibliotheken. Wichtige Ziele bei der Entwicklung Digitaler Bibliotheken sind Informationen leicht auffindbar zu machen, sie miteinander zu verknüpfen, sowie die Inhalte der Bibliothek für Mensch und Maschine nutzbar zu machen. Dazu stellen die Autoren wichtige Standards, Vokabulare und Ontologien für bibliographische (Meta-)Daten vor und diskutieren Herausforderungen beim Publizieren Digitaler

Bibliotheken als Linked Data. Zu den Herausforderungen gehören Datenmodellierung, Mapping, sowie Verknüpfung der Daten miteinander und mit anderen Datenbeständen. Als konkrete Anwendungsfälle geben die Autoren einen Überblick über die Europeana und die Deutsche Digitale Bibliothek (DDB), stellen aber auch weitere Digitale Bibliotheken vor, die Linked Data einsetzen.

In Kap. 12 erläutern Michael Gorriz und Kai Holzweißig den Einsatz von Linked Data Technologien bei einem großen deutschen Autohersteller. Laut ihrer Argumentation erlauben Unternehmen in der Automobilindustrie gegenwärtig einen tiefgreifenden Wandel. Informationstechnologie bestimmt immer mehr die Art und Weise, wie Unternehmen arbeiten, und insbesondere die Entstehungsprozesse ihrer Produkte und Dienstleistungen. Kurzum: Die Idee des digitalen Unternehmens ist auch heute schon in traditionell geprägten Industriezweigen wie der Automobilindustrie zur Wirklichkeit geworden. Aufgrund der verschiedenen technologischen, organisationalen und kulturellen Herausforderungen, die dieser Paradigmenwechsel bedingt, bedarf es neuer Konzepte und Technologien, um diesen Wandel nachhaltig zu unterstützen. Im Rahmen des vorliegenden Artikels wird aufgezeigt, dass die Idee von Linked Data ein solches Konzept darstellen kann. Neben einer kurzen Diskussion der Grundlagen wird detailliert aufgezeigt, welche Anwendungsfälle und Mehrwerte für eine Praxisanwendung von Linked Data existieren.

In Kap. 13 diskutieren Harald Sack und Jörg Waitelonis Einsatz von Linked Data zur Verbesserung der Auffindbarkeit audio-visueller Information. Insbesondere Videodaten sind auf dem besten Wege zur bedeutendsten Informationsquelle im World Wide Web zu werden. Bereits heute werden pro Minute mehr als 100 Stunden Videomaterial von den Benutzern auf Videoplattformen wie YouTube eingestellt. Bei dieser gewaltigen Menge an unstrukturierten multimedialen Daten wird auch die gezielte Informationssuche immer schwieriger, da eine inhaltsbasierte Suche mit Hilfe von textbasierten Metadaten realisiert wird, die entweder manuell oder mittels unzuverlässiger automatischer Analyseverfahren gewonnen werden. Hier bietet die semantische Videosuche einen Ausweg, die aufbauend auf einer Vielzahl unterschiedlicher Analyseverfahren versucht, textbasierte Metadaten inhaltlich miteinander in Bezug zu setzen und zielsicher die gewünschten Ergebnisse zu finden. Darüber hinaus ermöglicht es den zu Grunde liegenden Suchraum, d. h. das gesamte Videoarchiv ähnlich dem Stöbern in einem gutsortierten Bücherregal zielstrebig zu durchmustern und auf diese Weise hilfreiche neue Informationen zu finden. Die Videosuchmaschine yovisto.com implementiert zahlreiche visuelle Analyseverfahren und kombiniert diese prototypisch in einer explorativen semantischen Suche.

Kapitel 14 beschließt den Band mit einer kompakten Darstellung des Einsatzes von Linked Data beim deutschen Fachverlag Wolters Kluwer Deutschland. Christian Dirschl und Katja Eck zeigen anhand von Businessanforderungen, wie sich die Wertschöpfungskette innerhalb eines Medienhauses unter Einbeziehung von Linked Data weiterentwickeln kann. Insbesondere die systematische Trennung von textlichem Content und Metadaten eröffnet völlig neue Möglichkeiten im Gesamtprozess. Die strukturierte Einbindung externer Wissensquellen stellt dabei ein nicht zu vernachlässigendes Potential dar. Die

dynamische Entwicklung in diesem Bereich erfordert die Analyse und Abschätzung der technischen Konzepte und Werkzeuge um mittel- bis langfristig neue wertschöpfende Geschäftsmodelle zu etablieren.

Wien/Potsdam/Bonn im Mai 2014 Tassilo Pellegrini, Harald Sack und Sören Auer

Inhaltsverzeichnis

Teil III Fallbeispiele

Teil I
Grundlagen

Linked Data in Unternehmen. Methodische Grundlagen und Einsatzszenarien

1

Andreas Blumauer

Zusammenfassung

Der Einsatz von Linked Data Technologien im Enterprise Data Management birgt vielschichtige Vorteile in der Aggregation, Verwaltung und Bewirtschaftung von geschäftsrelevanten Datenquellen. Linked Data basierte Datenmodelle sind weniger abstrakt als XML-Schemata oder relationale Datenbankmodelle und sind deshalb besser geeignet, Informationen für Mensch und Maschine in einem Modell zu verknüpfen. Semantische Wissensgraphen können hierbei modular und inkrementell entwickelt werden und mit den Geschäftsanforderungen flexibel mitwachsen. Linked Data Graphen tragen direkt zur Verbesserung der User-Experience bei und generieren Netzeffekte rund um interoperable Datenbestände. Der Beitrag führt diese Aspekte in weiteren Details aus und diskutiert vier Anwendungsfälle für den praktischen Einsatz von Linked Data im Unternehmen.

1.1 Einleitung

Das Konzept von *Linked Data* ist eine praktische Umsetzung des *semantischen Webs*. Die Grundidee dafür geht zurück auf Sir Tim Berners-Lee. Berners-Lee, Direktor des World Wide Web Konsortiums (W3C), hat bereits in seinem Grundsatzpapier „Information Management: A proposal“ [8] Ende der 1980er Jahre eine Entwicklungsstufe des Webs skizziert, in dem Informationsbausteine und Prozesse mit Hilfe smarter Software-Agenten automatisch verlinkt werden. Seither wurde die Entwicklung eben dieses smarteren Webs unter der Schirmherrschaft des W3C vorangetrieben. Dazu wurden zahlreiche Spezifikationen und Standards entwickelt und veröffentlicht, die nun die Grundlage für

A. Blumauer ✉
Semantic Web Company GmbH, Mariahilfer Straße 70, 1070 Wien, Österreich
e-mail: a.blumauer@semantic-web.at

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_1

ein weitreichendes Spektrum an Linked Data Technologien bilden, das von Daten- und Wissensmodellierung über graph-basierte Abfragesprachen bis hin zum automatischen Reasoning reicht.

Neben ihrer technischen Fundierung sind Entwicklungen, vor allem im Umfeld des Internet, dann nachhaltig und zukunftsweisend, wenn sich auf Basis offener Standards auch eine breite Community aus Software-Entwicklern, Beratern und Business-Developern etablieren kann, die die Vorteile ihrer Produkte letztlich auch gegenüber der Industrie demonstriert. Mit dem W3C im Kern wuchs eine solch weltumspannende Community heran, die 10 Jahre nach ihrer Initiierung zu einem Software- und Dienstleistungsmarkt herangereift ist. Das Semantic Web konnte in akademischen Kreisen Fuß fassen und hat sich in unterschiedlichsten Branchen und Industrien als Lösungsmethode für diverse Herausforderungen im Daten-, Informations- und Wissensmanagement etabliert.[1] Semantic Web Technologien ziehen damit in den Alltag ein: sowohl, um Arbeits- und Produktionsprozesse effizienter zu gestalten, als auch bei Entscheidungen oder der Aneignung von Wissen zu unterstützen.

In diesem Überblicksartikel soll zunächst die Idee, auf die sich der Begriff Linked Data bezieht, vermittelt werden. Damit sollen auch LeserInnen angesprochen werden, die sich bislang mit dem semantischen Web bzw. Linked Data nur gelegentlich befasst haben. Daran angeknüpft werden Anwendungsszenarien beschrieben, die den Mehrwert von Linked Data plastisch vor Augen führen. Abschließend wird auf den aktuellen Entwicklungsstand und auf Zukunftsperspektiven von *Linked Enterprise Data* eingegangen.

1.2 Linked Data, Semantic Web, Web of Data – eine Kurzdarstellung

Die Begriffe „Semantische Technologien“ und das „Semantische Web“ werden oft synonym gebraucht, obwohl wesentliche Unterschiede bestehen: Geht es in beiden Fällen darum, Informationen *und* ihre Bedeutung zu verarbeiten, so dienen semantische Technologien der (meist automatischen) Bedeutungserschließung, wohingegen das Semantic Web die bedeutungstragenden Elemente verknüpft und inhaltlich kontextualisiert. Im Semantic Web dreht sich alles um die Frage, wie Entitäten (Produkte, Organisationen, Orte, etc.) sinnvoll zu so genannten *Wissensgraphen* (oder Linked Data Graphen) verwoben werden können. Die zugrundeliegenden Linked Data Technologien setzen dabei auf dem Paradigma der größtmöglichen *Interoperabilität* durch *offene Standards* auf.

Parallel zum allgemein bekannten *Web of Documents* (als Sammlung von HTML-Files), dessen wesentliches Merkmal Hypertext ist, entwickelt sich also ein *Web of Data* (als Sammlung von RDF-Daten), in dem anstelle von Dokumenten Entitäten unterschied-

[1] Über diesen Band hinausgehend dokumentiert insbesondere die englische Literatur zahlreiche Anwendungsbeispiele etwa bei [11], [19] und [20]. Siehe auch die Use Case Sammlung des W3C: http://www.w3.org/2001/sw/sweo/public/UseCases/, aufgerufen am 22.02.2014.

licher Kategorien (z. B. Orte, Organisationen, Produkte, Themen, Personen, . . .), Bezeichnungen, Attribute und deren Relationen zueinander verwaltet werden. Das *Web of Data* wird damit zu einer weltweit verteilten, aber hochgradig vernetzten Datenbank.

Die dem *Web of Data* zugrundeliegenden Design-Prinzipien, wie sie von Tim Berners-Lee [9] definiert wurden, bestehen aus vier Regeln und sind ebenso trivial wie effektiv. Die Prinzipien lauten:

1. Verwende URIs (kurz für: Uniform Resource Identifiers) um Entitäten (Dinge oder Resources) zu bezeichnen.
2. Verwende http-URIs, damit Software-Anwendungen und auch User auf diese Entitäten einfach zugreifen können.
3. Wird eine URI aufgerufen, so liefere nützliche Informationen zurück und verwende dabei offene Standards (RDF, SPARQL).
4. Biete dabei auch Links auf andere URIs an, damit weitere Dinge entdeckt werden können.

Ein kleines Beispiel soll die eben besprochenen (rekursiven) Linked Data Prinzipien veranschaulichen:

1. Tim Berners-Lee hat (unter anderem) die URI http://dbpedia.org/resource/Tim_Berners-Lee
2. Diese URI und damit zahlreiche Fakten zur entsprechenden Entität können von Software-Anwendungen als auch mit einem einfachen Browser aufgerufen werden.
3. Das Resultat ist maschinenlesbar und Standard-basiert. Es werden relevante Fakten in strukturierter Weise retourniert, u. a.:
   - Tim Berners-Lee wurde am 08.06.1955 in London geboren.
   - Tim Berners-Lee heißt auch „TimBL“.
   - Tim Berners-Lee ist der Direktor des „World Wide Web Consortiums (W3C)“.
4. Das W3C hat wiederum eine URI http://dbpedia.org/resource/World_Wide_Web_Consortium, auf die in Folge verlinkt wird, usw.

Abbildung 1.1 zeigt eine Teilansicht des Linked Data Graphen, der sich rund um die URI http://dbpedia.org/resource/Tim_Berners-Lee aufspannt.

Unter Berücksichtigung dieser Linked Data Prinzipien wurde im Jahr 2006 das *DBpedia-Projekt*[2] ins Leben gerufen. Als semantische Version der Wikipedia bildet die DBpedia den Nukleus der stetig wachsenden *Linked Open Data Cloud* (LOD Cloud)[3], eines gigantischen Wissensgraphen, der auch zunehmend kommerziell genutzt wird.

[2] Die öffentlich zugängliche Datenbasis ist mit Stand Februar 2014 in 119 Sprachen verfügbar. Siehe http://dbpedia.org/About, aufgerufen am 22.02.2014.
[3] Siehe http://datahub.io/de/group/lodcloud, aufgerufen am 22.02.2014.

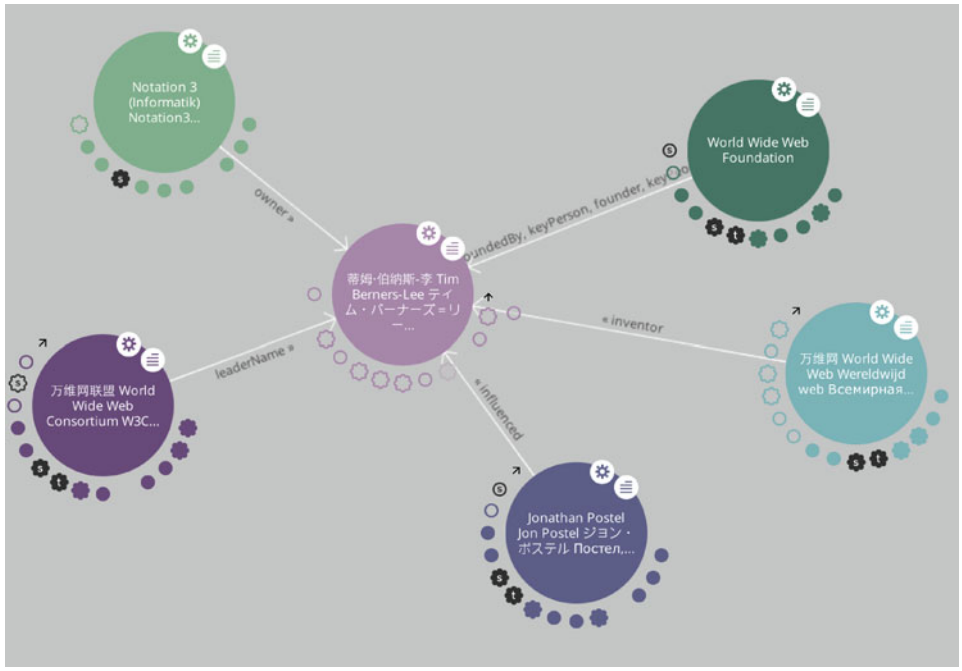


Abb. 1.1 Linked Data Graph zu Tim Berners Lee erstellt mit LodLive (http://www.lodlive.it/), am 19.03.2014

1.3 Linked Data im Enterprise-Einsatz

Unternehmen organisieren Informationen traditioneller Weise mit Hilfe von Informationstechnologien wie relationalen Datenbank-Managementsystemen (RDBMS) oder Dokumenten-Managementsystemen (DMS). Damit werden einerseits strukturierte und andererseits unstrukturierte Informationen gespeichert. Strukturierte Informationen, wie z. B. Kundenstammdaten, folgen einem explizit vorliegenden Datenbank- oder Metadaten-Schema, in dem bereits große Teile der semantischen Information codiert sind. Unstrukturierte Informationen, wie z. B. Gesprächsprotokolle oder Nachrichtentexte, verfügen über keine oder kaum gesondert ausgewiesene Schemata und „funktionieren" auf Basis der Regeln einer natürlichen Sprache wie Deutsch oder Englisch.

Eine wesentliche Idee von Linked Data ist es, dass Daten und Informationen unterschiedlichster Herkunft und Struktur auf Basis von Standards interpretiert, (weiter-) verarbeitet, verknüpft und schließlich dem User in einer Form präsentiert werden können, sodass dieser seine Aufwände zur Informationsgewinnung und -aufbereitung verringern kann.[4] Dementsprechend unterstützen Linked Data Technologien die Datenintegration

[4] Eine umfassende Darstellung des Linked Data Lifecycles findet sich bei Auer et al. [2].

mittels eines ausdrucksstarken Datenmodells, dem so genannten *Resource Description Framework (RDF)*[5], das als Integrationsschicht für die unterschiedlichsten Repräsentationsformen (relationale Datenbanken, XML, natürlich sprachlicher Text etc.) dient. Dazu müssen sowohl Syntax als auch Semantik der zu verknüpfenden Informationen mit Hilfe von Wissensgraphen (u. a. kontrollierte Vokabulare und Ontologien) aufeinander abgestimmt werden.

Gegenüber traditionellen, oftmals auf XML basierenden Techniken zur Datenintegration können zumindest folgende sechs Nutzenargumente angeführt werden, die für den Einsatz von Linked Data sprechen.

1.3.1 Linked Data basierte Datenmodelle sind weniger abstrakt als XML-Schemata oder relationale Datenbankmodelle

Datenbankmodelle werden von Informatikern für Techniker, z. B. für Softwareentwickler formuliert. Menschen verwenden jedoch weder Tabellen, Primärschlüssel noch Normalformen, um ein Modell der Realität zu entwickeln oder sich etwas zu merken. Mit konventionellen Methoden wird Fachwissen – technisch bedingt – oft auf eine Weise repräsentiert, sodass ab einem gewissen Komplexitätsgrad die Nachvollziehbarkeit der semantischen Beziehungen kaum noch gewährleistet ist und spätere Änderungen und Erweiterungen am Modell nur mit hohen Aufwänden möglich sind. Der nachhaltige Nutzen eines Daten- bzw. Wissensmodells, unabhängig vom erfassten Fachbereich, beruht jedoch auch auf seiner Nachvollziehbarkeit und Adaptionsfähigkeit. Hier setzt der graphenbasierte Ansatz des Semantic Web an.

Abstraktes Wissen (oft auch Schema-Wissen genannt) kombiniert mit konkreten Fakten (Faktenwissen) kann in ein semantisches Netz verwoben und auf Basis entsprechender Linked Data Graphen repräsentiert werden. Abstraktes Wissen könnte sich auf einfache Regeln beziehen, z. B. dass Länder stets eine Hauptstadt haben oder Hotels einen Ort. Die Tatsache hingegen, dass der Ort „Wien“ die Hauptstadt von Österreich ist, wird zu den konkreten Fakten gezählt. Beide Arten von Wissen können mittels dem bereits erwähnten Resource Description Framework (kurz: RDF) bzw. mittels RDF-Schema ausgedrückt werden. Es entstehen in Folge unzählige Wissensbausteine (so genannte *RDF-Statements oder RDF-Triples*), die explizit und zunächst unabhängig von jeglicher Anwendung vorliegen können. Ein Triple könnte z. B. ausdrücken, dass eine „Organisation“ einen „Direktor“ hat und, daran in einem weiteren Triple geknüpft, dass „Tim Berners-Lee“ „Direktor von“ „W3C“ ist. Zur besseren Veranschaulichung könnten diese Zusammenhänge wie in Abb. 1.2 visualisiert werden.

Das Prinzip Wissen, Fakten und Modelle von den später darauf zugreifenden Software-Anwendungen zu entkoppeln, führt also dazu, dass diese auch für Nicht-Informatiker verständlicher und einfacher zugänglich werden.

[5] Siehe http://www.w3.org/RDF/, aufgerufen am 22.02.2014.

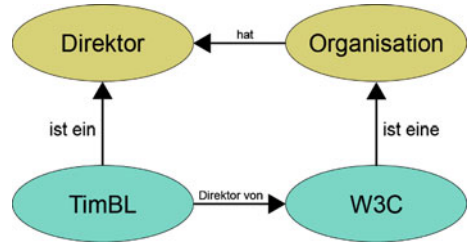


Abb. 1.2 Beispiel für einen einfachen Wissensgraphen bestehend aus vier Triples

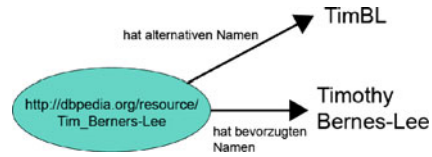


Abb. 1.3 Entitäten, URIs und ihre Labels

1.3.2 Linked Data basierte Datenmodelle verknüpfen Informationen für Mensch und Maschine in einem Modell

Linked Data basiert (im Gegensatz zu relationalen Datenbanken oder XML-basierten Datenstrukturen) auf Graphen, in denen beliebige Entitäten bzw. Ressourcen (z. B. Produkte, Orte, Personen etc.) miteinander semantisch verknüpft werden. Alle Entitäten sind für eine Maschine stets eindeutig über ihren Uniform Resource Identifier (URI) adressierbar. Die abstrakten Zeichenketten der URIs werden für den Menschen mittels „Labels“ (Namensauszeichnungen) lesbar gemacht, wie in Abb. 1.3 veranschaulicht.

Labels können jedoch mehrdeutig sein und sich potentiell auf verschiedene Entitäten beziehen (so genannte „Homonyme“) – ein häufig zitiertes Beispiel hierfür ist „Java“, das sowohl eine Insel, als auch eine Programmiersprache bezeichnen kann. Gibt nun jemand einen Begriff wie „timbl“ in eine herkömmliche Suchmaschine ein, so werden auch Dokumente ausgegeben, die vom „Tilburg Memory Based Learner“ (kurz: TiMBL) handeln. Der Linked Data Ansatz erlaubt durch die Zuordnung eindeutiger URIs die unterschiedlichen Bedeutungen von Labels abzugrenzen. So können Mehrdeutigkeiten bereits während des Indexierens berücksichtigt und leichter abgefangen werden bzw. können Eingabehilfen für den Endanwender bereitgestellt werden.

1.3.3 Semantische Wissensgraphen können modular und inkrementell entwickelt werden und mit den Anforderungen flexibel mitwachsen

Die Rigidität von Schemata in herkömmlichen, relationalen Datenbanksystemen rührt noch von einer Zeit her, als Computersysteme mit einem Bruchteil der heute üblichen Ressourcen und deutlich langsameren CPUs in einer angemessenen Zeit Abfragen über Datenbanken ausführen mussten. Performance wurde „erkauft“, indem das Datenbank-Design zumeist auf wenige spezielle Anwendungsfälle hin optimiert und fixiert wurde.

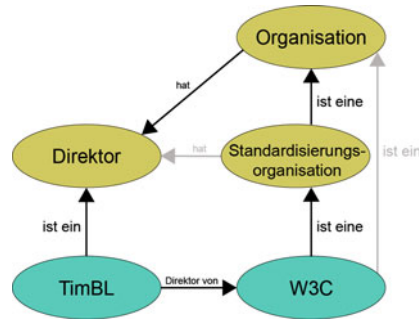


Abb. 1.4 Beispiel dafür, wie ein Wissensgraph Schritt für Schritt erweitert werden kann

Änderungen daran waren hingegen schwierig und wurden tunlichst vermieden, und zwar auch dann, wenn sich die modellierte Realität längst verändert hatte. Diese Starrheit im technischen System hat sich aber nicht nur auf die Geschäftsprozesse, sondern sogar auf das Denken der Software-Entwickler und Datenbank-Manager übertragen: Änderungswünsche an IT-Systemen von Seite der Fachabteilungen, die bis auf die Datenbankebene reichen, werden auch heute noch von IT-Abteilungen gerne als „problematisch“ eingestuft, um es vorsichtig auszudrücken.

Mit dem Aufkommen von leistungsstarken NoSQL- und speziell Graph-Datenbanken, die mit entsprechenden Rechnerleistungen, vor allem aber enormen RAM-Kapazitäten im Server, erst möglich wurden, stellt sich nun allmählich ein Umdenken ein: Daten- und Datenbankmodelle werden als flexibel und verhältnismäßig unaufwändig an die repräsentierte Realität anpassbar gedacht.

Ein Beispiel für die Flexibilität: Als Betreiber eines Informationsportals zu den Themenkreisen „Semantic Web“, „Data & Text Analytics“ und „Big Data“ greifen wir den Wissensgraphen aus Abb. 1.4 auf. Wir fügen eine neue Kategorie von Organisationen ein, nämlich „Standardisierungsorganisation“, um entsprechende Abfragen bzw. Suchfilter zu ermöglichen.

In einem weiteren Triple kann nun die Tatsache hinzugefügt werden, dass das W3C nicht nur einfach eine Organisation ist, sondern eben eine Standardisierungsorganisation.

Dieses Wissensmodell kann also beliebig erweitert werden, sowohl auf Schema- als auch auf Faktenebene, ohne dabei bereits bestehende Software-Anwendungen grundsätzlich zu beeinträchtigen.

Der Wissensgraph aus Abb. 1.2 ist also entsprechend erweitert worden.

Die grau gekennzeichneten Kanten im Graphen und die entsprechenden Fakten (Tripples) können mit Hilfe von automatischem Reasoning und der entsprechenden Ontologie hergeleitet werden:

- Da eine Standardisierungsorganisation eine Organisation ist und jede Organisation einen Direktor hat, hat auch eine Standardisierungsorganisation einen Direktor.
- Da das W3C eine Standardisierungsorganisation ist, ist es auch eine Organisation (im Allgemeinen).

Für zahlreiche Anwendungen, insbesondere in dynamischen Wissensdomänen, benötigen wir Datenmodelle mit einer höheren Flexibilität als jener von relationalen Datenbankmodellen. Linked Data Graphen können ähnlich wie das Strom- oder Straßennetz mit den Anforderungen mitwachsen. Das zugrundeliegende Resource Description Framework (RDF) und RDF-Schema (RDFS), die Web Ontology Language (OWL) bzw. die Abfragesprache SPARQL, jeweils vom W3C standardisiert, bilden dafür die technische Grundlage.[6]

1.3.4 Mit Linked Data können sowohl strukturierte als auch unstrukturierte Informationen semantisch erfasst und verknüpft werden

In vielen Unternehmen wird der Großteil des entscheidungsrelevanten Wissens aus unstrukturierten Informationen gewonnen, z. B. aus E-Mails, Pressemitteilungen oder aus Gesprächsprotokollen. Wenn diese Informationen nun mit möglichst einfachen Mitteln, z. B. mit Fakten aus Produkt-Datenbanken verknüpft werden können, so werden tiefgreifende Analysen zur Wettbewerbssituation oder zu aktuellen Marktentwicklungen möglich.

Linked Data Warehouses, wie z. B. der PoolParty Semantic Integrator[7], können sowohl Daten aus relationalen Datenbanken erfassen, als auch Daten und Fakten aus beliebigen Arten von Texten, um diese beiden Welten schließlich miteinander in Kombination abfragbar zu machen. Abbildung 1.5 veranschaulicht die grundsätzliche Funktionsweise eines solchen Systems.

Eine fundamentale Methode, die dies ermöglicht, beruht auf der automatischen Extraktion von Entitäten bzw. Konzepten aus Texten (Named Entity Recognotion). So könnte das Dokument mit dem Textfragment „Der Erfinder des World Wide Web, Tim Berners-Lee, ist Direktor des W3C“ gemäß dem Wissensgraphen aus Abb. 1.4 analysiert und annotiert werden. Die extrahierten Entitäten sind in vielen Fällen Personen, Organisationen, Produkte oder Orte. In unserem Fall würde der Text mit den beiden Entitäten „Tim Berners-Lee“ und „World Wide Web Consortium“ annotiert bzw. verknüpft werden. Das Dokument kann demzufolge auch automatisch mit den Kategorien „Standardisierungsorganisation“ und „Direktor“ in Verbindung gebracht werden. Diese automatisierbare Umwandlung von unstrukturierten Texten in Linked Data Graphen bildet die Grundlage, um in Folge komplexe Abfragen über Content-Repositorien mit unterschiedlichsten Namen und Bezeichnern bzw. Strukturen (Metadaten- und Kategorisierungssystemen) absetzen zu können.

[6] Für einen Überblick über Semantic Web Standards siehe http://www.w3.org/standards/semanticweb/, aufgerufen am 22.02.2014.

[7] Für einen Überblick über den PoolParty Semantic Integrator siehe http://www.poolparty.biz/portfolio-item/semantic-integrator/, aufgerufen am 22.02.2014.

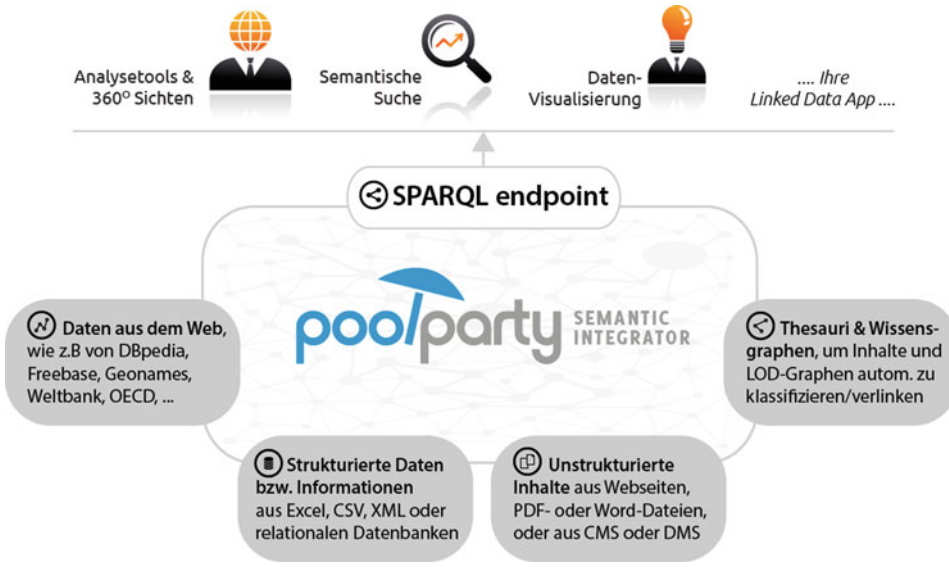


Abb. 1.5 High-Level Architecture eines Linked Data Warehouses, in diesem Beispiel des PoolParty Semantic Integrators

1.3.5 Linked Data besitzt eine mächtige Abfragesprache: SPARQL

SPARQL (kurz für SPARQL Protocol and RDF Query Language)[8] ist ein weiterer wesentlicher W3C-Standard im Semantic Web Stack, mit dem hoch expressive Abfragen von RDF-Graphen ermöglicht werden.

Mit SPARQL können ähnlich komplexe Abfragen wie mit für Business Intelligence Anwendungen typischen OLAP-Würfel ausgeführt werden. Was mit SPARQL nur in wenigen Zeilen formuliert werden kann, würde mit SQL oft nur mit deutlich mehr Aufwand zu bewerkstelligen sein. Die Tatsache, dass mit SPARQL Graphen und nicht Tabellen abgefragt werden, führt dazu, dass z. B. kürzeste Verbindungen zwischen Knoten in einem Netzwerk mit einer einzigen Abfrage ausgeführt werden können.

Seit Veröffentlichung dieses Standards im Jahre 2009 haben sich kaum abweichende SPARQL-Dialekte entwickelt. Dies ermöglicht einerseits die einfachere, herstellerunabhängige Portierung von Daten zwischen RDF-Stores, andererseits können Daten, die sogar in unterschiedlichen Datenbanken gespeichert sind, mit Hilfe von föderierten SPARQL Abfragen „on-the-fly“ und mit einer einzigen Abfrage zusammengefasst und erschlossen werden.

[8] Siehe http://www.w3.org/TR/rdf-sparql-query/, aufgerufen am 24.02.2014.

1.3.6 Linked Data Graphen können direkt zur Verbesserung der User-Experience beitragen

Anwendungen können oftmals nur dann reibungslos bedient werden, wenn dem User das Daten- bzw. Wissensmodell geläufig ist, das der Applikation zugrunde liegt. Mit Hilfe von Linked Data Technologien werden Wissensmodelle leichter zugänglich gemacht. Aus technischer Sicht genügt die Bereitstellung eines SPARQL-Endpoints, um Bedienungselemente wie z. B. Auto-Complete und Auswahloptionen wie Suchfacetten gemäß eines Anwendungs- und User-Kontexts anzubieten.

Wer von uns kennt nicht die prekäre Situation, in der der gewünschte Zug in wenigen Minuten abfährt und der Fahrkartenautomat nur dann das passende Ticket auswerfen würde, wäre man an das Vorhaben mit größerem Vorwissen herangegangen? Ein semantischer Wissensgraph könnte hier – so wie bei jeglichen Suchanwendungen – die erforderlichen Hilfestellungen bereitstellen.

Google's Knowledge Graph (vgl. Abb. 1.6) ist wohl eines der aktuell am sichtbarsten Beispiele, die anschaulich vor Augen führen, wie Linked Data Technologien die User-Experience einer Suchanwendung steigern können. Dabei wird versucht, Suchanfragen in Entitäten, also Ressourcen im Sinne von RDF zu übersetzen. Ist dies erst einmal gelungen, so werden zur initialen Suchanfrage weiterführende Fakten und daran geknüpfte, vertiefende Suchanfragen in einer „Factbox" zusammengefasst. Welche Arten von Fakten angezeigt werden, ist dabei abhängig vom Typ der Ressource. Handelt es sich z. B. um eine Musikgruppe, so werden populäre Musik-Alben der Künstler als Kontextinformation mit ausgegeben, sucht man z. B. nach einer Person wie Tim Berners-Lee, so werden manche seiner Bücher oder seine Eltern als weiterführender Knoten im Wissensnetz zum Anklicken angeboten.

1.3.7 Netzwerkeffekte

Da Linked Data auf einer Vielzahl von offenen Standards beruht, die von Modellierungssprachen wie RDF-Schema und OWL[9], über darauf beruhende Vokabulare und Ontologien wie SKOS (Simple Knowledge Organization System)[10] oder GoodRelations[11] bis hin zur Abfragesprache SPARQL reichen, lassen sich von und für Daten-Provider genauso wie für Endnutzer signifikante Netzwerkeffekte erzeugen.

Primäre Netzwerkeffekte lassen sich dadurch erzielen, dass die Kosten der Informationsintegration und -vernetzung durch standard- und graph-basierte Repräsentation von Information sinken, wobei der Wiederverwendungswert der Information signifikant steigt.[12]

[9] Siehe http://www.w3.org/2001/sw/wiki/OWL, aufgerufen am 24.02.2014.
[10] Siehe http://www.w3.org/2004/02/skos/, aufgerufen am 24.02.2014.
[11] Siehe http://www.w3.org/wiki/WebSchemas/GoodRelations, aufgerufen am 24.02.2014.
[12] Beschreibungen beider Aspekte finden sich bei Cranford [12] und Mitchell & Wilson [16]. Aus Perspektive der Enterprise Search siehe Benghozi & Chamaret [7].

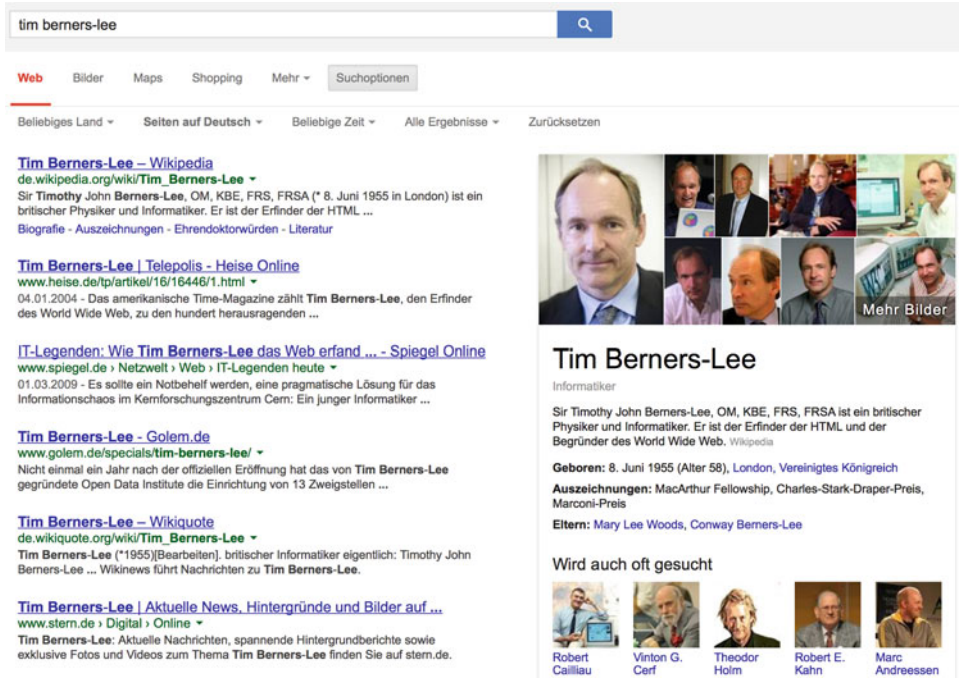


Abb. 1.6 Suche auf Google – die Factbox, rechts neben den herkömmlichen Suchresultaten, wird von Google's Knowledge Graph abgeleitet

Eine anschauliche Entwicklung, die auf diesen Umstand zurückgeführt werden kann, ist das stete Anwachsen der so genannten „Linked Open Data Cloud". Die LOD Cloud[13], die sich zunächst aus zahlreichen, herausragenden Datenbanken und Informationsdiensten wie Wikipedia (bzw. ihrer „semantischen Schwester" DBpedia)[14], Geonames.org[15] oder dem CIA Factbook[16] konstituiert hat, konnte die branchenübergreifende Produktion von Linked Data in unterschiedlichsten Organisationen stimulieren. Prominente Beispiele von Linked Open Data Anbietern sind öffentliche Einrichtungen wie die Europäische Union, die Britische Regierung, Bibliotheken wie die Deutsche Nationalbibliothek oder die Library of Congress, sowie große Medienhäuser wie Wolters Kluwer, New York Times oder BBC.

Aus Sicht von Unternehmen, die auf diesen Zug aufspringen wollen, heißt dies, dass mit jedem Triple, das zur LOD Cloud hinzugefügt wird, der potentielle Wert einer eigenen Linked Data Infrastruktur zunimmt. Dies heißt aber nicht notgedrungen, dass Unterneh-

[13] Siehe http://datahub.io/de/group/lodcloud, aufgerufen am 24.02.2014.
[14] Siehe http://dbpedia.org/About, aufgerufen am 24.02.2014.
[15] Siehe http://www.geonames.org/, aufgerufen am 24.02.2014.
[16] Siehe https://www.cia.gov/library/publications/the-world-factbook/, aufgerufen am 24.02.2014.

men ihre Daten als Linked *Open* Data veröffentlichen müssen. Es können auch interne Linked Data Warehouses um Daten aus der LOD Cloud mit vergleichsweise geringen Aufwänden angereichert werden ohne eigene Daten offenlegen zu müssen.

1.4 Anwendungsszenarien von Linked Data in Unternehmen

Es können zumindest drei grundlegende Szenarien für den unternehmerischen Einsatz von Linked Data unterschieden werden:

1. **Linked Data als Datenintegrationsprinzip anwenden**
   Das Unternehmen verwendet die Linked Data Prinzipien und Semantic Web Technologien intern, um Datenintegration und Mashups (z. B. für ein Wissensportal) zu realisieren bzw. neue Möglichkeiten von semantischer Suche zu erschließen. Dies ist grundsätzlich in allen Geschäftsprozessen bzw. Fachabteilungen von Interesse, in denen durch integrierte Sichten über umfassende, oftmals heterogene und verteilte Datenbestände fundiertere Entscheidungen bei kürzeren Recherchezeiten ermöglicht werden.
2. **Daten aus der Linked Data Cloud einbinden**
   Das Unternehmen konsumiert Daten aus der Linked Data Cloud, um damit z. B. interne Datenbanken oder Inhalte anzureichern. Ein einfaches Beispiel dazu: Werden in den Helpdesk eingehende E-Mails um Geodaten (z. B. von Geonames.org) angereichert, so kann eine Kartenvisualisierung dynamisch erzeugt werden, die anzeigt, aus welchen Regionen zu einem Zeitpunkt die häufigsten Störmeldungen gemeldet werden. Dies kann z. B. für Mobilfunkbetreiber bzw. Stromversorger von Interesse sein.
3. **Daten in die Linked Data Cloud publizieren**
   Das Unternehmen publiziert eigene Daten und Inhalte in die Linked Data Cloud und erschließt sich damit neue Distributionswege und Verwertungskanäle für digitale Assets. Zu diesen Assets gehören in Folge neben Inhalten und Instanzdaten auch die Metadaten, Vokabulare und Wissensmodelle, mit denen diese organisiert werden. Eine klug versionierte Veröffentlichung und Teilverwertung der Metadata-Assets unter kombinierter Verwendung offener und geschlossener Lizenzmodelle ist vor allem für Medienunternehmen oder medienähnlich agierende Unternehmen ein neu aufkeimendes Betätigungsfeld.

Die in Folge detailliert dargestellten Anwendungsfälle orientieren sich an den eben vorgestellten drei Szenarien.

Anwendungsfall 1: Enterprise Search basierend auf Linked Data Enterprise Search funktioniert grundlegend anders als die Suche über Internet-Inhalte. Anders als im WWW kann die Relevanz eines Suchergebnisses nicht auf Basis des Verlinkungs-Grades eines Dokuments berechnet werden, da Firmen-Intranets bei weitem weniger Link-Strukturen

aufweisen als das Internet. Umso wichtiger wird daher die semantische Analyse jedes zu indizierenden Dokuments mit Hilfe linguistischer Verfahren und mit Verfahren des Text-Minings. Damit kann das System nicht nur besser „verstehen“, welche Inhalte ein Dokument hat, sondern auch lernen, wie Begriffe, Phrasen bzw. Entitäten (Orte, Personen, Produkte, Branchen etc.) eines Unternehmens zueinander in Beziehung stehen. Somit können Suchmaschinen z. B. Personen als Experten für gewisse Produkte oder Branchen identifizieren und zusätzlich zu relevanten Dokumenten ausgeben.

Aufbauend auf den eben aufgezählten Grundeigenschaften liegt der Kern eines leistungsstarken Such-Systems für das unternehmerische Umfeld also in der Möglichkeit, Texte und ihre Bedeutung analysieren zu können und mit Hilfe intelligenter Suchdialoge bzw. -assistenten durchsuchbar zu machen. Ein wesentliches Element dabei ist das Erkennen und Extrahieren jener Entitäten (Geschäftsobjekte), die für ein Unternehmen von besonderer Bedeutung sind. Dazu zählen zumeist Produktnamen, Unternehmen (Kunden, Partner, Tochter- und Schwesterfirmen), Projekte, Personen usw. Sind diese erst einmal jedem Dokument zuordenbar, können komplexere Suchanfragen abgesetzt werden, die zumeist schon einem Frage-Antwort-System nahe kommen, z. B.: Wer ist Ansprechpartner für ein bestimmtes Produkt? Oder welche Projekte wurden am Standort X in einem gewissen Zeitraum durchgeführt?

Das Informationsbedürfnis eines Mitarbeiters, der in einer wissensintensiven Branche komplexe Aufgaben zu bewältigen hat, entspricht in vielen Fällen einer Beratungssituation. Nicht eine singuläre Suchanfrage wie „Pizza Wien“, was für die Suche im Web typisch ist, sondern eine Abfolge an Fragestellungen ist zu unterstützen. Diese „moderierte Suche“, die mit Hilfe von Such-Assistenten wie Facetten-Suche und dem so genannten „Drill Down“ ermöglicht wird, gehört zu den aktuellen Features einer Suchmaschine, die auf dem Stand der Zeit ist.

Suchmaschinen der neuesten Generation können nicht nur einzelne Entitäten erkennen und extrahieren, sondern sogar Fakten und Aussagen, und diese können zueinander in Beziehung gesetzt werden. Zum Beispiel wird erkannt, dass Person X in einem gewissen Zeitraum der CEO eines Unternehmens Y war, und weiters, dass dieses Unternehmen Y in den Jahren zuvor das Tochterunternehmen einer Firma Z war usw. Dokumente werden also mittels der extrahierten Entitäten und Fakten mit dem Linked Data Graphen verknüpft, wodurch auch unstrukturierte Informationen reichhaltig kontextualisiert und via der Abfragesprache SPARQL zugänglich gemacht werden können.

Anwendungsfall 2: Mitarbeiterportal Mitarbeiterportale sind wesentlicher Bestandteil eines Wissensmanagement-Systems und bieten für jeden Mitarbeiter vor allem bei der Informationsbeschaffung einen zentralen Anlaufpunkt. Ob nun eine datenbank- und anwendungsübergreifende integrierte Sicht auf die betriebliche Informationslandschaft am Portal erzeugt werden kann, hängt davon ab, ob Doppelgleisigkeiten beim Aufbau von Referenz- und Identifikations-Systemen vermieden werden können.

Ein Beispiel dazu: Wird in der einen Datenbank von „Kunde“ gesprochen, in der anderen aber vom „Klienten“, so beziehen sich zwar beide Bezeichner auf dasselbe Geschäftsobjekt, jedoch bleibt der Maschine diese Beziehung verborgen. Eine übergreifende Suche nach allen Kunden oder die ganzheitliche Sicht auf einen Kunden ist damit nicht möglich. Ausweg aus dieser in der Praxis häufig anzutreffenden Situation kann wiederum ein URI-System bieten: Jedes Geschäftsobjekt ist via Uniform Resource Identifier (URI) eindeutig gekennzeichnet und adressierbar.

Damit ist die Basis zur Entwicklung kontextsensitiver, „mitdenkender“ Widgets für ein Mitarbeiterportal gelegt: Inhalte, die von Mitarbeitern eingestellt werden und über ein Tagging-System annotiert werden, das auf Basis eines SKOS-basierten Thesaurus funktioniert, können mit anderen Inhalten aus dem Intranet intelligent verknüpft werden. So kann z. B. die Suche nach ähnlichen Inhalten realisiert werden, was auch dabei helfen kann, das Rad nicht stetig neu zu erfinden, Doppelarbeiten zu vermeiden und weiterführende Quellen zu erschließen.

Anwendungsfall 3: Content Augmentation Content Augmentation bezeichnet jenen Vorgang, in dem Inhalte, die von Autoren oder Mitarbeitern z. B. im Rahmen eines Enterprise-Content-Management-Systems erstellt werden, mit anderen Inhalten angereichert werden. So können mittels Geo-Daten übersichtliche Kartendarstellungen eingebunden und mit weiterführenden sinnvollen Kontextinformationen kombiniert werden. Die Zusatzinhalte stammen aus Internetquellen wie Wikipedia, aus Nachrichtendiensten, Fachdatenbanken oder aus statistischen Zeitreihen sensorgesteuerter Echtzeitdaten – womöglich in Form von Open Data.

Dies kann einerseits für den User bedeuten, dass dieser gewinnbringende Zusatzinformationen ohne weiteren Rechercheaufwand beziehen kann, andererseits können diese zusätzlichen Daten dazu dienen, die Inhalte mit weiteren Metadaten aufzuwerten, was wiederum zu einer effizienteren Inhaltserschließung führen kann und insbesondere im Kontext von Big Data Anwendungen erfolgskritisch ist.[17]

Anwendungsfall 4: Market Intelligence Mit Hilfe integrierter Sichten und mittels Content Augmentation, der zielgerichteten Anreicherung von Dossiers mit Inhalten aus dem Web oder aus anderen Datenquellen, können u. a. folgende Market-Intelligence-Funktionen unterstützt werden:

1. *Prognosefunktion und Trend Scouting*
   Chancen und Entwicklungen werden durch gezieltes Web-Mining frühzeitig aufgedeckt und antizipiert. Veränderungen des marktrelevanten Umfelds können besser abgeschätzt und deren Auswirkungen auf das eigene Geschäft durch semantisches Trend Mining aufgezeigt werden.

[17] Siehe dazu etwa den McKinsey Report zu Big Data [15].

2. *Unsicherheitsreduktionsfunktion durch verbesserte Kontextualisierung*
   Durch die Präzisierung und Objektivierung von Sachverhalten bei der Entscheidungsfindung wird eine typischerweise schlecht strukturierte Problemstellung besser beherrschbar.
3. *Selektionsfunktion*
   Relevante Informationen können aus der Flut umweltbedingter Informationen besser ausgewählt werden.

1.5 Zusammenfassung und Ausblick

Als kennzeichnende Elemente einer anhaltenden Entwicklung, in welcher das traditionelle Web of Documents mit einem Web of Data verknüpft wird, sind zusammenfassend zu nennen:

1. Die Linked Data Initiative des W3C, das ein einfaches Framework, bestehend aus vier Regeln entwickelt hat, um eine weltweite, verteilte Datenbank, das „Web of Data“ zu realisieren [9].
2. Die Übersetzung der Wikipedia in maschinenlesbares Semantic Web Format unter Berücksichtigung der Linked Data-Prinzipen als Nukleus für ein „Web of Data“ [3]. Die daraus resultierende DBpedia ist in der Zwischenzeit in 119 Sprachen verfügbar und bildet den Nukleus der stetig wachsenden „Linked Open Data Cloud“ (LOD Cloud).
3. Die Verwendung von Uniform Resource Identifier (URIs) aus der LOD Cloud, um in Kombination mit automatischen Text-Extraktionsverfahren Web-Dokumente um Metadaten anzureichern, die im Sinne des Semantic Web quellenübergreifend referenzierbar sind. Dieses Grundprinzip macht auch Google für sich nutzbar, indem auf Basis des Google Knowledge Graphs Webinhalte indiziert und verknüpft werden. Damit können beliebige unstrukturierte Informationen als semantischer Graph repräsentiert werden. Werden URIs aus der LOD Cloud verwendet, also z. B. von DBpedia.org, so werden Inhalte aus dem WWW und in weiterer Folge auch aus dem Corporate Web besser verlinkbar und vergleichbar.
4. Das ursprüngliche „Henne-Ei-Problem“, ohne Semantic Web Daten gibt es keine entsprechenden Anwendungen und so fort, wird nicht nur durch Wrapper und Extraktions-Frameworks [14] überwunden, sondern auch durch die zunehmende Verbreitung der Semantic Web Standards über gebräuchliche Plattform- und Datenbank-Technologien wie Drupal, Wordpress oder MarkLogic, die Metadatenformate wie RDF immer stärker ins Zentrum ihrer Architektur rücken. Parallel dazu propagieren auch Suchdienste wie Google zunehmend die Verwendung von Linked Data Standards wie RDFa oder JSON-LD [18].
5. Die BBC als ein europäisches Leitunternehmen hat 2008 schließlich mit „BBC Music beta“ ein Linked Data Projekt vorgestellt, das aufzeigt, welche neuartigen Verwertungsstrategien für Medienunternehmen mit Hilfe des Semantic Web möglich wer-

den [13]. Die Plattform reichert eigene Informationen um Ressourcen aus MusicBrainz und der Music Ontology an und kann damit nicht nur Mashups mit Wikipedia oder MySpace automatisch generieren, sondern bietet so auch neue kostengünstige Möglichkeiten für andere Plattformen an, um Inhalte der BBC einzubinden. Die BBC fühlte sich nach dem Erfolg dieses Projektes veranlasst, den Einsatz von Linked Data Technologien auszuweiten und setzte u. a. damit das Informationsportal der Olympischen Sommerspiele 2012 in London um [5].

6. Dem BBC-Beispiel folgten zahlreiche weitere Unternehmen, u. a. andere Konzerne aus der Medien- und Verlagsbranche wie Wolters Kluwer oder Elsevier, aber auch Betriebe aus anderen Branchen wie der Automobilindustrie, der Pharmaindustrie oder der öffentlichen Verwaltung [4]. Insbesondere öffentliche Einrichtungen wie Ordnance Survey (UK), die Europäische Union, die Weltbank oder Bibliotheken wie die Deutsche Nationalbibliothek tragen immer mehr zur Verbreitung von Daten auf Basis von Linked Data Standards bei. Das Semantic Web hat also begonnen, Einzug in diverse Branchen und Industrien zu halten.

Obwohl sich das neuartige Gebiet *Linked Enterprise Data* zunächst auf semantische Lösungen für die Probleme in den kontrollierten IT-Umgebungen der Unternehmen konzentriert, ist dies zugleich auch eine wichtige Grundlage, um mit der Zeit den Fokus zu erweitern und Entwicklungen hervorzubringen, die über Unternehmensgrenzen hinweg vernetzt auf globale Dimensionen eines Public Semantic Web skalieren. Eine Schlüsselrolle in diese Richtung nimmt dabei der Linked Data Ansatz ein, der es erlaubt, das Unternehmenswissen mit externen Daten (Linked Open Data Clouds) anzureichern bzw. als „Linked Open Data“ anderen zur Verfügung zu stellen.

Eine vollständig integrierte Sichtweise auf ein Corporate Semantic Web kann dann gelingen, wenn ein Unternehmen als Organisation begriffen wird, die Inhalte, Prozesse und Informationen nicht nur innerhalb der Unternehmensgrenzen produziert und einsetzt, sondern im Sinne eines vernetzten Unternehmens im Ökosystem Internet agiert. Interne und externe Inhalte sinnvoll und kostenschonend zu verknüpfen, kann nur in einem interoperablen Framework wie dem Semantic Web gelingen. Die umfangreiche Nutzbarmachung von Linked Data Technologien für den kommerziellen und industriellen Einsatz ist damit weniger eine Frage der Technologie als vielmehr ihrer organisationalen Verankerung. Entsprechend werden folgende Fragestellungen an Bedeutung gewinnen:

- *Geschäftsmodelle im Semantic Web*
  Wie können Wertschöpfungsmodelle entwickelt werden, die sich vom Rohdaten-Lieferanten bis hin zum Endkunden erstrecken und über geeignete Daten-Transformationen hin zu Linked Data darauf aufbauende Mehrwert-Dienste und Mashups ermöglichen [6]? Wie können im Zusammenhang damit entsprechende Lizenz- und Preismodelle entwickelt werden [17]?

- *Herkunft der Information („Provenance")*
  Im Zuge der Mehrfachverwertungsmöglichkeiten digitaler Inhalte und den damit verbundenen Strategien, Inhalte als Mashups oder in Form von Gadgets in Informationsportale einbinden zu können, wird die Frage umso virulenter, woher die Informationsbausteine stammen, welche Qualitätsmerkmale und Vertrauenswürdigkeit sie aufweisen und welchen Lizenzbestimmungen sie unterliegen. Ansatzpunkte dazu bieten voiD, ein Vokabular zur Beschreibung von Linked Data Quellen [1], bzw. das Vokabular PROV[18], eine vom W3C entwickelte Spezifikation zur Beschreibung von Herkunftsinformationen speziell von Linked Data.
- *Qualitätssicherung im Semantic Web*
  Während im *Web of Documents* gilt, dass es pro Webseite zumindest eine Autorität gibt, die auch die Qualität der Inhalte zu verantworten hat, so ist im *Web of Data* und den damit verbundenen Möglichkeiten, „Webseiten" als Mashups von Daten unterschiedlicher Provenienz zu konzipieren, die Qualitätssicherung bei weitem komplexer. Linked data Curation entlang eines Linked data Lifecyles gewinnt an Bedeutung [3].
- *Skalierbarkeit*
  Waren die ersten Semantic Web Datenbanksysteme und deren RDF Triple Stores darauf ausgelegt, dass zunächst ein „Proof-of-Concept" erfolgen konnte, so wurden in den letzten Jahren RDF Graph-Datenbanken immer performanter, um schließlich in Umgebungen mit umfassenden Datenbeständen (bis zu 150 Milliarden Triples) auch komplexe Abfragen in akzeptabler Zeit ausführen zu können [10]. Im Zuge voranschreitender Big Data Anwendungen werden Linked Data Prinzipien und deren kreative Beherrschung zu einem integralen Bestandteil des betrieblichen Datenmanagements.

Literatur

1. Alexander, Keith et al. 2009. Describing linked datasets: on the design and usage of void, the vocabulary of interlinked datasets. In *Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09)*
2. Auer, Sören et al. 2007. *DBpedia: A nucleus for a web of open data. The Semantic Web*, 722–735. Berlin Heidelberg: Springer
3. Auer, Sören, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N. Mendes, Bert van Nuffelen, Claus Stadler, Sebastian Tramp, und Sebastian Williams. 2012. Managing the Life-Cycle of Linked Data with the LOD2 Stack. In *The Semantic Web – ISWC 2012. Proceedings of the 11th International Semantic Web Conference* Boston, MA, USA. Lecture Notes in Computer Science, Bd. 7650, 1–16
4. Baker, Thomas et al. 2012. *Semantic Web Case Studies and Use Cases*. http://www.w3.org/2001/sw/sweo/public/UseCases/

[18] Siehe http://www.w3.org/TR/prov-o/, aufgerufen am 24.02.2014.

5. Bartlett, Oliver 2013. *Linked Data: Connecting together the BBC's Online Content*. http://www.bbc.co.uk/blogs/internet/posts/Linked-Data-Connecting-together-the-BBCs-Online-Content
6. Bauer, Florian et al. 2011. *Linked Open Data: The Essentials*. Vienna: Edition mono/monochrom
7. Benghozi, Pierre-Jean, und Cécile Chamaret. 2010. Economic Trends in Enterprise Search Solutions. In *JRC Scientific and Technical Report, EUR 24383 EN – 2010*, Hrsg. Ramón Compañó: European Commission Joint Research Centre: Institute for Prospective Technological Studies
8. Berners-Lee, Tim 1989/2002. Information Management: A proposal. In *Multimedia. From Wagner to Virtual Reality*, Hrsg. Randel Packer, und Ken Jordan, 208–224. New York: Norton & Company
9. Berners-Lee, Tim 2006. *Linked Data*. http://www.w3.org/DesignIssues/LinkedData.html
10. Boncz, Peter et al. 2013. *Berlin SPARQL Benchmark Results for Virtuoso, Jena TDB, BigData, and BigOWLIM*. http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V7/
11. Cardoso, Jorge, Martin Hepp, und Miltiadis Lytras. 2008. *The Semantic Web: Real-World Applications from Industry (Semantic Web and Beyond)*. New York: Springer
12. Cranford, Steve 2009. *Spinning a Data Web. In: Price Waterhouse Coopers (Ed.). Technology Forecast, Spring 2009*. http://www.pwc.com/us/en/technology-forecast/spring2009/index.jhtml. Zugegriffen: 24. Februar 2014
13. Kobilarov, Georgi et al. 2009. *Media meets Semantic Web 2009: How the BBC uses DBpedia and linked data to make connections. The Semantic Web: Research and Applications*, 723–737. Berlin Heidelberg: Springer
14. Lehmann, Jens et al, 2012. DBpedia: a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal*, 5, 1–29
15. McKinsey Global Institute 2011. *Big data: The next frontier for innovation, competition and productivity. Research Report*. http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation. Zugegriffen: 24. Februar 2014
16. Mitchell, Ian, und Mark Wilson. 2012. *Linked Data. Connecting and exploiting big data. Fujitsu White Paper, March 2012*. http://www.fujitsu.com/uk/Images/Linked-data-connecting-and-exploiting-big-data-%28v1.0%29.pdf. Zugegriffen: 24. Februar 2014
17. Pellegrini, Tassilo 2014. Linked Data Licensing – Datenlizenzierung unter netzökonomischen Bedingungen. In *Transparenz. Tagungsband des 17. Internationalen Rechtsinformatik Symposium IRIS 2014*, Hrsg. Erich Schweighofer, Franz Kummer, Walter Hötzendorfer, 159–168. Wien: Verlag der Österreichischen Computergesellschaft
18. Sporny, Manu 2013. *JSON-LD is the Bee's Knees*. http://manu.sporny.org/2013/json-ld-is-the-bees-knees/. Zugegriffen: 24. Februar 2014
19. Wood, David (Hrsg.). 2010. *Linking Enterprise Data*. New York: Springer
20. Wood, David (Hrsg.). 2011. *Linking Government Data*. New York: Springer

Linked Data Technologien – Ein Überblick

2

Harald Sack

Zusammenfassung

Der vorliegende Beitrag bietet einen grundlegenden Überblick über das Thema Linked Data und führt in die dazugehörigen Basistechnologien ein. Ausgehend von URIs zur eindeutigen Identifikation von Ressourcen wird das Resource Description Framework (RDF) zur einfachen Modellierung von Fakten detailliert eingeführt. Linked Data lebt von der Verknüpfung der Fakten untereinander sowie mit zugrundeliegenden Wissensrepräsentationen (Ontologien). In diesem Zusammenhang werden auch RDF(S) und OWL als formale Ontologiebeschreibungssprachen kurz vorgestellt, um Möglichkeiten und Grenzen des Ansatzes aufzuzeigen. Weiterführend werden Möglichkeiten zur Nutzung von Linked Data in eigenen Anwendungen vorgestellt sowie auf die Publikation eigener Datensätze als Linked Data eingegangen.

2.1 Was ist Linked Data?

„*Was heißt und zu welchem Ende studiert man*“ *Linked Data*? Diese Frage steht, dem klassischen Beispiel Friedrich Schillers folgend, am Anfang dieser kurzen Darstellung [1]. Bevor aber darauf eine Antwort gegeben werden kann, soll auf die aktuellen Problemstellungen im World Wide Web (WWW) bei der großmaßstäblichen Verarbeitung traditioneller Daten eingegangen werden. Das WWW ist heute nach gut 25 Jahren seines Bestehens zu einer globalen Wissensbasis herangewachsen, die schon seit geraumer Zeit auch unser tägliches Leben mitbestimmt. Angefangen mit der E-Mail an unsere Freunde und Bekannten, über Online-Käufe, Online-Banking, bis hin zur Kontaktpflege über soziale Netzwerke

H. Sack ✉
Hasso-Plattner-Institut für Softwaresystemtechnik GmbH, Universität Potsdam, 14482 Potsdam, Deutschland
e-mail: harald.sack@hpi.uni-potsdam.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_2

Abb. 2.1 Zugriff auf Daten im WWW erfolgt gekapselt über den Web-Server. Der Endbenutzer kann über das HTTP Protokoll auf den Web-Server zugreifen, der selbst über ein weiteres Protokoll, z. B. JDBC (Java Database Connectivity) auf eine Datenbank zugreift

und der multimedialen Unterhaltung: Alles findet heute auch bzw. zunehmend im WWW statt. Ursprünglich als ein dokumentenzentriertes, dezentrales Netzwerk geplant, stehen heute ebenfalls multimediale Dokumente, Anwendungen und große Mengen von strukturierten Daten im WWW zur Verfügung. Allerdings gestaltet sich der Zugriff speziell auf die darin enthaltenen Daten meist als schwierig. Einerseits können Web-Dokumente auf unterschiedliche Weise kodiert sein, z. B. via HTML (Hypertext Markup Language), PDF (Portable Document Format), oder auch als proprietäres Word-Dokument, Excel Spreadsheet oder als graphische Präsentation. Enthalten diese Dokumente Daten, liegen sie entweder in unstrukturierter Form vor, eingebettet in den umgebenden Text, oder formatiert als Tabelle. Daher müssen die in frei zugänglichen WWW-Dokumenten enthaltenen Daten oft aufwändig aus den zugrundeliegenden Dokumenten extrahiert werden, wobei der Extraktionsprozess fehlerbehaftet ist und bei Änderungen dieser Dokumente stets angepasst werden muss. Andererseits basieren Webportale und Unternehmenswebseiten oft auf Content Management Systemen (CMS), denen eine Datenbank zugrunde liegt, deren Inhalte via Templates zu Webseiten umgeformt und präsentiert werden.

2.1.1 Von Dateninseln und proprietären Schnittstellen

Ein direkter Zugriff auf die strukturierten Daten in den zugrundeliegenden Datenbanksystemen ist meist nicht möglich, da das CMS einen Web-Server als Schnittstelle zur Interaktion mit dem Client vorsieht (vgl. Abb. 2.1). Aus diesem Grund bieten zahlreiche Anbieter großer Datenbestände vorgefertigte Schnittstellen (Application Programming Interfaces, API), über die kontrolliert auf die strukturierten Daten zugegriffen werden kann. Allerdings unterliegen diese APIs meist generellen Restriktionen (bzgl. beschränkt verfügbarer Datenmengen oder notwendiger Autorisierungen). Jeder Anbieter eines APIs

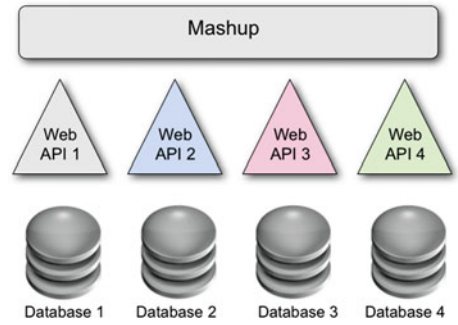


Abb. 2.2 Mashup Anwendungen verwenden mehrere unterschiedliche Web-APIs, um heterogene Daten aus dem WWW in neuem Kontext zusammenzuführen und daraus neue Informationsangebote zu generieren

trifft diese Reglementierungen inklusive der Art und Weise, wie ein Client die Schnittstelle bedienen muss gemäß seinen eigenen, speziellen Anforderungen. Es existiert also keine gemeinsame bzw. gleichartige Schnittstelle, die für alle Datenanbieter verbindlich wäre. So kann z. B. ein Benutzer der Amazon-API damit nicht auf die Daten des Google-Suchindexes zugreifen und umgekehrt. Für jede Schnittstelle muss ein spezielles API bedient werden wie Abb. 2.2 veranschaulicht. Die Interaktion des APIs mit dem Web-Server erfolgt über Standardprotokolle, wie z. B. HTTP (Hypertext Transfer Protocol) oder SOAP (System Open Access Protocol). Die Software, die auf Anwenderseite dieses API verwendet, unterliegt ebenfalls dem Lebenszyklus der Interfacereglementierungen seitens des Datenanbieters, d. h. ändert der Datenanbieter die Spielregeln bzw. die im Interface verwendeten Datenschemata, muss die Anwendungssoftware ebenfalls angepasst werden, was einen hohen Aufwand mit sich bringt. Diese hohen Aufwände verhindern bislang die Vernetzung der (strukturierten) Daten im WWW. Auf diese Weise sind in den vergangenen Jahrzehnten zahlreiche voneinander abgeschottete Datensilos entstanden, eine aggregierte gemeinsame Nutzung verfügbarer Datenangebote findet nicht statt. Der via Linked Data beschrittene Lösungsweg verspricht eine einfache und praktikable Lösung zur Nutzung der im WWW vorhandenen Daten, um daraus neue aggregierte Informationsangebote zu entwickeln [2].

Die Grundvoraussetzung zur Realisierung von Linked Data besteht in zwei einfachen Prämissen:

1. **Strukturierte Daten ermöglichen differenzierte und anspruchsvolle Anwendungen**
   Ausschlaggebend für eine einfache Wiederverwendung von Daten ist der Grad ihrer Strukturiertheit. Klar definierte Datenstrukturen ermöglichen die Entwicklung einfacher Werkzeuge, die diese Daten korrekt lesen und zuverlässig weiterverarbeiten können. Das Web-Dokumenten zugrunde liegende HTML als semistrukturierte Markupsprache muss zuerst aufwändig analysiert werden um strukturierte Daten aus HTML-Dokumenten zu gewinnen. Via microformats (µformats)[1] können strukturierte Daten

[1] Siehe http://microformats.org/, aufgerufen am 02.04.2014.

in HTML-Dokumente integriert werden, die auf einfache Weise zuverlässig wieder ausgelesen werden können. Der Nachteil der Verwendung von microformats liegt in der geringen Anzahl verfügbarer Datenschemata und zugehöriger Attribute zur Beschreibung der eigenen Daten. Oft ist es damit nicht möglich, Beziehungen zwischen einzelnen Entitäten differenziert darzustellen. So ist es zwar möglich, eine Person als Teilnehmer einer Veranstaltung zu definieren, ob diese aber lediglich als passiver Teilnehmer oder als Vortragender teilnimmt, kann nicht dargestellt werden.

2. **Verteilte Daten können über Hyperlinks miteinander verbunden werden**
   Damit folgen Daten dem Grundprinzip des WWWs, d. h. sie können dezentral verteilt vorliegen und ihr Zusammenhang kann auf einfache Weise via Hyperlinks dargestellt werden. Die Mehrzahl der verfügbaren Web-APIs bezieht sich auf Daten, die lediglich lokal in einer vorbestimmten Website vorliegen. Eine Referenzierung externer Daten ist über ein Web-API in der Regel nicht möglich.

Die Lösung des zu Beginn dargestellten Problems besteht darin, die heute bestehenden Datensilos aufzubrechen und die verfügbaren Daten in strukturierter Form derart anzubieten, dass diese von anderen Anwendungen problemlos abgerufen, genutzt und weiterverarbeitet werden können, welche die daraus neu entstehenden Daten ebenfalls wieder in gleicher Form zur Verfügung stellen.

2.1.2 Semantic Web Technologien

Die mit Linked Data beschrittene Lösung macht sich existierende Technologien zu Nutze, die vom W3C[2] als Semantic Web Technologien standardisiert wurden. Die Idee des Semantic Webs basiert auf den Überlegungen des Entwicklers und Initiators des World Wide Webs, Sir Tim Berners-Lee, der bereits in seiner ursprünglichen Entwurfsskizze für das WWW den vorhandenen Verknüpfungen und Links eine differenzierte Bedeutung zugewiesen hatte. Allerdings erwies sich die Umsetzung seines Konzepts als wesentlich einfacher, wenn man die Hyperlinks zwischen den Dokumenten des WWWs undifferenziert, d. h. ohne Beachtung deren Bedeutung verwendet [3]. Das WWW wurde in erster Linie zum Gebrauch für den Menschen entwickelt. Dies spiegelt sich auch in der den Web-Dokumenten zugrundeliegenden HTML-Kodierung. HTML enthält lediglich Strukturierungsinformationen, die sich auf die Darstellung von Information in natürlicher Sprache bzw. multimedialer Information beziehen, und ist nicht in der Lage, auch die Bedeutung der dargestellten Information festzulegen. Natürliche Sprache ist durch einen hohen Grad an Mehrdeutigkeit und Ungenauigkeit gekennzeichnet. Dies setzt sich in Web-Dokumenten fort. Eine einfache Suche im WWW nach einem mehrdeutigen Suchbegriff wie z. B. „*Jaguar*“, liefert Web-Dokumente als Suchergebnis zurück, die sich

[2] World Wide Web Consortium, http://www.w3.org/, aufgerufen am 02.04.2014.

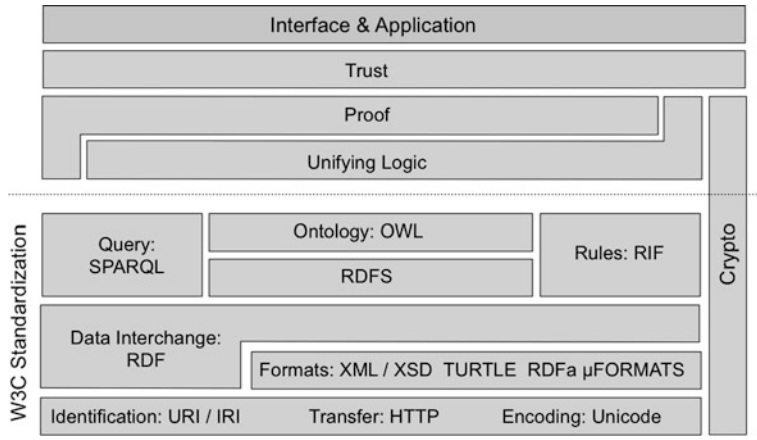


Abb. 2.3 Semantic Web Technologie-Stack des W3C[4]

sowohl auf die Großkatze, den Automobilhersteller oder aber auch auf das (historische) gleichnamige Apple Betriebssystem oder die (historische) Spielekonsole beziehen[3]. Andererseits liefert die Web-Suche nach „*Jaguar*“ auch nicht alle Dokumente zurück, die sich inhaltlich z. B. auf die Großkatze gleichen Namens beziehen, da, um in der Suchmaschine gefunden zu werden, der Suchterm „*Jaguar*“ im ermittelten Dokument enthalten sein muss. Wird im Dokument lediglich ein Synonym verwendet, wie z. B. die lateinische Gattungsbezeichnung „*Panthera Onca*“, dann wird das Dokument nicht gefunden, da der originale Suchbegriff nicht enthalten ist, obwohl die Bedeutung von Synonym und originalem Begriff identisch ist. WWW-Suchmaschinen suchen also nach Termäquivalenzen und nicht nach inhaltlichen Bedeutungsäquivalenzen.

Die Idee hinter dem Semantic Web liegt darin, die Bedeutung von sprachlichen Begriffen und anderen bedeutungstragenden Entitäten explizit in einer maschinenlesbaren und vom Computer korrekt interpretierbaren Form anzugeben. Diese Form von Wissensrepräsentation wird in der Informatik als Ontologie bezeichnet. Ontologien sind explizite, formale Spezifikationen von gemeinsamen Konzeptualisierungen. Unter einer Konzeptualisierung versteht man ein formales Modell, bestehend aus Klassen, Relationen und Instanzen. Dieses Modell muss explizit sein, d. h. kein Bestandteil des Modells darf unspezifiziert bleiben. Das Modell muss zudem formal sein, d. h. es muss in einer maschinenlesbaren Form, die korrekt von einem Computer interpretiert werden kann, abgelegt werden, wie z. B. mit Hilfe einer mathematischen Logik. Um diese Wissensrepräsentation gemeinsam verwenden zu können, müssen sich alle Beteiligten auch auf deren gleiche Bedeutung geeinigt haben [4]. Zu diesem Zweck wurden über das W3C aufeinander aufbauende semantische Technologien standardisiert, mit denen sich Wissen im

[3] Siehe http://de.wikipedia.org/wiki/jaguar_(Begriffskl%C3%A4rung), aufgerufen am 02.04.2014.

WWW formal repräsentieren lässt. Abbildung 2.3 zeigt den vorgeschlagenen Semantic Web Technologie-Stapel, der das traditionelle World Wide Web ergänzt, und gibt den aktuellen Stand der Standardisierung an.

Basierend auf den existierenden Standard-Webtechnologien URI (Uniform Resource Identifier) und IRI (Internationalized Resource Identifier) lassen sich nicht nur Dokumente, sondern auch Objekte aus der realen Welt identifizieren und adressieren, über die mit Hilfe der darauf aufbauenden Wissensrepräsentationen Aussagen getroffen werden. Der Zugriff auf semantische Daten erfolgt über das im WWW standardisierte HTTP Protokoll (Hypertext Transfer Protocol). Dabei macht man sich den Mechanismus der sogenannten Content-Negotiation zu Nutze, um zwischen einem menschlichen Benutzer zu unterscheiden, der lediglich an lesbaren Informationsressourcen zu einem semantischen Objekt interessiert ist, oder ob eine Maschinenanfrage direkt auch mit maschinenlesbarer Information beantwortet werden soll. Zur allgemeinen maschinenlesbaren Kodierung von semantischen Daten wurde ursprünglich XML (Extended Markup Language) und XML Schema Definition Language (XSD) eingesetzt, die heute von weiteren Technologien, wie z. B. Turtle, RDFa und µFormats (microformats) ergänzt werden. Allgemein werden Fakten im Semantic Web mit Hilfe des Resource Description Frameworks (RDF) formuliert und ausgetauscht, das über die im Technologiestapel darunterliegenden Kodierungsformate (XML, Turtle, RDFa, etc.) serialisiert werden kann. Darauf aufbauend können via RDFS (RDF Schema, RDF Vocabulary Description Language) einfache Datenmodelle konstruiert werden, die über die Web Ontology Language (OWL) um zusätzliche Semantik, wie z. B. logische Einschränkungen (Constraints) ergänzt werden können. OWL basiert auf einer speziellen mathematischen Beschreibungslogik (SHROIQ(D)) und stellt ein ausdrucksmächtiges Instrument zur Modellierung von Wissensrepräsentationen (Ontologien) dar [34]. Zusätzlich bietet die Abfragesprache SPARQL die Möglichkeit, RDF-kodierte semantische Daten strukturiert abzufragen, wobei bereits (beschränkter) Gebrauch der Möglichkeit logischer Schlussfolgerungen gemacht werden kann. Mit dem Rule Interchange Format (RIF) stellt das W3C einen Standard zum Austausch logischer Regeln bereit, die einerseits bzgl. ihrer semantischen Ausdruckskraft über die Möglichkeiten von OWL hinausgehen, aber andererseits eine Anschlussmöglichkeit bereits existierender regelbasierter Systeme (Expertensysteme) gewährleistet. Eine weiter vereinheitlichende Logik und darauf basierende wissensverarbeitende, logikgetriebene Systeme (wie z. B. Schlussfolgerungssysteme) sind bislang noch nicht Teil der durch das W3C standardisierten Technologien, ebenso wie das auf kryptografischen Technologien basierende Trust-Management sowie weitere Schnittstellen und Anwendungen. Linked Data nutzt einen Teil dieser Technologien, die im Folgenden detaillierter dargestellt werden.

2.2 Die Linked Data Prinzipien

Bereits 2006 griff Tim Berners-Lee seine Idee des Datenwebs, bzw. des „Web of Data“, auf und veröffentlichte auf der Basis der Entwicklung des Semantic Webs einen knappen, aus vier kurzen Punkten bestehenden kleinen Best Practice Leitfaden zur Veröffentlichung

von Linked Data im Web[5]. Ebenso wie das traditionelle Dokumentenweb besteht das Web of Data aus Dokumenten. Aber im Gegensatz zum traditionellen Dokumentenweb, in dem Hyperlinks stets nur auf andere Dokumente verweisen, können im Web of Data Hyperlinks als beliebige Arten von Beziehungen zwischen Dingen (Entitäten) aufgefasst werden, die mit Hilfe von RDF beschrieben werden. URIs zur Adressierung von Dokumenten im Dokumentenweb werden zu allgemeinen Adressangaben, die jeden beliebigen Gegenstand und jedes abstrakte Konzept identifizieren können. Sowohl für das traditionelle Dokumentenweb als auch für das Web of Data gilt das selbe Grundprinzip: Mit dem Grad der Vernetzung (Verlinkung) steigt auch der Nutzen für den Anwender. Der Wert der Daten steigt, wenn diese mit anderen Datenquellen verbunden werden.

Folgende vier Regeln werden auch als Linked Data Prinzipien bezeichnet [2]:

1. Verwende URIs (Uniform Resource Identifiers), um Dinge, Gegenstände und Konzepte zu identifizieren und zu adressieren.
2. Verwende URIs, die via HTTP (Hypertext Transfer Protocol) aufgelöst (dereferenziert) werden, damit sie im Web nachgeschlagen werden können.
3. Wird ein URI im Web abgerufen, dann halte nützliche Informationen dazu bereit und verwende die W3C Standards RDF(S) und SPARQL.
4. Verknüpfe deine Daten mit anderen URIs, damit Benutzer noch mehr im Web of Data entdecken können.

In den folgenden Abschnitten wird genauer auf diese 4 Linked Data Prinzipien eingegangen.

2.2.1 Verwende URIs

Uniform Resource Identifier (URIs) sind nach RFC 3986 standardisiert und bilden ein globales Adressierungsschema, das sogenannte URLs (Uniform Resource Locator) und URNs (Uniform Resource Names) zusammenfasst [6]. URLs sind aus dem traditionellen Web als Dokumentenlinks wohlbekannt und mitverantwortlich für die große Popularität, die das Web heute erlangt hat. Während URLs nur die aktuelle Adresse eines Web-Dokuments angeben, können mit URNs Web-Dokumente eindeutig und dauerhaft benannt werden. Ändert ein Dokument den Ort, an dem es verfügbar war, ändert sich auch dessen URL. URNs dagegen sind dauerhafte Namen, die unabhängig vom Ort bzw. der Adressangabe des Dokuments verwaltet werden können.

Während im traditionellen Web lediglich Dokumente via Hyperlinks miteinander verknüpft werden konnten, erlaubt das Web of Data die Verknüpfung unterschiedlichster Daten, die selbst wieder für Objekte der realen Welt stehen können. Generell muss man hier die Unterscheidung treffen zwischen einer Informationsressource, die Informationen

[5] Siehe http://www.w3.org/designissues/linkeddata.html, aufgerufen am 02.04.2014.

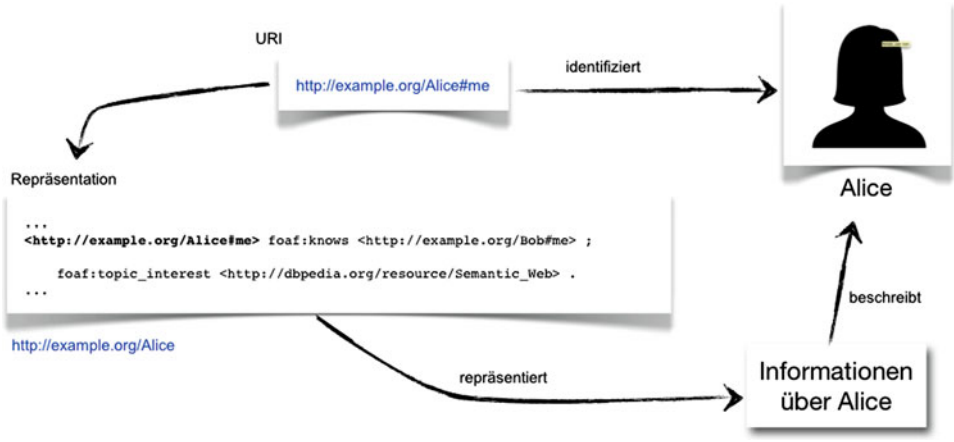


Abb. 2.4 Beziehung zwischen Objekt der realen Welt, seiner Repräsentation im Web of Data und den URIs, die das Objekt identifizieren/referenzieren bzw. Informationen über das Objekt repräsentieren

(Metadaten[6]) über ein Objekt der realen Welt liefert, und dem Objekt selbst, das im Web über diese Informationsressource repräsentiert werden soll. Ein einfaches Beispiel (vgl. Abb. 2.4): Auf einem Web-Server liegt ein Dokument bereit, das Informationen über Alice beinhaltet. Das Dokument soll unter dem URI http://example.org/person verfügbar sein. Das Dokument ist natürlich nicht Alice, sondern stellt Informationen über Alice bereit. Eine Möglichkeit, die Information über Alice und einen Repräsentaten für Alice im Web zu trennen, kann z. B. über die Verwendung eines Fragment-Identifiers erfolgen, der auf einen Teil des Dokuments http://example.org/person verweist, der für Alice im Web steht. Fragment-Identifier werden gemäß dem vorgegebenen URI-Schema mit Hilfe eines an den URI angehängten „#"-Zeichens kodiert, sodass der URI für den Repräsentanten von Alice im Web http://example.org/person#Alice lauten könnte. Werden jetzt Aussagen über Alice als Linked Data veröffentlicht, kann Alice unter Zuhilfenahme des festgelegten URIs http://example.org/person#Alice identifiziert und referenziert werden [7].

2.2.2 Verwende URIs mit HTTP

Um auf die Daten im Web of Data zugreifen zu können, wird neben der Adressierung via URIs ein Zugriffsprotokoll benötigt, über das die Daten ausgetauscht werden. Das traditionelle Web verwendet als einfachen Transportmechanismus das HTTP Protokoll,

[6] Als Metadaten werden in diesem Zusammenhang strukturierte, kodierte Daten bezeichnet, die Charakteristika informationstragender Objekte beschreiben, damit diese identifiziert, recherchiert, beurteilt oder verwaltet werden können [5].

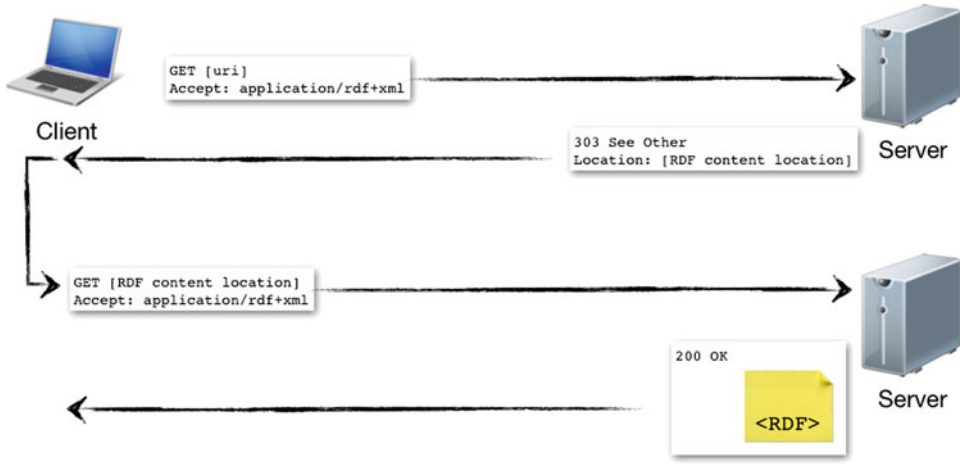


Abb. 2.5 HTTP Content Negotiation via 303 Redirect

das daher ebenfalls im Web of Data zum Einsatz kommt. Einer der dadurch gewonnenen Vorteile besteht darin, prinzipiell eine Kompatibilität zu allen bestehenden Webanwendungen zu gewährleisten. HTTP erlaubt in der Kommunikation zwischen Client und Server eine sogenannte Content Negotiation, d. h. der Client – im traditionellen Web ist dies der Web-Browser – übergibt in der HTTP-Anfrage an einen Web-Server einen Satz von Parametern, mit denen die Form des angefragten Inhalts näher bestimmt wird. Üblicherweise sind dies z. B. die Angabe der vom Benutzer verwendeten Sprache im Web-Browser, damit die jeweils entsprechenden Versionen dieser Dokumente vom Web-Server korrekt ausgeliefert werden können, falls diese in der angegebenen Sprache vorliegen. Im Zusammenhang mit Linked Data macht man sich HTTP Content Negotiation zu Nutze, indem sich der anfragende Client z. B. als menschlicher Benutzer bzw. als anfragendes Programm identifizieren kann, um eine bedarfsgerechte Auslieferung der angefragten Daten zu ermöglichen. Dies kann z. B. im Falle des menschlichen Benutzers, ein HTML-Dokument sein, das die Daten in für den Menschen einfach lesbarer Form wiedergibt und beschreibt. Im Fall des anfragenden Programms werden die Daten kodiert als RDF-Daten ausgeliefert, die direkt weiterverarbeitet werden können. Gesteuert wird die Content-Negotiation über das in der HTTP-Kommunikation vom Client übergebene Feld des „`Accept-Headers`“ in der HTTP-Anfrage (HTTP Request). Beinhaltet das Feld „`Accept`“ den Wert „`text/html`“, wird die für den Menschen bestimmte Variante ausgeliefert, während bei einem Feldwert „`application/rdf+xml`“ die maschinenlesbare Variante als RDF/XML-Datei vom Web-Server ausgeliefert wird. Der detaillierte Ablauf der Content-Negotiation ist in Abb. 2.5 dargestellt und kann folgendermaßen erfolgen:

1. Der Client setzt eine `HTTP GET` Anfrage an eine URI ab, die ein Konzept aus der realen Wert referenziert. Handelt es sich beim Client um eine Linked Data Anwendung, die das Ergebnis in RDF/XML kodiert bevorzugt, sendet er `Accept: application/rdf+xml`. Ein Web-Browser dagegen würde die Anfrage mit `Accept: text/html` versenden.
2. Der Web-Server erkennt die URI als eine URI, die ein Konzept aus der realen Welt referenziert. Da der Web-Server nicht die angefragte Ressource direkt ausliefern kann, sendet er den HTTP Statuscode `303 See Other` in seiner Antwort an den Client und übermittelt dem Client gleichzeitig die URI eines Web-Dokuments, welches das angefragte Konzept in der gewünschten Repräsentationsform beschreibt.
3. Der Client führt jetzt einen `HTTP GET` Request an die vom Web-Server übermittelte, weitergeleitete URI aus.
4. Der so angefragte Web-Server übermittelt via HTTP den `Statuscode 200 OK` und sendet dem Client das angefragte Dokument, das die originale Ressource beschreibt, im angefragten Format zurück.

2.2.3 Verwende W3C-Standards (RDF und SPARQL)

Um einer möglichst großen Zahl von unterschiedlichen Anwendungen die Nutzung von Linked Data Ressourcen zu ermöglichen, muss deren Inhalt vorgegebenen und allgemein akzeptierten Formatstandards entsprechen. Der für Linked Data Ressourcen vorgesehene Standard ist das Resource Description Format (RDF), das ein sehr einfaches und universelles Datenmodell vorgibt, um strukturierte Daten über das Web zu publizieren [8]. Das Datenmodell von RDF ist denkbar einfach. Jede mögliche Aussage wird darin auf ein einfaches Tripel-Schema heruntergebrochen, das ganz ähnlich aufgebaut ist wie ein einfacher Satz in natürlicher Sprache, i. e. aus *Subject*, *Property* und *Object*.

Ein einfaches Beispiel: Die Aussage

> Document.html hat den Autor Andreas.

lässt sich übersetzen in das Tripel

Document.html	*hatAutor*	*Andreas.*
(Subject)	(Property)	(Object)

Die einzelnen Bestandteile des Tripels, also Subject, Property und Object müssen als URI dargestellt werden, also z. B.

```
<http://example.org/Document.html>
<http://example.org/vocabular#hatAutor>
<http://example.org/Andreas#me>.
```

Dies ist bereits die vollständige Serialisierung eines RDF-Tripels in der einfachen Terse RDF Triple Language (Turtle), die vorgibt, dass URIs stets in spitzen Klammern geschrieben werden müssen und ein Tripel immer mit einem Punkt abgeschlossen wird. Man

unterscheidet in RDF Aussagen, deren Object selbst ein URI ist – also Aussagen, die den im Subject beschriebenen Gegenstand mit einem anderen Gegenstand verknüpfen –, von Aussagen, deren Objekt durch ein Literal oder einen Datenwert dargestellt wird. Dies hängt davon ab, ob das verwendete Property ein sogenanntes Object Property ist (bedingt einen URI als Objekt) oder ein sogenanntes Datatype Property (bedingt ein Literal oder einen Datenwert als Object). Zur Vereinfachung der Schreibweise können in Turtle URIs über eine Präfixdefinition verkürzt werden.

```
@prefix ex: <http://example.org/>.
```

Auf diese Art wird die Schreibweise weiterer URIs, die alle dasselbe URI-Präfix verwenden, folgendermaßen verkürzt:

```
@prefix ex: <http://example.org/>.
ex:Document.html ex:vocabular#hatAutor ex:Andreas#me.
```

Angenommen, Name und Alter des Autors sollen in unseren Daten ergänzt werden, können weitere Datatype-Properties aus bereits existierenden Vokabularen zu den RDF-Tripeln hinzugenommen werden, z. B. `<http://xmlns.com/foaf/0.1/name>` und `<http://xmlns.com/foaf/0.1/age>`.

Für das neu verwendete Vokabular wird ein weiteres Präfix definiert:

```
@prefix ex: <http://example.org/>.
@prefix voc: <http://example.org/vocabular#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

ex:doc.html voc:hatAutor ex:Andreas#me.
ex:Andreas#me foaf:name "Andreas";
              foaf:age 32.
```

Ein Literal im Object wird in Hochkommata angegeben, Zahlenwerte können direkt angegeben werden bzw. werden mit einer zusätzlichen speziellen Datentypangabe ergänzt, die aus dem XML-Schema Namensraum[7] adaptiert wird.

```
ex:Andreas#me foaf:age 32.
```

ist äquivalent zu

```
ex:Andreas#me
  foaf:age
  32|http://www.w3.org/2001/XMLSchema#integer>.
```

[7] Siehe http://www.w3.org/2001/xmlschema, aufgerufen am 02.04.2014.

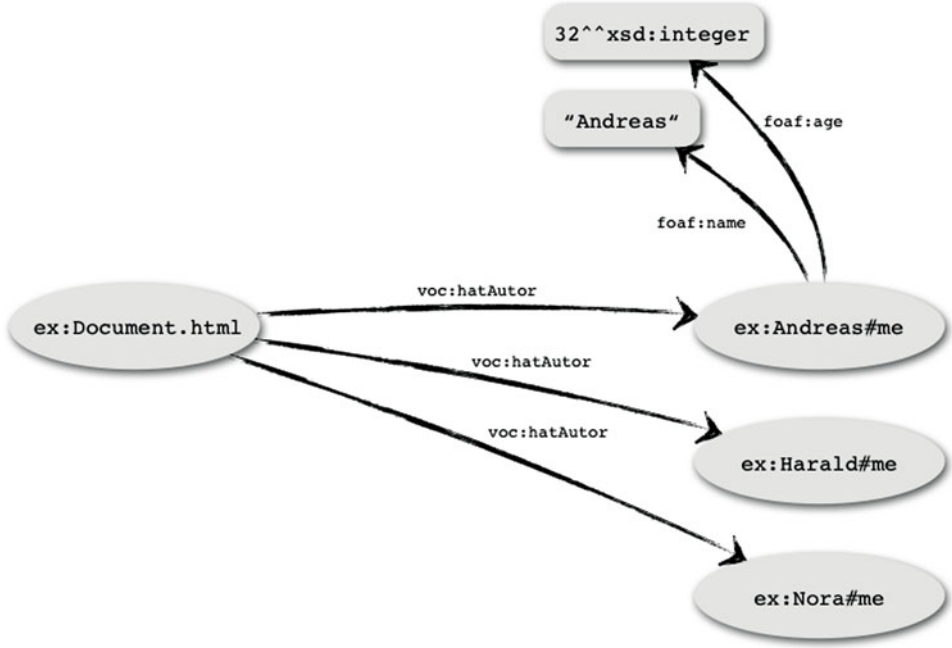


Abb. 2.6 RDF Graphrepräsentation des Beispiel: Die URIs wurden zur besseren Übersichtlichkeit in Präfixnotation angegeben

Beziehen sich Aussagen (RDF-Statements) auf ein und dasselbe Subject, wie im o. a. Beispiel auf den Autor `ex:Andreas#me`, dann können Tripel unter Verwendung eines Semikolons abgekürzt werden und das Subject muss nur einmal genannt werden. Weiter verkürzen lässt sich die Schreibweise, wenn einer Kombination aus gleichem Subject und Property verschiedene Werte als Object zugeordnet werden sollen, also wenn ein Dokument z. B. mehrere Autoren besitzt. Dann können die verschiedenen Object-Werte mit einem Komma voneinander getrennt angegeben werden.

```
@prefix ex: <http://example.org/>.
ex:Document.html ex:vocabular#hatAutor ex:Andreas#me,
                                       ex:Harald#me,
                                       ex:Nora#me.
```

Alternativ zur hier dargestellten Turtle Serialisierung lassen sich RDF-Tripel auch in einer XML-basierten Darstellung serialisieren oder auch als Graph darstellen. In der Graphdarstellung bilden Subject und Object die Knoten des Graphen, während Properties als Kanten dargestellt werden. Abbildung 2.6 zeigt das o. a. Beispiel in der RDF Graphrepräsentation.

Die RDF/XML-Serialisierung des Beispiels lautet:

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:voc="http://example.org/vocabular#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/">
 <rdf:Description
     rdf:about="http://example.org/Document.html">
  <voc:hatAutor>
   <rdf:Description
         rdf:about="http://example.org/Andreas#me">
     <foaf:name>Andreas</foaf:name>
       <foaf:age
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">
          25
       </foaf:age>
   </rdf:Description>
  </voc:hatAutor>
  <voc:hatAutor
        rdf:resource="http://example.org/Harald#me"/>
  <voc:hatAutor
        rdf:resource="http://example.org/Nora#me"/>
 </rdf:Description>
</rdf:RDF>
```

In der RDF/XML-Serialisierung lassen sich ebenfalls Präfixe zur übersichtlichen Schreibweise der URIs definieren. Jedoch können Präfixe entsprechend den XML Syntaxregeln nicht innerhalb von Hochkommata verwendet werden.

Um RDF direkt in HTML-Web-Dokumente einzubetten wurde RDFa (RDF in Attributes) entwickelt. Dabei bedient man sich der Option, RDF-Code direkt innerhalb verwendeter HTML-Tags mit Hilfe vorbestimmter HTML-Attribute festzulegen. Auf diese Weise wird RDF nicht wie in früheren Varianten als Kommentar in die HTML-Dokumente eingebracht, sondern direkt mit dem HTML Domain Object Model (DOM) verknüpft, sodass eine Zuordnung von HTML-Dokumentenabschnitten und zugehöriger RDF-Information auch feingranular ermöglicht wird. Eine detaillierte Einführung in die Benutzung von RDFa gibt der vom W3C herausgegebene RDFa Primer [8]. Die Verwendung von RDFa ist insbesondere dann von Vorteil, wenn die im vorangegangenen Abschnitt erläuterte Dereferenzierung via `HTTP 303 Redirect` nicht oder nur umständlich möglich ist, wie z. B. in den meisten Content Management Systemen. Dort können RDFa-Annotationen leicht in vorhandene HTML-Templates zur Erzeugung neuer HTML-Seiten eingebunden werden (vgl. auch Kap. 5).

Eine weitere Variante der RDF Serialisierung bietet RDF/JSON [9], mit der RDF-Graphen in der in der Web-Entwicklung bereits weit verbreiteten JSON (Java Script Ob-

ject Notation) dargestellt werden können. Auf diese Weise können Webentwickler RDF-Daten auf unkomplizierte Weise benutzen, ohne zu diesem Zweck aufwändige RDF-Parser oder weitere Software-Bibliotheken in ihre Anwendungen einbinden zu müssen. Die RDF/JSON Repräsentation des RDF-Tripel

```
<http://example.org/Document.html>
<http://example.org/vocabular#hatAutor>
<http://example.org/Harald#me>.
```

lautet:

```
{
  "http://example.org/Document.html": {
      "http://example.org/vocabular#hatAutor": [
         { "value": "http://example.org/Harald#me",
           "type": "uri"} ]
  }
}
```

Eine weitere Besonderheit von RDF liegt in der Verwendung sogenannter leerer Knoten (*Blank Nodes*). Diese können nicht von außerhalb eines RDF-Dokuments referenziert werden, besitzen aber eine strukturierende Aufgabe, wie z. B. in der Zuweisung von Mehrfachattributwerten. Darüber hinaus gestattet RDF auch die Zusammenfassung einzelner Ressourcen in geordneten bzw. ungeordneten Listen (sogenannte *RDF-Container* und *RDF-Collections*) sowie die Möglichkeit der Reifikation. Unter Reifikation versteht man in diesem Zusammenhang die Formulierung einer Aussage, d. h. eines RDF-Tripels, über das selbst wiederum eine Aussage getroffen wird. Auf diese Weise können einfache Provenienzinformationen (z. B. *Person X hat gesagt* ...) spezifiziert werden. Ein RDF-Statement (Tripel) wird auf diese Weise selbst als Ressource referenzierbar. Eine vollständige Aufzählung und detaillierte Beschreibung der RDF-Syntax kann dem vom W3C veröffentlichten RDF-Primer entnommen werden [10].

Mit Hilfe von RDF werden lediglich einfach strukturierte Aussagen definiert, die sich in Form eines Graphen wiedergeben lassen, aber selbst noch wenig Semantik tragen. Lediglich benannte Beziehungen zwischen Entitäten und Entitätenattributen lassen sich festlegen. Um eine Wissensrepräsentation im Sinne eines Datenmodells zu entwerfen, wird eine Modellierungssprache benötigt, die es erlaubt Klassen (als Zusammenfassung von einzelnen Entitäten mit gleichen oder ähnlichen Eigenschaften) zu definieren, Instanzen dieser Klassen festzulegen, sowie Properties als Relationen zwischen festgelegten Klassen abbilden zu können. Daneben sollten Klassenhierarchien und Property-Hierarchien konstruiert werden können. Dies wird mit Hilfe der RDF Schema Description Language, kurz RDF Schema oder RDFS ermöglicht [11]. RDF Schema basiert ebenfalls auf der RDF Tripeldarstellung. Um eine Klasse zu definieren, reicht es zunächst aus, dieser einen Namen zu geben:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://example.org>.
:Hund rdf:type rdfs:Class.
```

Im Beispiel wurde die Klasse „Hund“ mit Hilfe der `rdf:type` Typangabe als `rdfs:Class` definiert. Um jetzt festzulegen, was ein Hund ist, kann eine Klassenhierarchie via `rdfs:subclassOf` aufgebaut werden, die den Hund als Säugetier bzw. als Tier einordnet. Dabei gilt die Transitivitätsbeziehung, d. h. wenn ein Hund ein Säugetier ist, und wenn ein Säugetier ein Tier ist, dann ist ein Hund ebenfalls ein Tier. Die hinter der Definition von RDFS als logische Basis dienende formale Semantik lässt diese Art von Schlussfolgerungen automatisch zu.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://example.org>.
:Hund rdf:type rdfs:Class.
:Saeugetier rdf:type rdfs:Class.
:Tier rdf:type rdfs:Class.
:Hund rdfs:subclassOf :Saeugetier.
:Saeugetier rdfs:subclassOf Tier.
```

Um jetzt einen speziellen Hund zu benennen, kann eine Instanz der Klasse Hund definiert und benannt werden:

```
:Struppi rdf:type Hund.
```

Aus den vorangegangenen Klassendefinitionen kann jetzt geschlossen werden, dass Struppi ebenso ein Säugetier bzw. ein Tier ist. Das Schlüsselwort `rdf:type` kann in Turtle mit „a“ abgekürzt werden, das vereinfacht als „*ist ein*“ gelesen werden kann. In gleicher Weise lassen sich mit Hilfe von RDFS Properties, Propertyhierarchien und Einschränkungen von Properties bzgl. Grundmenge und Bildmenge (also über den Typ bzw. die Klassenzugehörigkeit des damit verbundenen Subjects und Objects) festlegen. Wir ergänzen unser Beispiel und legen fest, dass Andreas ein Haustier besitzt, einen Hund mit Namen Struppi.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://example.org>.
:Tier a rdfs:Class.
:Person a rdfs:Class.
:hatHaustier a rdfs:Property;
     rdfs:domain :Person;
     rdfs:range :Tier.
   :Andreas#me :hatHaustier :Struppi.
```

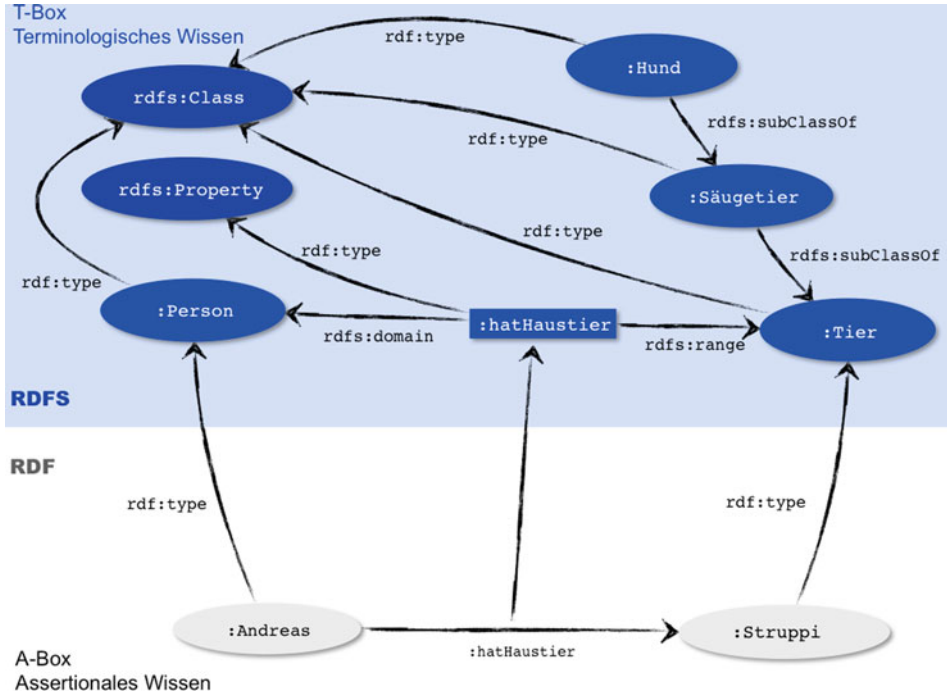


Abb. 2.7 RDF(S) Wissensbasis

Durch die Festlegung von Grund- und Bildbereich des Properties `:hatHaustier` lässt sich jetzt auch automatisch ableiten, dass Andreas eine Person und Struppi ein Tier sein muss. Eine vollständige Aufzählung und detaillierte Beschreibung der RDFS-Syntax findet sich in der vom W3C veröffentlichten RDFS-Referenz [11].

RDFS Definitionen legen die Struktur und Eigenschaften von Klassen und Properties fest, während via RDF einzelne Instanzen dieser Klassen und Klassenbeziehungen definiert werden können. RDFS Definitionen werden im Kontext der Wissensrepräsentation auch als terminologisches Wissen (T-Box) bezeichnet, während die über RDF definierten Instanzen und Instanzbeziehungen untereinander als assertionales Wissen (A-Box) bezeichnet werden, die den Zustand der damit modellierten „Welt“ beschreiben. Gemeinsam bilden T-Box und A-Box eine RDF(S) Wissensbasis (vgl. Abb. 2.7).

Über RDF(S) ist nur eine relativ eingeschränkte Wissensrepräsentation möglich. Zur Festlegung komplexerer logischer Strukturen, Einschränkungen und Abhängigkeiten des repräsentierten Wissens können darauf aufbauend die vom W3C standardisierte Web Ontology Language (OWL) [12] oder logische Regeln (Rule Interchange Format, RIF) [13] herangezogen werden. OWL ermöglicht die Festlegung komplexer Klassen (z. B. „*Ein Streichquartett ist eine Gruppe von genau vier Musikern, die alle (mindestens) ein Streichinstrument spielen*“). Darüber hinaus erlaubt OWL auch die Festlegung von Gleichheits-

und Ungleichheitsbeziehungen zwischen Instanzen, Klassen oder Properties. Auf diese Weise können über den Einsatz von Werkzeugen zum logischen Schlussfolgern (Reasoner) Widersprüche und Inkonsistenzen in Wissensbasen entdeckt werden sowie neues (implizit verborgenes) Wissen abgeleitet werden. Das Rule Interchange Format erlaubt zusätzlich die Festlegung logischer Regeln (z. B. „Wenn Andreas ein Vegetarier ist und im Restaurant ein Gericht bestellt hat, das eine Zutat aus Fleisch enthält, dann wird Andreas sein bestelltes Gericht nicht mögen.“) als Axiome für die Wissensbasis und ermöglicht eine Kompatibilität zu bestehenden (regelbasierten) wissensverarbeitenden Systemen (z. B. Expertensystemen).

Ein weiterer W3C Standard, der den Zugriff und die gezielte Abfrage von RDF-basierten Wissensbasen erlaubt, ist die Abfragesprache SPARQL (SPARQL Protocol and RDF Query Language). Angelehnt an die populäre Datenbankabfragesprache SQL, über die ein einfacher Zugriff auf relationale Datenbanken erfolgt, kann auf RDF-Graphen via SPARQL gezielt zugegriffen und diese abgefragt werden. Eine SPARQL-Abfrage basiert auf der einfachen Festlegung eines sogenannten „Graph-Pattern“, d. h. einem Muster aus vorgegebenen RDF-Tripeln, die im während der Abfrage traversierten RDF-Graphen aus diesem herausgefiltert werden. Werden dabei alle Komponenten eines RDF-Tripels ausspezifiziert (d. h. Subject, Property und Object), dann wird genau dieses RDF-Tripel aus dem RDF-Graphen herausgefiltert.

```
:Andreas#me rdf:type :Person.
```

Um aber z. B. zu bestimmen, welche RDF-Subjects einem bestimmten Muster aus vorgegebenem Property und Objekt folgen, ersetzt man dieses im Graph-Pattern einfach durch eine Variable.

```
?person rdf:type :Person .
```

Das angegebene Beispiel ermittelt auf diese Weise alle Personen aus dem RDF-Graphen. Über eine SPARQL-Abfrage können zudem das gewünschte Ausgabeformat oder die jeweils abzufragenden RDF-Graphen explizit festgelegt werden.

```
SELECT ?person FROM <http://example.org>
WHERE {
  ?person rdf:type :Person .
}
```

Um die verkürzte Präfix-Schreibweise von Ressourcen zu verwenden, müssen in SPARQL ähnlich wie in RDF(S) zunächst entsprechende Präfix-Definitionen vorgenommen werden. Die o.a. SPARQL-Anfrage selektiert alle Ressourcen (Variable `?person`) aus dem RDF-Graphen `<http://example.org>`, die vom Typ Person sind und in einem RDF-Tripel als Subjekt vorkommen. Graph-Patterns können in SPARQL miteinander kombiniert werden, um komplexere Abfragen zu definieren. Ebenso erlaubt

SPARQL die Verwendung von Aggregationsfunktionen oder die Kombination mehrerer Anfragen auf unterschiedliche RDF(S)-Graphen. Zudem können via SPARQL auch neue RDF(S)-Tripel in den Graphen eingefügt bzw. wieder gelöscht werden. Eine vollständige Auflistung der Möglichkeiten von SPARQL findet sich in der vom W3C standardisierten SPARQL Referenz [14].

Gemäß den Linked Data Prinzipien sollen strukturierte Daten im Web of Data via RDF als universelles Datenmodell repräsentiert werden. Die so repräsentierten strukturierten Daten bilden einen RDF-Graphen, in dem Ressourcen über URIs dargestellt werden, die alle dereferenziert werden können sollten (vgl. Abschn. 2.2). Um eine einfache Wiederverwendbarkeit von Linked Data Ressourcen gewährleisten zu können, sollten RDF-Autoren folgende Einschränkungen beachten [2]:

1. RDF-Reifikation sollte vermieden werden, da via Reifikation repräsentierte Information relativ komplexe SPARQL-Abfragen erfordert.
2. Ebenso sollten RDF-Container und -Collections weitgehend vermieden werden, da sich der Zugriff via SPARQL als relativ aufwändig gestaltet.
3. Blank-Nodes sollten ebenfalls vermieden werden, da diese sich nicht extern referenzieren lassen. Dies erschwert die Integration verschiedener Datensätze, da für Blank-Nodes keine URI als gemeinsamer Schlüssel existiert.

2.2.4 Verknüpfe deine Daten mit anderen URIs

Das Grundprinzip hinter Linked Data besteht in der Vernetzung der Daten zu einem Web of Data. Daher sollten neue im Web of Data veröffentlichte Datensätze stets mit bereits bestehenden Datensätzen verknüpft werden, um deren Wiederverwendbarkeit durch Dritte zu ermöglichen. Bei inhaltlichen Zusammenhängen beschriebener RDF-Ressourcen eines Datensatzes sollten diese Zusammenhänge über URI-Verknüpfungen mit RDF-Ressourcen aus anderen Datensätzen explizit angegeben werden. Auf diese Weise wird der Datensatz mit neuen Informationen in Bezug gesetzt und es entsteht ein Mehrwert. Technisch gesehen versteht man unter einem solchen RDF-Link ein RDF-Tripel, dessen *Subject* im Namensraum des einen Datensatzes liegt, wobei *Property* oder *Object* in einem anderen Namensraum referenziert werden. Werden die URIs der externen RDF-Ressourcen dereferenziert, liefert ein entfernt liegender Server eine Beschreibung dieser Ressourcen zurück. Auf diese Weise kann eine Navigation entlang der Verknüpfungen im Web of Data erfolgen, die von Linked Data Anwendungen, wie z. B. Linked Data Browsern oder Suchmaschinen-Robots ausgenutzt werden.

Prinzipiell lassen sich drei verschiedene Varianten von RDF-Links unterscheiden [2]:

1. **Relationship Links** verweisen auf andere Ressourcen aus dem selben oder aus einem externen Datensatz, die im Bezug zu den originalen Ressourcen stehen. Dies kann sich z. B. auf Personen, Organisationen oder Orte beziehen, aber auch auf weiterführende Informationen, wie z. B. bibliografische Angaben.
2. **Identity Links** stellen einen Bezug zu anderen Datensätzen her, indem sie von einer Ressource auf die identische Ressource in einem anderen Datensatz verweisen. Z.B. wird Albert Einstein sowohl in der DBpedia als auch in der Wissensbasis Freebase[8] als Entität referenziert. Die Identität beider Entitäten wird in der DBpedia durch folgendes Tripel beschrieben:

   ```
   <http://dbpedia.org/resource/Albert_Einstein>
   <http://www.w3.org/2002/07/owl#sameAs>
   <http://www.freebase.com/m/0jcx>.
   ```

   Auf diese Weise können z. B. zusätzliche Informationen über die Ressource ermittelt werden oder auch unterschiedliche Standpunkte und Meinungen aus unterschiedlichen Datenquellen bezogen werden.
3. **Vocabulary Links** verweisen auf Definitionen und Erläuterungen von Begriffen und Termen, die in einem Datensatz benutzt werden, und dienen so der Dokumentation sowie dem besseren Verständnis des verwendeten Vokabulars.

Folgt man den genannten Prinzipien bei der Publikation eigener Linked Data Ressourcen, werden diese zu einem Teil des Web of Data. Die eigenen Daten können dann sowohl auf einfache Weise mit zusätzlichen Daten aus dem Web of Data in Bezug gebracht und ergänzt werden und können gleichzeitig auch von anderen genutzt werden, um einen Mehrwert daraus zu generieren. Linked Data bietet in diesem Sinne ein vereinheitlichendes einfaches Datenmodell, das sich in erster Linie auf RDF stützt. Im Gegensatz dazu verwenden die übrigen im Web verwendeten Publikationsverfahren für strukturierte Daten eine Vielzahl unterschiedlicher Datenmodelle, deren Integration nur sehr aufwändig vorgenommen werden kann. Darüber hinaus bietet Linked Data über den einheitlichen Zugriffsmechanismus des HTTP Protokolls einen bereits etablierten Standard, während Web-APIs nur über verschiedenartige, meist proprietäre Schnittstellen abgerufen werden können. Da URIs als globales Adressierungs- und Identifikationsverfahren für jegliche Entitäten im Web of Data verwendet werden, sind Linked Data Anwendungen selbst zur Laufzeit in der Lage, neue Daten und Informationen zu entdecken und direkt zu verwenden, während über Web-API abrufbare Daten zunächst stets isoliert bleiben. Werden zur

[8] Siehe http://www.freebase.com/, aufgerufen am 02.04.2014.

Beschreibung von Linked Data Ressourcen Vokabularien verwendet, deren formale Beschreibung in Form einer Ontologie hinterlegt wurde, werden diese selbsterklärend und können von Linked Data Anwendungen „verstanden[9]“ und entsprechend weiterverarbeitet werden.

2.3 Das Web of Data und seine Inhalte

Die Anwendung der im vorangegangenen Kapitel beschriebenen Linked Data Prinzipien führt zur Entstehung eines Netzwerks strukturierter Daten, auf das über das WWW zugegriffen werden kann, das Web of Data [15]. Das Web of Data bildet einen gigantischen Graphen aus Milliarden von RDF Statements, die aus zahlreichen unterschiedlichen Quellen stammen und Informationen aller Art bereithalten, wie z. B. enzyklopädische Daten, Informationen zu Personen, Orten und Organisationen, Medieninformationen zu Filmen, Fernsehen und Radioprogrammen, aber auch biomedizinische Informationen, Informationen zu Arzneimitteln und klinischen Tests, statistische Daten und Daten aus sozialen Netzwerken. Damit kann man das Web of Data als eine zusätzliche Abstraktionsschicht betrachten, die über dem klassischen Dokumenten-Web liegt und eng mit diesem verwoben ist. Abschnitt 2.3 gibt einen Überblick über die Entwicklung und die Inhalte des Web of Data, gefolgt von Abschn. 2.4, der kurz die Vokabulare und Ontologien darstellt, die zur Verknüpfung der Daten im Web of Data eingesetzt werden.

2.3.1 Die Linking Open Data Cloud

Die Ursprünge des Web of Data liegen in der Forschung um das Semantic Web begründet. Insbesondere das im Januar 2007 gestartete W3C Gemeinschaftsprojekt „Linking Open Data“ (LOD)[10] kann als Initialzündung des Web of Data angesehen werden. Das ursprüngliche Ziel des LOD Projekts lag in der Identifikation bereits bestehender, öffentlich verfügbarer Datensätze, deren Umsetzung nach RDF gemäß den Linked Data Prinzipien sowie ihrer anschließenden Offenlegung im WWW. Dabei unterlag das LOD Projekt als W3C Gemeinschaftsprojekt keinerlei Zugangsbeschränkungen. Jeder kann teilnehmen und seine Datensätze hier entsprechend den Linked Data Prinzipien publizieren. Diese Offenheit ist mitverantwortlich für den großen Erfolg und die Popularität von Linked Open Data als Keimzelle des Web of Data. Seither ist das Web of Data auf mehr als 60 Milliarden RDF Tripel aus mehr als 900 Datenquellen angewachsen[11]. Abbildung 2.8 zeigt einen Überblick über die sogenannte „Linking Open Data Cloud“, eine

[9] Unter „verstehen“ wird in diesem Zusammenhang „maschinenlesbar“ und „korrekt interpretierbar“ verstanden.
[10] Siehe http://www.w3.org/wiki/sweoig/taskforces/communityprojects/linkingopendata, aufgerufen am 02.04.2014.
[11] http://stats.lod2.eu/ (Stand: April 2014).

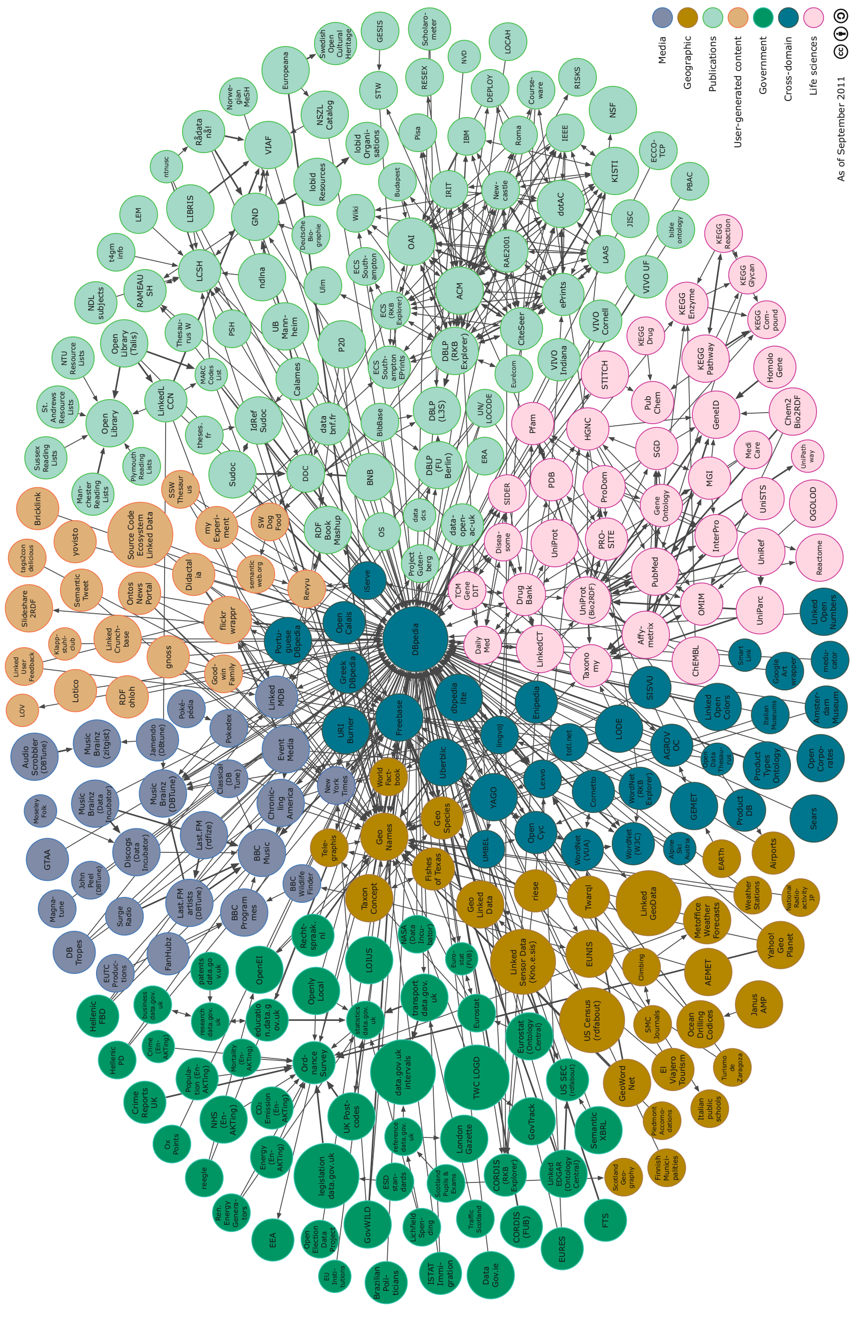


Abb. 2.8 Linking Open Data Cloud, letzter Stand: September 2011. Die unterschiedlichen Farben signalisieren unterschiedliche inhaltliche Themenkategorien

schematisch aggregierte Darstellung des Web of Data. Jeder Knoten des abgebildeten Netzwerks repräsentiert einen Datensatz, jede Kante deutet auf existierende RDF-Links zwischen zwei Datensätzen hin, wobei die Dicke der Kanten mit der Anzahl bestehender Links korrespondiert. Aktualisierte Varianten dieser Übersichtsgrafik sind verfügbar über http://lod-cloud.net und werden in regelmäßigen Abständen überarbeitet. Mehr Informationen über die Inhalte der einzelnen Datensätze der LOD-Cloud finden sich im Comprehensive Knowledge Archive Network (CKAN)[12], einem Katalog offener und frei zugänglicher Datensätze, an dem sich in gleicher Weise jeder beteiligen kann, um seine Daten in die LOD-Cloud aufzunehmen. Zur Beschreibung der Datensätze der LOD-Cloud werden aktuell mehr als 400 verschiedene Vokabularien verwendet[13]. Die unterschiedlichen Farben der Datensätze in Abb. 2.8 weisen auf eine inhaltliche Kategorisierung hin und repräsentieren LOD-Datensätze aus den Bereichen Medien, User Generated Data, bibliografische Daten, Biomedizin (Life Sciences), domänenübergreifende Daten (enzyklopädische Daten), Geografie und öffentliche Daten (Government Data). Ein zentraler Nukleus, um den herum sich zahlreiche Datensätze aller Kategorien verlinken, ist die *DBpedia*, deren Inhalte aus der populären Online-Enzyklopädie Wikipedia automatisiert gewonnen werden und die fachbereichsübergreifend als URI-Referenz Verwendung findet. Alle Wikipedia-Artikel korrespondieren zu DBpedia-Entitäten, die aus den Wikipedia-Artikeln gewonnene strukturierte Daten als RDF-Statements bereit halten. Der Wikipedia-Artikel zu Albert Einstein http://en.wikipedia.org/wiki/Albert_Einstein korrespondiert über ein identisches URI-Suffix zur DBpedia-Entität gleichen Namens http://dbpedia.org/resource/Albert_Einstein, wobei hier üblicherweise die englischsprachige Variante der Wikipedia herangezogen wird, die aber über sogenannte Interlanguage-Links mit den übrigen Sprachversionen der Wikipedia bzw. auch mit den lokalen Versionen der DBpedia verknüpft sind.

Eine weitere Online-Initiative in diesem Bereich ist *Wikidata*[14]. Dem Konzept der von den Benutzern bereitgestellten und gepflegten Daten bzw. Artikeln der Wikipedia folgend, setzt sich Wikidata zum Ziel, eine benutzergenerierte Plattform für strukturierte Daten im Web zu bieten. Eine Motivation für Wikidata besteht darin, strukturierte Daten für Wikipedia-Artikel zentral an einer Stelle im Web zu pflegen, die dann für alle Sprachversionen der Wikipedia als Ergänzung zur Verfügung stehen sollen. Des Weiteren bevorzugt Wikidata keine einheitlichen Meinungen und lässt unterschiedliche Angaben und Fakten nebeneinander unter Angabe des jeweiligen Urhebers bestehen, damit der Benutzer die Zuverlässigkeit bzw. Vertrauenswürdigkeit der Daten selbst beurteilen kann. Im Geografiebereich dient der frei verfügbare *Geonames* Datensatz[15] mit Informationen zu mehr als 8 Millionen Ortsangaben häufig als Referenz für weitere Datensätze, die geografische Angaben referenzieren. Im Medienbereich dienen die frei verfügbaren Datensätze von *BBC*

[12] Siehe http://datahub.io/group/lodcloud/, aufgerufen am 02.04.2014.
[13] Siehe http://lov.okfn.org/dataset/lov/, aufgerufen am 02.04.2014.
[14] Siehe https://www.wikidata.org/, aufgerufen am 02.04.2014.
[15] Siehe http://www.geonames.org/, aufgerufen am 02.04.2014.

Music[16] und *MusicBrainz*[17] häufig als Referenz für Musikmedien. Der Bereich der öffentlichen Daten wächst aktuell sehr schnell, da zahlreiche Staaten statistische Daten zur Wirtschaft, Infrastruktur, Bildung, etc. auch als Linked Data publizieren. Obwohl dieser Bereich aktuell den zahlenmäßig größten Teilbereich des Web of Data repräsentiert, ist dort jedoch die interne Verlinkung mit weiteren Datensätzen nur sehr gering ausgeprägt. Im biomedizinischen Bereich hat die Aufbereitung der strukturierten Daten für Wissensbasen bereits eine längere Tradition. Daher stammen zahlreiche Datensätze des Web of Data auch aus diesem Bereich. So bildet der Datensatz des *Bio2RDF*[18] Projekts ein zentrales Bindeglied für mehr als 30 Datensätze aus der Biomedizin. Aber auch kommerzielle Produktdatenbanken, wie z. B. *ProductDB*[19] oder *Sears*[20] in Verbindung mit der *GoodRelations*[21] Ontologie stellen heute einen Teil des Web of Data.

2.3.2 Was das Web of Data im Inneren zusammenhält

Zusammengehalten wird das Web of Data mit Hilfe gemeinsam genutzter Vokabulare und Ontologien. Das *Linked Open Vocabulary*[22] Portal hält eine stets aktualisierte Übersicht aller im Rahmen der Linking Open Data Cloud verwendeten Vokabularien und Ontologien bereit. Einige der zentralen, häufig verwendeten Vokabularien sind in Abb. 2.9 dargestellt. Während im LOD Cloud Diagramm (vgl. Abb. 2.8) Verknüpfungen der Datensätze untereinander auf der Instanzenebene angegeben sind, die meist einzelne Entitäten via einer Identitätsverknüpfung z. B. über `owl:sameAs` mit einem anderen Datensatz verknüpfen, steht in Abb. 2.9 die konzeptionelle Ebene (Klassenebene) im Vordergrund und die Vokabularien, die zur Verknüpfung der Klassen untereinander zur Anwendung gelangen. Eine der zentralen Nahtstellen ist die *umbel*[23] Ontologie, die auf der seit Jahrzehnten etablierten und manuell gepflegten *OpenCyc Upper Ontology*[24] basiert und speziell dafür geschaffen wurde, um die Interoperabilität heterogener Datensätze zu ermöglichen. Umbel stellt eine breit aufgestellte, zusammenhängende Ontologie aus mehr als 28.000 Konzepten zur Verfügung (über die Klasse `skos:Concept`), die untereinander über strukturierende Properties z. B. in Form von Klassenhierarchien verknüpft sind. Darüber hinaus beinhaltet umbel auch Identitätlinks zu DBpedia, Geonames und Verweise auf Wikipedia-Artikel. Während die vom W3C zur Ontologiebeschreibung festgelegten Standardsprachen RDFS und OWL nur wenige strukturierende Elemente (Properties) bereithalten – z. B. Defini-

[16] Siehe http://www.bbc.co.uk/music, aufgerufen am 02.04.2014.
[17] Siehe http://musicbrainz.org/doc/database, aufgerufen am 02.04.2014.
[18] Siehe http://bio2rdf.org/, aufgerufen am 02.04.2014.
[19] Siehe http://productdb.org/, aufgerufen am 02.04.2014.
[20] Siehe http://www.sears.com/, aufgerufen am 02.04.2014.
[21] Siehe http://www.heppnetz.de/projects/goodrelations/, aufgerufen am 02.04.2014.
[22] Siehe http://lov.okfn.org/dataset/lov/, aufgerufen am 02.04.2014.
[23] Siehe http://www.umbel.org/, aufgerufen am 02.04.2014.
[24] Siehe http://www.cyc.com/platform/opencyc, aufgerufen am 02.04.2014.

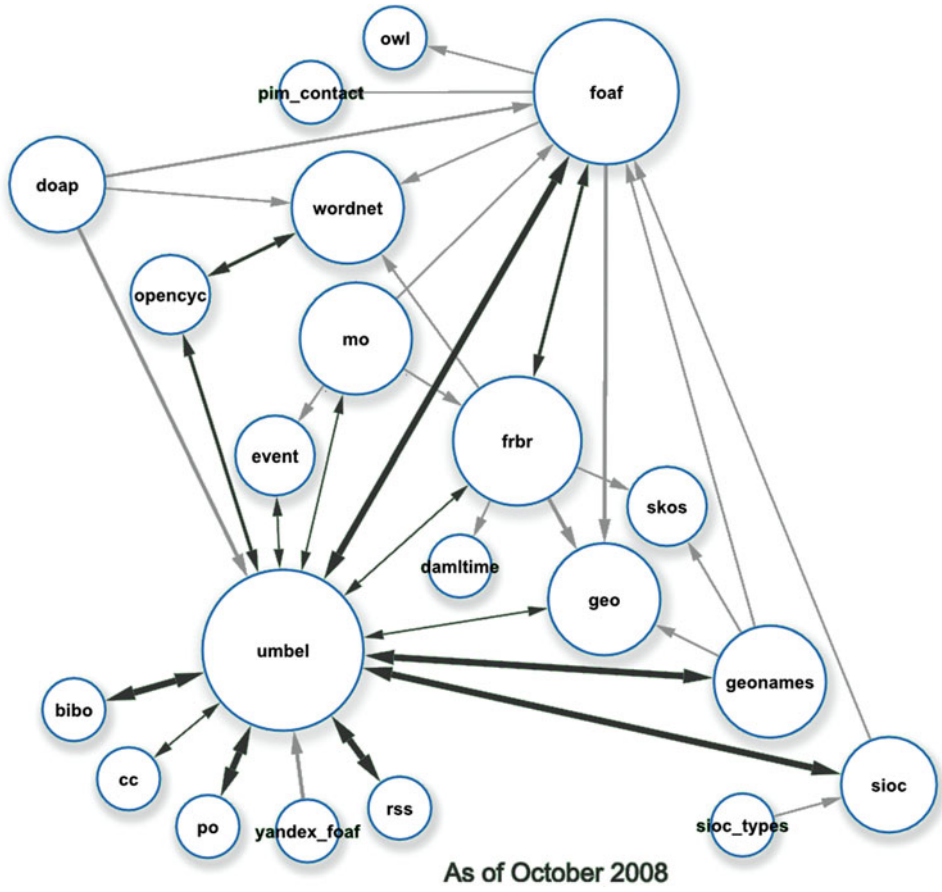


Abb. 2.9 Häufig verwendete Vokabularien und Ontologien des Web of Data

tion von Klassenäquivalenz via `owl:equivalentClass` oder Klassenhierarchien via `rdfs:subClassOf` –, erlauben Thesauri einen detaillierteren Grad der Strukturierung. Wordnet[25] ist ein als Thesaurus strukturiertes online Wörterbuch, das strukturelle Beziehungen, Synonyme, Partonyme, Meronyme, etc. zu Klassen und Konzepten bereithält und ebenfalls zur Strukturierung des Web of Data verwendet wird.

In gleicher Weise wird das SKOS (*Simple Knowledge Organization for the Web*) Vokabular[26] eingesetzt, das zur Abbildung von Thesauri als Linked Data Ressourcen geschaffen wurde [17]. Daneben wird das *Friend-of-a-friend* (FOAF) Vokabular[27] eingesetzt, um Informationen zu Personen und deren Beziehungen untereinander anzugeben.

[25] Siehe http://wordnet.princeton.edu/, aufgerufen am 02.04.2014.
[26] Siehe http://www.w3.org/2004/02/skos/, aufgerufen am 02.04.2014.
[27] Siehe http://www.foaf-project.org/, aufgerufen am 02.04.2014.

2.4 Linked Data benutzen

Das Web of Data stellt eine Vielzahl unterschiedlicher Datenquellen über einen einfachen standardisierten Zugriffsmechanismus (HTTP Protokoll) mit einem einheitlichen Datenmodell (RDF) zur Verfügung. Diese können, sofern zugänglich, auch für eigene Webanwendungen genutzt werden. Da eine Vielzahl dieser Datenquellen ständig aktualisiert wird, wie z. B. die DBpedia, die in regelmäßigen Abständen aus den strukturierten Daten der Wikipedia gewonnen wird, können auch stets die aktuellsten Informationen via Linked Data abgegriffen werden. Ein Vorteil, der sich daraus für Anbieter von Webanwendungen ergibt, liegt darin, dass eine Webseite mit stets aktuellen Daten aus dem Web of Data ergänzt werden kann, ohne dass dazu ein neuer Bearbeitungsaufwand von Anbieterseite her notwendig wird. Abschnitt 2.1 demonstriert am Beispiel der BBC Music Website eine gelungene Integration von Linked Data Ressourcen in die eigene Webseite. Danach werden in Abschn. 2.2 die zur Integration von Linked Data Ressourcen in die eigene Webseite nötigen Schritte beschrieben. Abschnitt 2.3 beschließt dieses Kapitel mit der Vorstellung allgemeiner Architekturprinzipien für Linked Data Anwendungen.

2.4.1 Ein gelungenes Beispiel zur Nutzung von Linked Data Ressourcen

Ein prominentes und frühes Beispiel für eine Webanwendung, die aktuelle Informationen aus dem Web of Data nutzt und dabei selbst auch strukturierte Daten via Linked Data bereithält, ist die Musik-Webplattform der britischen BBC[28], einem der größten öffentlich-rechtlichen Runfunksender weltweit. Bereits 2006 starteten die Bemühungen der BBC, das eigene Radioprogramm mit zusätzlichen Informationen zu Künstlern, deren Werken und darauf bezogenen Events, wie z. B. Konzerten und öffentlichen Auftritten, zu verknüpfen. Bei einem ständig wechselnden Programm stellt sich das Problem, dass der Aufwand zur manuellen Pflege einer Website mit den jeweils aktuellen und relevanten Informationen zum laufenden Programm viel zu kostspielig wäre. Daher lag die Idee nahe, die zusätzlichen Informationen via Linked Data Mechanismen aus öffentlich verfügbaren Datensätzen, wie z. B. DBpedia und MusicBrainz zu beziehen und automatisiert in die Webseiten zu integrieren [18] (vgl. Abb. 2.10). Motiviert durch den Erfolg dieses Konzepts wurden auch weitere Teilbereiche (z. B. News, Wetterberichte, Fernsehprogramme, BBC Archive, etc.) der BBC Programminhalte via Linked Data Mechanismen miteinander verknüpft.

2.4.2 Wie integriere ich Linked Data in meine eigenen Webanwendungen?

Um Linked Data Ressourcen in eigene Webanwendungen zu integrieren, sind prinzipiell zwei Schritte notwendig:

[28] Siehe http://www.bbc.co.uk/music, aufgerufen am 02.04.2014.

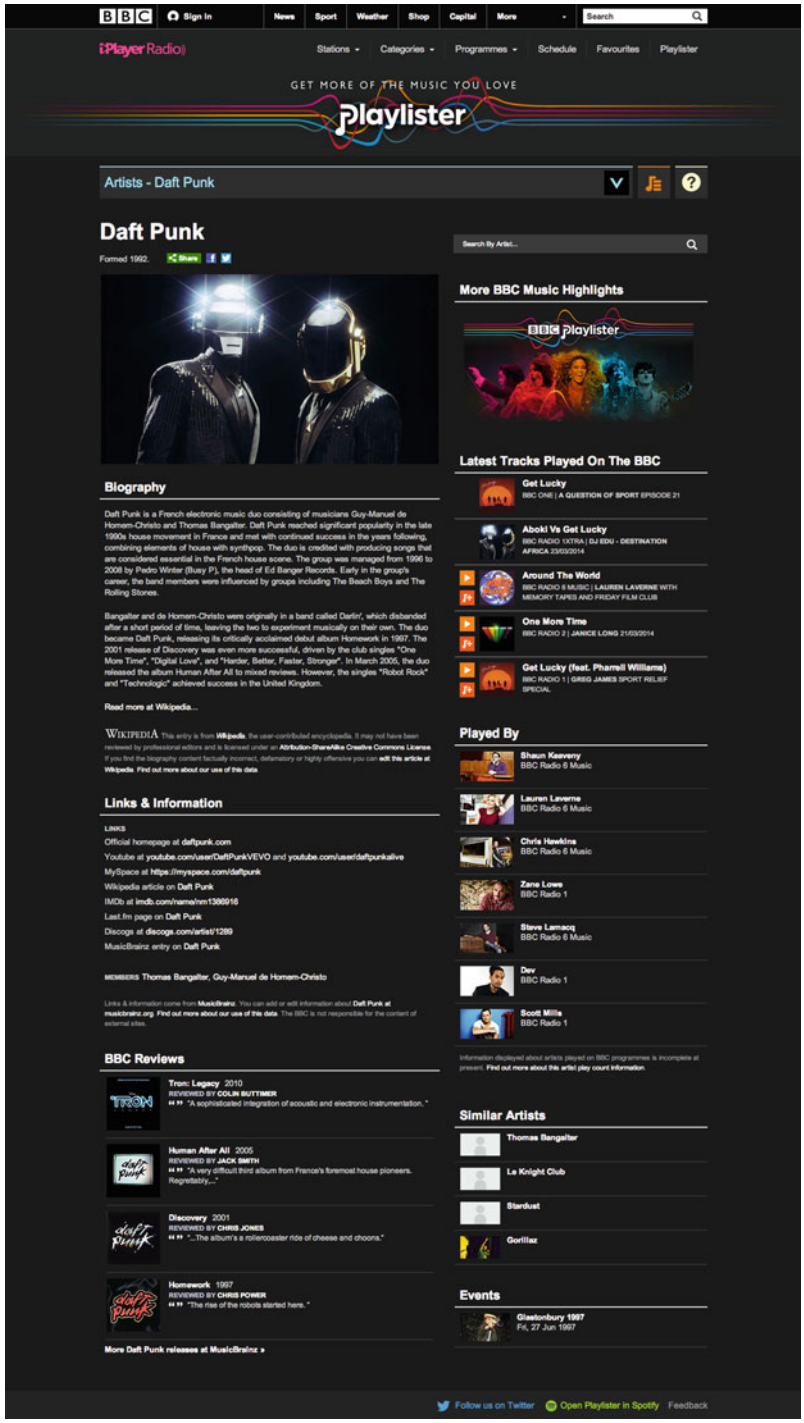


Abb. 2.10 BBC Music Website mit darin integrierten Linked Data Inhalten zur Popband „Daft Punk“

1. Vorbereitung der eigenen Daten
Meist liegen die eigenen Daten noch nicht als Linked Data Ressourcen vor. Einer der ersten Schritte zur Vorbereitung der eigenen Daten besteht darin, URIs für Objekte und Entitäten festzulegen (mehr Details zur Aufbereitung der eigenen Daten als Linked Data finden sich im folgenden Abschn. 2.5). Die nächste Herausforderung besteht darin, geeignete Vokabulare zu finden, mit denen sich die eigenen Daten am besten beschreiben lassen. Die Verwendung bereits bestehender Vokabulare birgt den Vorteil, dass zum Einen der Aufwand zur Entwicklung eines eigenen Vokabulars entfällt bzw. nur darauf reduziert wird, bereits bestehende Vokabulare nötigenfalls zu ergänzen. Andererseits erhöht die Verwendung etablierter Vokabulare den Nutzwert der eigenen Daten für externe Benutzer. Z.B. eignet sich zur Beschreibung von personenbezogenen Daten das Friend-of-a-friend (FOAF) Vokabular[29] oder zum Beschreiben von Ereignissen die LODE Ontologie (Linking Open Descriptions of Events)[30]. Die Umsetzung von strukturierten Daten aus relationalen Datenbanken nach RDF kann meist auf sehr einfache Weise, z. B. mit entsprechenden Werkzeugen wie RDF2RDF (Relational Database to RDF)[31], D2R Server[32], oder Triplify[33] erfolgen.

2. Auswahl geeigneter Zieldatensätze
Die Lokalisierung und Auswahl geeigneter Zieldatensätze aus dem Web of Data kann auf unterschiedliche Art und Weise erfolgen. Natürlich kann man, ausgehend von einer der zentralen Nahtstellen im Web of Data, wie z. B. der DBpedia, die Linked Data Ressourcen und deren Verknüpfungen untereinander manuell erkunden (auch oft als Follow-Your-Nose Prinzip, FYN bezeichnet). Wird ein vorhandener SPARQL-Endpunkt via Semantic Sitemaps Extension beschrieben, kann man aus diesen Sitemaps Extensions Informationen über die vorhandenen Daten gewinnen [20]. Einem ähnlichen Prinzip folgt das „Vocabulary of Interlinked Datasets“ (voiD) zur Beschreibung von Linked Open Data Ressourcen [21] (vgl. Abb. 2.11).

Diese Vorgehensweisen sind meist mit großem Aufwand verbunden. Einen einfacheren Zugang bieten Semantic Web Suchmaschinen, wie z. B. *Sindice*[34] [22] oder *Sigma*[35] [23], die vorhandene Linked Data Ressourcen indexieren und für den Endnutzer auffindbar machen. Die aufgefundenen Linked Data Ressourcen können in unterschiedlicher Form vorliegen. Die einfachste Variante besteht in einem Dump des RDF-Datensatzes, i. e. einer Datei, die in einer RDF-Serialisierung alle RDF-Statements des Datensatzes enthält. Der Webanwender kann solche Dumps über das Web von einem Web-Server beziehen und führt diese Daten seiner lokalen Verarbeitung zu, um seine eigenen Daten

[29] Siehe http://www.foaf-project.org/, aufgerufen am 02.04.2014.
[30] Siehe http://linkedevents.org/ontology/, aufgerufen am 02.04.2014.
[31] Siehe http://www.rdb2rdf.org/, aufgerufen am 02.04.2014.
[32] Siehe http://d2rq.org/d2r-server, aufgerufen am 02.04.2014.
[33] Siehe http://triplify.org/overview, aufgerufen am 02.04.2014.
[34] Siehe http://sindice.com/, aufgerufen am 02.04.2014.
[35] Ebenda.

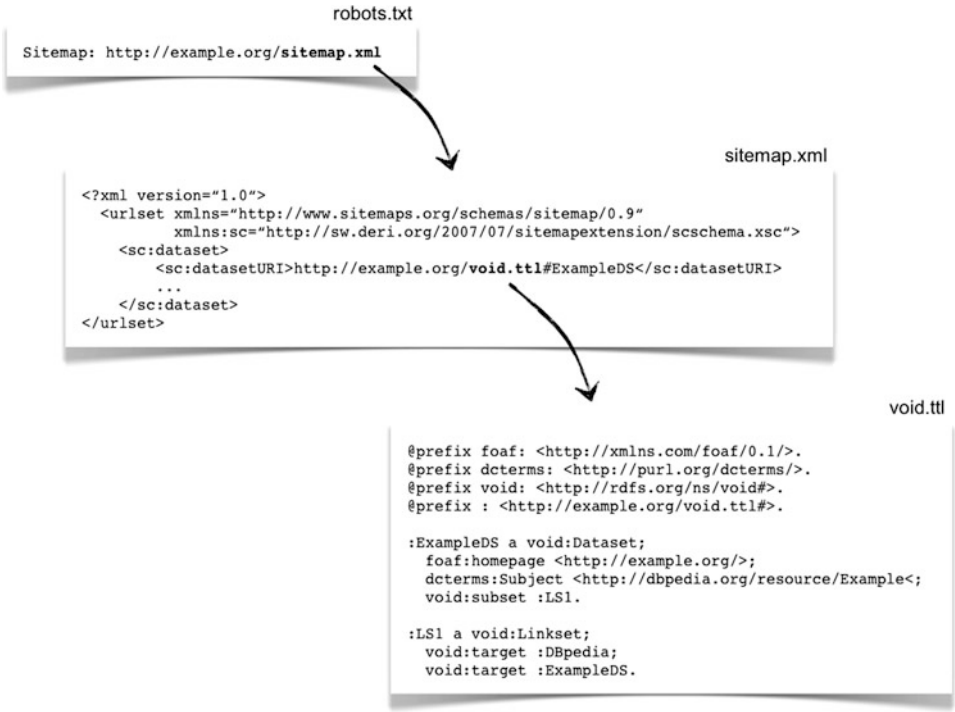


Abb. 2.11 Lokalisierung und Beschreibung von Linked Open Data Ressourcen via Semantic Sitemaps Extension und VoiD [24]

durch den aufgefundenen Datensatz zu ergänzen. Andererseits können Linked Data Inhalte auch via RDFa in vorhandenen HTML-Webseiten eingebettet vorliegen. Dann muss die jeweilige HTML-Seite vom Webanwender gelesen und die RDFa-Anteile daraus mit Hilfe geeigneter Parser extrahiert und weiterverarbeitet werden[36]. Im Idealfall liegen die Linked Data Ressourcen stets aktualisiert und interaktiv zugreifbar auf einem SPARQL-Endpunkt bereit. SPARQL-Endpunkte lassen sich auf einfache Weise via HTTP in der Art einer REST-API[37] abfragen. Die so über eine gezielte SPARQL-Abfrage gewonnenen RDF-Daten können in einem am besten zur Weiterverarbeitung geeigneten Format zurückgegeben und mit den eigenen Daten in Bezug gesetzt und dargestellt werden. Eine Übersicht aktueller Webanwendungen, die Linked Data Ressourcen nutzen, findet sich auf den Linked Data Applications Webseiten des W3C[38].

[36] Siehe z. B. RDFa 1.1 Distiller and Parser, http://www.w3.org/2012/pyrdfa/overview.html, aufgerufen am 02.04.2014.
[37] Representational state transfer, ein Programmierparadigma für Webanwendungen basierend auf der Idee, dass eine URL genau einen Webseiteninhalt als Ergebnis einer serverseitigen Aktion darstellt, vergleichbar mit HTTP für statische Inhalte.
[38] Siehe http://www.w3.org/wiki/sweoig/taskforces/communityprojects/linkingopendata/applications, aufgerufen am 02.04.2014.

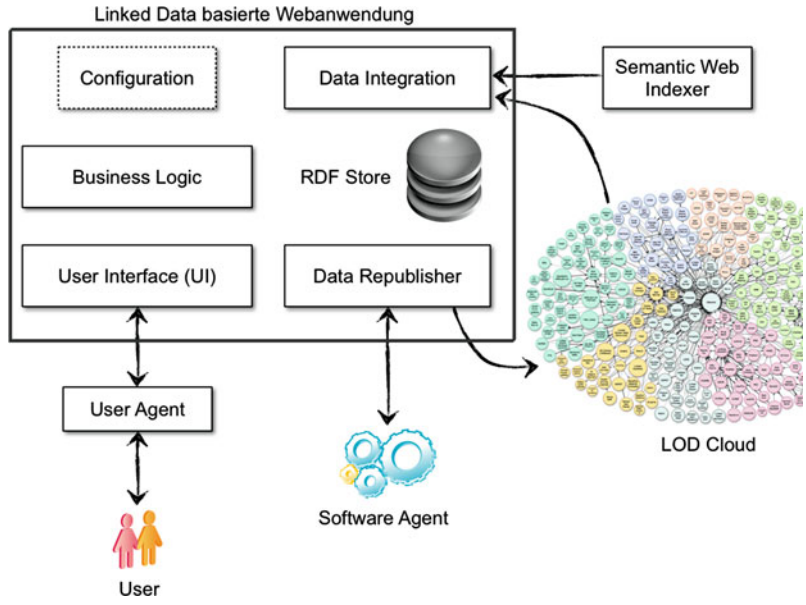


Abb. 2.12 Generisches Konzept einer Linked Data Webanwendung [24]

2.4.3 Linked Data Anwendungen

Wurden die geeigneten Linked Data Ressourcen ausgewählt, müssen sie wie in den vorangegangenen Abschnitten beschrieben abgefragt und entsprechend dem eigenen Verwendungszweck weiterverarbeitet werden. Oft bietet es sich in diesem Zusammenhang an, die eigenen Daten, ergänzt um die gefundenen zusätzlichen Informationen ebenfalls wieder über die Webanwendung oder einen dahinter liegenden SPARQL-Endpunkt als Linked Data zur Verfügung zu stellen, damit aus diesen weiterer Mehrwert gewonnen werden kann.

Der von Hausenblas in [24] angegebenen Beschreibung folgend, liegt einer typischen Linked Data Webanwendung meist eine aus den folgenden generischen Komponenten bestehende Architektur zugrunde:

- Ein lokales RDF Speichersystem, z. B. ein Triplestore, in dem RDF Datenstrukturen abgelegt werden. Dies ist nicht notwendigerweise eine zwingende Voraussetzung. RDF-Daten lassen sich auch in traditionellen relationalen Datenbanksystemen verwalten.
- Wie in traditionellen Webanwendungen benötigt man auch für Linked Data Anwendungen eine Programmlogik (Controller) zur Implementierung der Anwendung (Business Logic) sowie eine geeignete Benutzerschnittstelle (User Interface) zur Interaktion mit dem System.

- Eine Komponente zur Datenintegration, mit der Linked Data aus dem Web of Data bzw. auch über Semantic Web Suchmaschinen (Semantic Web Indexer) abgerufen werden kann
- Eine Komponente zur Veröffentlichung der eigenen Daten als Linked Data (Data Republisher)

Abbildung 2.12 zeigt die generischen Komponenten einer Linked Data Webanwendung.

2.5 Linked Data selbst publizieren

Wie bereits im vorangegangenen Kapitel angesprochen, erhöht sich der Nutzwert der eigenen Daten, wenn diese auch von anderen verwendet werden können. So werden die eigenen Daten mit weiteren Daten in Verbindung gebracht, d. h. es ergeben sich neue Verknüpfungen über das Web of Data. Für den Informationsanbieter ergeben sich neue Möglichkeiten über die breite Verwendung der eigenen Daten auch einen Rückfluss auf die eigene Webpräsenz und damit eventuell auch auf das eigene Angebot zu generieren. Die Sichtbarkeit im Web wird dadurch erhöht, was sich durchaus auch in höheren Besucherzahlen auf der eigenen Webpräsenz und damit positiv auf das Geschäft auswirken kann. Daher macht es Sinn, stets über die Möglichkeit der Veröffentlichung eigener Daten im Web of Data nachzudenken.

2.5.1 Mit dem 5-Sterne-Plan zur Linked Data Publikation

Für den Anwender bieten sich verschiedene Möglichkeiten, die eigenen Daten in das Web of Data zu integrieren. Tim Berners-Lee hat als Urheber des Web of Data einen Vorschlag zur Qualitätsbewertung der eigenen Bemühungen im Linked Data Publishing angeregt, den sogenannten „*Five Star Plan to Publishing Linked Data*“[39]. Darin werden fünf unterschiedliche Qualitätsstufen bzgl. der Veröffentlichung der eigenen Daten unterschieden:

⋆ **Eigene Daten im Web unter offener Lizenz frei verfügbar machen** Dabei kommt es nicht darauf an, ob die vorliegenden Daten auch tatsächlich in maschinenlesbarer Form vorliegen, sondern z. B. als gescanntes PDF-Dokument. Wichtig ist einzig die Verfügbarkeit der Daten über das WWW. Während Daten zuvor meist nur innerhalb von Webanwendungen lokal verfügbar und damit für den Rest der Welt verschlossen waren, erhöht eine freie Verfügbarkeit und damit verbundene vielfache Nutzung die eigene Sichtbarkeit und auch die Zugriffe auf die eigene Webpräsenz. Natürlich kann es immer auch gute Gründe dafür geben, die eigenen Daten nicht frei verfügbar bereitzustellen, wie z. B. die

[39] Tim Berners-Lee: Linked Data – Design Issues, http://www.w3.org/designissues/linkeddata.html, aufgerufen am 02.04.2014.

Wahrung von Wettbewerbsvorteilen, Datenschutz, oder bereits bestehende, restriktivere Lizenzvereinbarungen.

⋆⋆ **Eigene Daten als maschinenlesbare, strukturierte Daten unter offener Lizenz frei verfügbar machen** In welcher Form die jetzt maschinenlesbaren strukturierten Daten auf dieser Stufe vorliegen müssen, ist ohne Belang, d. h. anstelle der gescannten PDF-Datei können die Daten auch über proprietäre Datenformate, wie z. B. Microsoft Excel-Spreadsheets, XML oder JSON vorliegen. Dies erhöht gegenüber der vorab genannten, einfacheren Variante die unkomplizierte Wiederverwendbarkeit durch Dritte und damit auch die Kombination der eigenen Daten mit Drittdaten.

⋆⋆⋆ **Eigene Daten als maschinenlesbare, strukturierte Daten in offenen Formaten unter offener Lizenz frei verfügbar machen** Lagen die eigenen Daten in der Variante zuvor noch in proprietären Formaten vor, zu denen spezielle, meist nur kommerziell verfügbare Werkzeuge zum Lesen und Interpretieren notwendig waren, werden die Daten jetzt in öffentlichen (non-proprietären) Formaten, wie z. B. CSV, XML, JSON oder auch RDF kodiert und frei über das Web zur Verfügung gestellt. Die einfache Wiederverwendbarkeit der Daten steht dabei im Vordergrund und erhöht so den damit erzielbaren Mehrwert.

⋆⋆⋆⋆ **Eigene Daten als maschinenlesbare, strukturierte Daten in W3C-standardisierten, offenen Formaten unter offener Lizenz frei verfügbar machen und URIs zur Identifikation verwenden** Objekte bzw. Entitäten in den eigenen Daten werden über einen URI identifiziert und damit global eindeutig adressierbar und abrufbar. Im Gegensatz zu traditionellen Wegen, Daten im Web zu veröffentlichen, bietet die Verwendung von URIs den Vorteil, dass die Daten nicht nur als Ganzes, sondern feingranular auf der einzelnen Datenobjekt-Ebene identifiziert, adressiert und weiter verwendet werden können. Damit verbunden werden Daten, wie bereits beschrieben, dereferenzierbar und mit anderen Daten direkt vernetzbar. Werden die Daten zudem via RDF(S) kodiert und via SPARQL abfragbar, werden sie in einem einheitlichen einfachen Datenmodell abgelegt und über einen sehr einfachen Transportmechanismus (HTTP) auch über gängige Firewalleinschränkungen hinweg abrufbar.

⋆⋆⋆⋆⋆ **Zusätzlich zu den zuvor genannten Voraussetzungen werden die eigenen Daten mit bereits bestehenden Daten des Web of Data verlinkt** Hyperlinks kann man als die „Seele“ des World Wide Webs betrachten. In gleicher Weise dienen Hyperlinks auch der Vernetzung im Web of Data, ohne die die Daten jeweils isoliert und für sich alleine stünden. Erst wenn die eigenen Daten mit anderen in Bezug gesetzt werden können, entsteht ein Mehrwert. Auf diese Weise können Softwareagenten automatisch Verknüpfungen verfolgen und so neue interessante Daten und Fakten entdecken.

2.5.2 Wie veröffentliche ich meine Daten im Web of Data?

Soll es jetzt daran gehen, bestehende Daten im Unternehmen als Linked Open Data zu veröffentlichen, müssen zunächst die jeweiligen Datenverantwortlichen vom Vorteil dieser Maßnahme überzeugt werden. Die jeweils dazu verwendeten W3C Standards sind gut dokumentiert und es existieren verschiedene Vorgehensmodelle, denen man bei der Bereitstellung folgen kann, wobei diese weitgehend demselben Muster in Bezug auf Spezifikation, Modellierung und Publikation folgen (z. B. [25]). Als nächster Schritt müssen die Daten ausgewählt werden, die zur Veröffentlichung vorgesehen sind. Idealerweise handelt es sich dabei um Daten, die verbunden mit bereits existierenden Daten aus dem Web of Data einen Mehrwert liefern. Insbesondere macht dies Sinn, wenn z. B. Personen oder Ortsangaben mit bestehenden Ressourcen verknüpft werden. Zu diesem Zweck müssen die eigenen Daten gemäß der Linked Data Prinzipien modelliert werden. Zur Überführung bestehender relationaler Daten in eine RDF Darstellung kann auf existierende Werkzeuge (z. B. RDB2RDF, triplify, etc.) zurückgegriffen werden. Zu diesem Schritt zählt auch die Benennung (Identifikation) der einzelnen Datenobjekte über URIs. Müssen zu diesem Zweck neue URIs eingeführt werden, sollten einige Designprinzipien berücksichtigt werden (vgl. auch [7]):

- Verwendung von dereferenzierbaren HTTP-URIs, damit sowohl eine bestimmte Ressource damit adressiert als auch Informationen über diese Ressource an einen menschlichen Benutzer oder an einen Softwareagenten weitergegeben werden können.
- Die via Dereferenzierung an einen Softwareagenten zurückgelieferte Beschreibung der Daten sollte in maschinenlesbarer Form, d. h. als RDF Beschreibung ausgeliefert werden.
- Die verwendeten URIs sollten möglichst dauerhaft sein und keine veränderbaren Anteile, wie z. B. Session-Tokens beinhalten.
- Man sollte nicht den Fehler begehen, zusätzliche Informationen in die URIs zu kodieren, die via Interpretation ausgelesen bzw. für einen Menschen daraus erschließbar sind. Alle Informationen zu einer Ressource sollten über die vom Web-Server gelieferte dereferenzierbare Beschreibung geliefert werden.

Zur Beschreibung der eigenen Daten sollten, falls vorhanden, bereits bestehende Vokabulare verwendet werden. Das W3C hat zu diesem Zweck Standardvokabulare zusammengestellt und publiziert, die zur Beschreibung bestehender Datenstrukturen verwendet werden können, wie z. B. das Data Catalog Vocabulary (DCAT) [26], die Organization Ontology [27], oder das RDF Datacube Vocabulary [28]. Um geeignete Vokabulare im Web of Data aufzuspüren, können auch Semantic Web Suchmaschinen, wie z. B. Sig.ma oder Sindice verwendet werden. Auf alle Fälle sollte man sich einen Überblick über existierende Vokabulare verschaffen und diese bzgl. ihrer Anwendbarkeit auf die eigenen Daten untersuchen, um diese ev. zu erweitern, bevor man sich an das Design eines neuen RDF-

Vokabulars wagt. Als Design- und Qualitätskriterien für RDF-Vokabulare können bei der Auswahl gelten:

- Die Herkunft der Vokabulare sollte nachprüfbar sein und die Quelle sollte als vertrauenswürdig gelten.
- Verwendete Vokabulare sollten dauerhafte URIs verwenden.
- Das Vokabular selbst sollte gut dokumentiert sein.
- Vokabulare sollten versioniert werden, wobei die Versionshistorie gut dokumentiert sein sollte.
- Die Vokabulare sollten möglichst selbstbeschreibend sein, d. h. jeder Term eines Vokabulars sollte einen Bezeichner, eine Definition und ev. Kommentare vorsehen.
- Zum breiteren Nutzen sollten Vokabulare in mehreren Sprachen dokumentiert werden.
- Vokabulare sollten auch von anderen Datensätzen verwendet werden.
- Vokabulare sollten dauerhaft verfügbar sein.

Um Vokabulare miteinander in Bezug zu setzen, können spezielle standardisierte Vokabulare, wie z. B. das Simple Knowledge Organization System (SKOS), verwendet werden [29].

Die entsprechend den ausgewählten Vokabularen beschriebenen Daten können jetzt mit Hilfe unterschiedlicher RDF Serialisierungen (RDF/XML, N3, Turtle, JSON-LD, RDFa) kodiert im Web angeboten werden. Die gewählte Form der Serialisierung liegt dabei jeweils im Interesse des Datenurhebers. Keine der genannten Serialisierungen besitzt konzeptionell einen Vorteil gegenüber einer anderen. Unterschiede bestehen lediglich bzgl. ihrer Komplexität und Lesbarkeit für den menschlichen Betrachter und in der unterschiedlichen Weiterverarbeitung.

2.5.3 Ergänzende Maßnahmen bei der Publikation von Linked Data

Generell sollten zu den eigenen veröffentlichten Daten auch Metadaten vorgehalten werden, die Angaben wie z. B. Herausgeber bzw. Urheber der Daten, Datum ihrer Erzeugung, Datum eventueller Modifikationen, Versionsnummer, Updatefrequenz und ev. auch eine Kontaktadresse beinhalten. Zu diesem Zweck kann das Semantic Sitemaps Protokoll[40] als Erweiterung des bekannten Sitemaps Protokolls verwendet werden, das Informationen zu Datenquellen im Web für Suchmaschinen und andere Webcrawler bereithält (vgl. Kap. 4) [20]. Ein weiterer Standard zur Veröffentlichung von Metainformationen zu Linked Data Vokabularen ist das voiD Vokabular (Vocabulary of Interlinked Datasets) [30]. Der Vorteil der Verwendung von voiD besteht darin, dass die Metadaten zum eigenen Vokabular selbst auch als RDF-Statements bereitgehalten werden können. Ein weiteres, weit verbreitetes Vokabular zur Bereitstellung von Informationen zum Urheber bzw. der Herkunft

[40] Siehe http://sw.deri.org/2007/07/sitemapextension/, aufgerufen am 02.04.2014.

(Provenienz) der Daten ist das im Bibliotheksbereich schon lange etablierte Dublin Core Vokabular, das auch als RDF Vokabular vorliegt und Properties wie z. B. `dc:creator`, `dc:publisher` oder `dc:date` bereithält. Zudem sollten die eigenen Daten stets unter eine die Urheberschaft und die Weiterverwendung regelnde Lizenz gestellt werden. Auch wenn Daten öffentlich im Web angeboten werden, heißt dies noch lange nicht, dass diese Daten in jeder Form, von jedem und zu jedem Zweck wiederverwendet werden können. Aus diesem Grund sollte der Urheber der Daten entscheiden, an welche Form der Wiederverwendung die eigenen Daten gebunden werden sollen [31]. Dies gibt Auskunft über die Möglichkeit, ob die Daten auch im kommerziellen Umfeld verwendet werden können oder ob z. B. der Urheber der Daten bei einer Wiederverwendung zu nennen ist. Das folgende RDF Codebeispiel zeigt, wie Metadaten für einen Linked Data Datensatz angegeben werden können.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix void: <http://rdfs.org/ns/void#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix cc: <http://creativecommons.org/ns#>.

<http://example.org/big-example-dataset>
  a void:Dataset;
  dc:title "Big Example Dataset";
  dcterms:date "2014-03-24T12:34:56Z"^^xsd:dateTime;
  dcterms:creator <http://example.org/>;
  dcterms:description "This is an example BigData Dataset";
  void:feature <http://www.w3.org/ns/formats/RDF_XML>;
  void:dataDump
        <http://example.org/big-example-dataset.rdf>;
  void:triples 100000000;
  foaf:primaryTopic <http://example.org/example-datasets>;
  cc:license
        <http://creativecommons.org/licenses/by-sa/3.0/>.
```

Entsprechend den Linked Data Publishing Prinzipien sollten die eigenen Daten mit bereits bestehenden Linked Data Ressourcen verknüpft werden, um so selbst ein Teil des Web of Data zu werden. Umgekehrt kann der eigene Datensatz auch leichter entdeckt werden, wenn bestehende Linked Data Ressourcen selbst mit dem neuen Datensatz via RDF-Links verknüpft werden. Zu diesem Zweck müssen die Verantwortlichen für diese Datensätze zunächst von dieser Möglichkeit überzeugt werden. Ein ausschlaggebender Punkt könnte der bei der Verlinkung entstehende Mehrwert sein, der durch die neuen Daten bereitsteht. Um es den Verantwortlichen leichter zu machen, ist es empfehlenswert, die betreffenden RDF-Links selbst vorzubereiten und den Urhebern externer Datensätze zur Integration anzubieten. Auf diese Weise wurden z. B. zahlreiche Datensätze mit

den Daten der DBpedia verlinkt. Die Art der Verlinkung (Identity-Links, Relationship-Links, Vocabulary Links, vgl. Abschn. 2.4) bestimmt das dabei verwendete Vokabular. Die einfachste, qualitativ wertvollste, aber auch aufwändigste Form der Verlinkung ist die manuelle Verlinkung, in der der Urheber des Datensatzes für jede Entität prüft, ob und wie eine Verlinkung zu einem externen Datensatz durchzuführen ist. Daneben existieren auch Werkzeuge zur automatisierten Verlinkung. Generell folgen diese meist einem der folgenden Ansätze:

- **Schlüssel-basierte Ansätze (key-based)**
  Die einfachste automatisierte Variante identifiziert gemeinsame Namensschemata (z. B. ISBN, EAN, DOI), die in den miteinander zu verknüpfenden Datensätzen verwendet werden.
- **Ähnlichkeitsbasierte Ansätze**
  Weitaus komplexer sind Verfahren, die die Ähnlichkeit von Entitäten aus zu verknüpfenden Datensätzen (meist heuristisch) ermitteln. Dabei werden einander möglichst ähnliche Entitäten bestimmt, die z. B. gleiche oder ähnliche Eigenschaften besitzen. Überschreitet die ermittelte Ähnlichkeit einen zuvor festgelegten Schwellwert, werden die Entitäten gleichgesetzt und eine Identitätsverlinkung vorgenommen. Beispiele für Werkzeuge zur automatischen Verlinkung von Linked Data Datensätzen sind das *Silk-Link Discovery Framework*[41] [32] oder *LIMES-Link Discovery Framework for Metric Spaces*[42] [33].

2.5.4 In welcher Form biete ich meine Daten als Linked Data im Web an?

Der Zugriff auf Linked Data Ressourcen kann im Web auf unterschiedliche Arten erfolgen. Der Datenbereitsteller muss dabei entscheiden, welche Arten des Zugriffs zur Verfügung gestellt werden sollen. Generell unterscheidet man

- Zugriff auf ein einzelnes Datenobjekt via Dereferenzierung eines URI
- Zugriff über eine RESTful API
- Zugriff über einen SPARQL-Endpunkt
- Zugriff über im HTML-Dokument bereitgestelltes RDFa
- Download eines RDF Datendumps.

Sollen statische RDF-Datendumps zum Download bereitgestellt werden – der übrigens einfachste Weg der Publikation – sollten diese in der RDF/XML Serialisierung publiziert werden, da für dieses Format die am weitesten verbreitete Werkzeugunterstützung besteht. Die Publikation als statischer RDF-Dump bietet sich vor allem immer dann an,

[41] Siehe http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/, aufgerufen am 02.04.2014.
[42] Siehe http://aksw.org/projects/limes.html, aufgerufen am 02.04.2014.

wenn relativ kleine Datenmengen veröffentlicht werden sollen, die nur selten Veränderungen unterworfen sind, wie z. B. Daten zu einer persönlichen Homepage. Ein anderer Fall wäre die Publikation eines RDF-Datendumps, der über ein Softwarewerkzeug oder ein Programm automatisch erzeugt wurde. Um die Dereferenzierbarkeit der Daten zu ermöglichen, muss auch in diesem Fall der Web-Server via Content-Negotiation je nach Art der Anfrage entweder eine für den Menschen lesbare Version der Daten oder eine für ein Softwarewerkzeug bestimmte, maschinenlesbare Version der Daten bereithalten. Letzteres wäre der RDF-Datendump, der über die entsprechende MIME type Anforderung im HTTP-Request spezifiziert wird (`application/rdf+xml`, vgl. Abschn. 2.2). Um die Daten auch über reguläre Web-Suchmaschinen auffindbar zu machen, kann die HTML-Webseite, die die für den Menschen lesbare Informationsseite zu den bereitgestellten Daten enthält, ein sogenanntes Autodiscovery-Pattern im Header der HTML-Seite aufnehmen:

```
<link rel="alternate"
        type="application/rdf+xml" href="example.rdf">
```

RDF Daten lassen sich auch direkt in ein HTML-Dokument via RDFa integrieren. Diese Möglichkeit bietet sich z. B. auch für automatisch erzeugte dynamische Webseiten an, die mit Hilfe eines Content Management Systems verwaltet werden. Dabei muss das jeweils verwendete Template, das die Inhalte des HTML-Dokuments verwaltet, um den jeweilig einzubettenden RDFa-Code erweitert werden. Werden HTML-Informationen und RDF-Daten in separaten Dokumenten verwaltet, besteht das Problem der Synchronizität, sobald sich der Inhalt eines der Dokumente verändert. Via RDFa können alle Veränderungen in einem einzigen Dokument vorgenommen werden und das Synchronisationsproblem wird für diesen Fall gegenstandslos. RDFa wird in gleicher Weise wie microformats oder microdata über spezielle Attribute in die HTML-Tags des HTML-Dokuments nahtlos integriert. Die Einbettung von RDFa in HTML-Code zeigt das folgende Beispiel, in dem in einem HTML-Paragraph-Tag über das Attribut „`vocab`“ ein verwendetes RDF-Vokabular angegeben wird, aus dem die über die Attribute „`property`“ bzw. „`typeof`“ verwendeten RDF-Properties oder RDF-Klassen referenziert werden.

```
<p vocab="http://xmlns.com/foaf/0.1/"
     resource="#harald" typeof="Person">
 My name is <span property="name">Harald Sack</span>
 and my phone number
 is
<span property="phone">1-800-555-0527</span>.
</p>
```

Wird zur Verwaltung der Linked Data Ressourcen ein RDF Triplestore eingesetzt, bietet dieses Werkzeug vielfältige Konfigurationsmöglichkeiten, über die der Administrator einer Webseite festlegen kann, welche Teile der vorhandenen Linked Data Ressourcen

auf welche Weise öffentlich zugreifbar gemacht werden. Bietet der Triplestore einen öffentlichen SPARQL Endpunkt, können Benutzer einfache sowie komplexere Abfragen in den zur Verfügung gestellten Linked Data Ressourcen ausführen. Triplestores sind spezielle Datenbanken, die auf die Verwaltung von RDF-Tripeln hin optimiert wurden. Stark vereinfacht kann man sich einen Triplestore als relationale Datenbank vorstellen, die genau eine Tabelle mit drei Spalten (Subject, Property, Object) bereithält, und die für einen schnelleren Zugriff mit verschiedenartigen Kombinationen von Indices ausgestattet ist. SPARQL-Anfragen werden dann in SQL-Anfragen übersetzt und bzgl. der Verarbeitungskomplexität optimiert. Ein weit verbreitetes Werkzeug, das zur webbasierten Auslieferung und Darstellung von Linked Data Ressourcen auf einen SPARQL-Endpunkt aufgesetzt werden kann, ist *Pubby*[43]. Pubby kodiert eine eintreffende Anfrage zur URI-Dereferenzierung in eine SPARQL-Abfrage an den Tripelstore um, stellt diese an den SPARQL-Endpunkt und liefert das Ergebnis entsprechend der Content-Negotiation entweder als HTML-Dokument oder als RDF-Serialisierung aus.

Der letzte und auch einer der wichtigsten Schritte bei der Veröffentlichung von Linked Open Data Ressourcen besteht auch in der Bekanntmachung des eigenen Datenangebots. Zu diesem Zweck empfiehlt es sich, das *Comprehensive Knowledge Archive Network* (CKAN)[44] zu nutzen und seine eigenen Daten dort zur freien Verwendung anzumelden (vgl. Abschn. 3.1). Um in diesen Katalog frei verfügbarer Datensätze des Web of Data aufgenommen zu werden, muss der eigene Datensatz mindestens die folgenden Bedingungen erfüllen:

- Alle Datenobjekte müssen über dereferenzierbare URIs abgerufen werden können.
- Der eigene Datensatz beinhaltet wenigstens 50 RDF-Links in andere Datensätze bzw. wenigstens ein externer Datensatz verweist mit mindestens 50 RDF-Links auf den eigenen Datensatz[45].

CKAN bietet darüber hinaus die Möglichkeit, den eigenen Datensatz mit Hilfe vielfältiger Attribute und Metadaten zu beschreiben, damit dieser besser von potenziellen Nutzern gefunden und verwendet werden kann.

Abbildung 2.13 stellt zusammenfassend verschiedene Möglichkeiten der Publikation von Linked Data Ressourcen zusammen [2]. Der Ausgangspunkt sind dabei oft bereits als strukturierte Daten vorliegende Datensätze, die z. B. ursprünglich in einer relationalen Datenbank vorliegen und über einen RDF Wrapper als Linked Data aufbereitet abgerufen werden können. RDF Wrapper erlauben es dem Benutzer selbst festzulegen, auf welche Weise vorhandene Tabellen in RDF Statements umgesetzt werden sollen und welche RDF Vokabularien für die Verknüpfung (Mapping) mit dem Web of Data genutzt werden sollen. Stellt eine bereits vorhandene Webanwendung ihre Daten über eine spezielle Pro-

[43] Siehe http://wifo5-03.informatik.uni-mannheim.de/pubby/, aufgerufen am 02.04.2014.
[44] Siehe http://datahub.io/group/lodcloud, aufgerufen am 02.04.2014.
[45] Siehe http://www.w3.org/wiki/taskforces/communityprojects/linkingopendata/datasets/ckanmetainformation, aufgerufen am 02.04.2014.

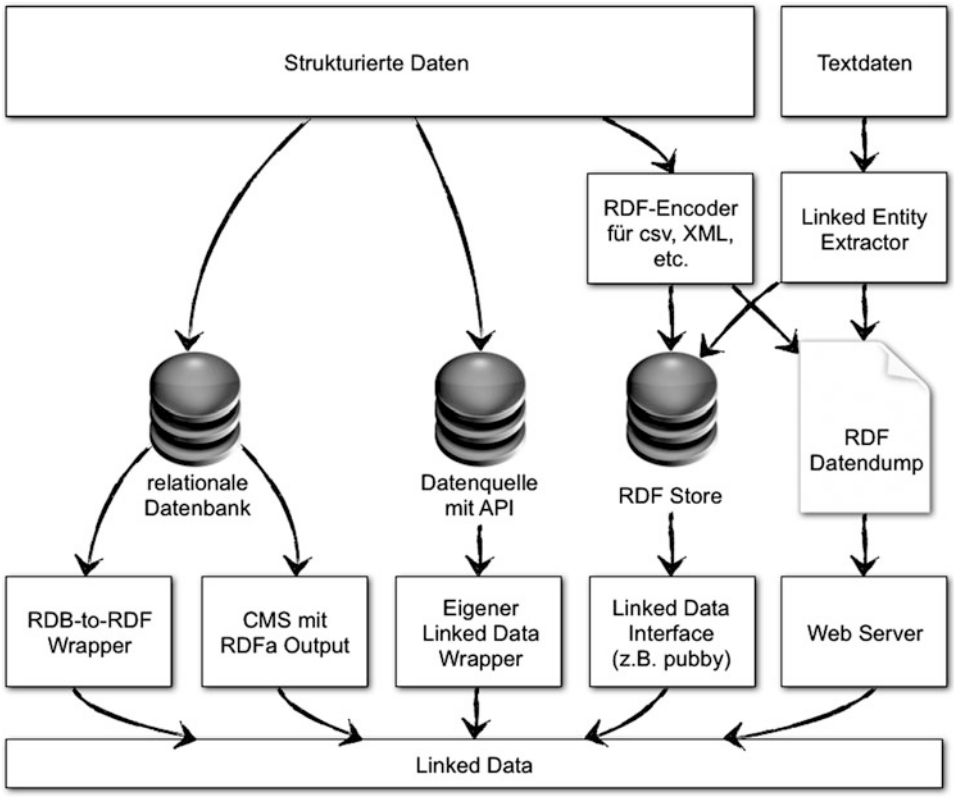


Abb. 2.13 Linked Data Publishing und damit verbundene Workflows [2]

grammschnittstelle (API) zur Verfügung, muss meist eine eigene Wrapper-Anwendung konzipiert und implementiert werden, um diese Daten in RDF-Statements zu übersetzen, sie auf geeignete Vokabulare zu mappen und so mit dem Web of Data zu verknüpfen. Generell folgen RDF/Linked Data Wrapper stets dem folgenden allgemeinen Schema:

- Den Datenressourcen, die ein API liefert, werden geeignete HTTP URIs zugewiesen.
- Wird eine dieser neuen HTTP URIs dereferenziert (Anfrage nach `application/rdf+xml`), wandelt der Wrapper diese HTTP Anfrage in einen Aufruf der Programmschnittstelle (API) um.
- Die Ergebnisse des API-Aufrufs werden vom Wrapper in RDF-Statements übersetzt und via HTTP als Antwort auf den ursprünglichen Request zur Dereferenzierung zurückgeliefert.

Dabei sollte darauf geachtet werden, dass auch in diesem Falle RDF-Links zu externen Linked Data Datensätzen mit den vom API gelieferten Daten verknüpft werden und dass der Betreiber des Wrappers auch die Nutzungsrechte der originalen Datenquelle beachtet.

Liegen die zu publizierenden Daten ursprünglich als CSV-Datendumps, statische XML-Dateien bzw. Daten in einem anderen, proprietären Dateiformat vor, müssen diese entsprechend der Linked Data Prinzipien in RDF Statements übersetzt werden, die anschließend entweder als statischer RDF-Datendump über einen Web-Server oder über einen SPARQL-Endpunkt im Web of Data zur Verfügung gestellt werden. Auch vorhandene Textdokumente lassen sich auf einfache Weise mit dem Web of Data verknüpfen, indem Werkzeuge zur semantischen Annotation genutzt werden, die meist mit linguistischen Mitteln die Texte untersuchen, sogenannte Named-Entities (bedeutungstragende Entitäten) identifizieren und diese mit einem korrespondierenden Linked Data URI annotieren. Soll z. B. der Satz „*Albert Einstein ist der Urheber der speziellen Relativitätstheorie*." semantisch annotiert werden, können die Entitäten „*Albert Einstein*" und „*spezielle Relativitätstheorie*" als bedeutungstragende Entitäten identifiziert und mit den korrespondierenden Entitäten aus der DBpedia `dbpedia:Albert_Einstein` und `dbpedia:Special_relativity` annotiert werden. Die Publikation semantischer Annotationen zusammen mit den zugrundeliegenden Texten erhöht deren Wiederauffindbarkeit und ermöglicht es Suchmaschinen und anderen Webanwendungen, die Texte mit zusätzlichen Informationen anzureichern und darzustellen. Verbreitete Werkzeuge zur Linked Entity Extraction sind z. B. OpenCalais[46] oder DBpedia Spotlight[47].

2.6 Stand der Technik und Forschungsherausforderungen

In den vergangenen Jahren ist die Zahl der verfügbaren Linked Data Datensätze stetig angewachsen. Anwendungen sowohl im Forschungsbereich als auch unter kommerziellen Gesichtspunkten machen heute immer häufiger Gebrauch von Linked Data Ressourcen. Insbesondere stellt die kommerzielle Nutzung von Linked Data Ressourcen hohe Ansprüche an die Datenqualität, d. h. Datenaktualität, Widerspruchs- und Fehlerfreiheit. Mängel in der Datenqualität von Linked Data Ressourcen lassen heute noch viele Entscheider vor der Verwendung dieser Daten- und Informationsquelle zurückschrecken. Daher liegt ein besonderer Augenmerk in der Forschung auf der Behebung bestehender Linked Data Datenqualitätsmängel (Linked Data Cleansing). Insbesondere die Nutzung der DBpedia als zentraler Dreh- und Angelpunkt zur Identifikation von Ressourcen im Web of Data beinhaltet aufgrund zahlreicher restriktiver Rahmenbedingungen immer wieder Fehler und Inkonsistenzen. Dies liegt weitgehend darin begründet, dass bereits Wikipedia als Grundlage der DBpedia aufgrund der an ihrer Entstehung und Wartung beteiligten Laien vielfach die erste Fehlerquelle darstellt. Die Extraktion strukturierter Daten aus der Wikipedia setzt die konsistente Wahrung vorgeschriebener Editierstandards in der Wikipedia voraus, die von Wikipedia-Autoren oft unwissentlich verletzt werden. Zudem sind Informationen in der Wikipedia oft auch unvollständig bzw. ungleichmäßig thematisch verteilt.

[46] Siehe http://www.opencalais.com/, aufgerufen am 02.04.2014.
[47] Siehe http://dbpedia-spotlight.github.io/demo/, aufgerufen am 02.04.2014.

Ein wichtiger, weiterer Schritt zur Hebung der Datenqualität von Linked Data Ressourcen ist die Einbeziehung von Provenienzinformationen. Hinter publizierten Daten und Informationen verbergen sich meist keine objektiven Wahrheiten, sondern sie sind stets aus dem Blickwinkel ihres Urhebers zu betrachten bzw. auch bzgl. ihrer Qualität und Konfidenz zu beurteilen. Zu einem Fakt kann es mehrere Meinungen geben (z. B. welche Stadt ist die attraktivste Stadt eines Landes). Fakten verändern sich mit der Zeit (z. B. die Einwohnerzahl einer Stadt). Erst wenn zu einem Fakt Urheber sowie ein Zeitpunkt seiner Gültigkeit veröffentlicht wird, kann damit sicher in der Weiterverarbeitung umgegangen werden.

Einer der substantiellen Kritikpunkte an bestehenden Linked Data Ressourcen ist aktuell noch ihre mangelnde Kohärenz, d. h. bezogen auf die Zahl der via Linked Data beschriebenen Ressourcen ist die Zahl der vorhandenen Verknüpfungen relativ gering. Die vorhandenen Verlinkungen sind meist manuell gepflegt und daher auch mit einem sehr hohen Aufwand verbunden. Die Nutzung und Verwendung von Linked Data Ressourcen im großen Stil, etwa im Sinne von Big Data Anwendungen stecken heute immer noch in den Kinderschuhen, insbesondere wenn es dabei um die Fusion oder Integration großer heterogener Datenmengen geht. Bislang existieren kaum Endbenutzer-bezogene Anwendungen, so dass eine weitreichende Verbreitung unter der Ausnutzung von Netzwerkeffekten aktuell kaum stattfindet. Allerdings hat auch die Industrie die mit Linked Data verbundenen Vorteile für sich entdeckt. Diese beziehen sich nicht immer auch auf Linked Open Data, d. h. oft werden auch nur unternehmensinterne Datenressourcen gemäß den Linked Data Prinzipien miteinander verknüpft und im Sinne einer unternehmensweiten einheitlichen Datenintegration genutzt. Gerade auch im Unternehmenskontext wachsen die zur Verfügung stehenden Datenressourcen quer durch alle Abteilungen an. Die Anwendung der Linked Data Prinzipien mit ihrem einheitlichen RDF-Datenmodell und den damit verbundenen einfachen feingranularen Zugriffsmechanismen bieten gegenüber traditionellen Ansätzen der Datenintegration unbestreitbare Vorteile, die sich auch in besserer Wartbarkeit verbunden mit geringeren Kosten niederschlagen.

Literatur

1. Schiller, Friedrich 1789. *Was heißt und zu welchem Ende studiert man Universalgeschichte? (Antrittsvorlesung in Jena, 26. Mai 1789)*. Jena: Akademische Buchhandlung
2. Heath, Tom, und Christian Bizer. 2011. *Linked Data – Evolving the Web into a Global Data Space*. Morgan & Claypool Publishers
3. Berners-Lee, Tim 1989. *Information Management: A Proposal, CERN*. http://www.w3.org/History/1989/proposal.html
4. Gruber, Tom 2008. Ontology. In *Encyclopedia of Database Systems*, Hrsg. Ling Liu, M.Tamer. Özsu: Springer-Verlag
5. Durrell, W.R. 1985. *Data Administration: A Practical Guide to Data Administration*. McGraw-Hill

6. Berners-Lee, T., R. Fielding, und L. Masinter. 2005. *RFC 3986, Uniform Resource Identifier (URI): Generic Syntax*. Internet Engineering Task Force
7. Sauermann, Leo, und Richard Cyganiak. 2008. *Cool URIs for the Semantic Web. W3C Interest Group, Note, W3C*. http://www.w3.org/TR/cooluris/
8. Herman, Ivan, Ben Adida, Manu Sporny, und Mark Birbeck. 2013. *RDFa 1.1 Primer – Second Edition, Rich Structured Data Markup for Web Documents, W3C Working Group Note*. http://www.w3.org/TR/xhtml-rdfa-primer/
9. Davis, Ian, Thomas Steiner, und Arnaud J Le Hors (Hrsg.). 2013. *RDF 1.1 JSON Alternate Serialization (RDF/JSON)*. W3C Editor's Draft 07 November 2013, https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-json/index.html
10. Schreiber, Guus, und Yves Raimond. 2014. *RDF 1.1 Primer, W3C Working Group Note*. http://www.w3.org/TR/rdf11-primer/
11. Brickley, Dan, und R.V. Guha. 2014. *RDF Schema 1.1, W3C Recommendation*. http://www.w3.org/TR/rdf-schema/
12. Hitzler, Pascal, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, und Sebastian Rudolph. 2012. *OWL 2 Web Ontology Language: Primer (Second Edition), W3C Recommendation*. http://www.w3.org/TR/owl2-primer/
13. Kifer, Michael, und Harold Boley (Hrsg.). 2013. *RIF Overview* (Second Edition). W3C Working Group Note 5 February 2013, http://www.w3.org/TR/rif-overview/
14. Harris, Steve und Andy Seaborne (Hrsg.). 2013. *SPARQL 1.1 Query Language*. W3C Recommendation 21 March 2013, http://www.w3.org/TR/sparql11-query/
15. Bizer, Christian, Tom Heath, und Tim Berners-Lee. 2009. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* 5(3): 1–22. doi:10.4018/jswis. 2009081901 5, 29
16. Miller, A.G. 1995. Wordnet: A lexical database for English. *Communications of the ACM* 38(11): 39–41
17. van Assem, Mark, R. Maarten Menken, Guus Schreiber, Jan Wielemaker, und Bob Wielinga. 2004. A Method for Converting Thesauri to RDF/OWL. In *Proc. of Int. Semantic Web Conference 2004 (ISWC 2004)* Lecture Notes in Computer Science, Bd. 3298, 17–31
18. Kobilarov, Georgi, Tom Scott, Yves Raimond, Oliver Silver, Chris Sizemore, Michael Smethurst, Christian Bizer, und Robert Lee. 2009. Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. In *Proc. of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications (ESWC 2009)*, 723–737. Berlin, Heidelberg: Springer-Verlag
19. Hausenblas, Michael. 2009. *Exploiting Linked Data to Build Web Applications, Internet Computing*, IEEE, vol.13, no.4, pp.68,73, July–Aug. 2009, doi: 10.1109/MIC.2009.79
20. Cyganiak, Richard, Holger Stenzhorn, Renaud Delbru, Stefan Decker, und Giovanni Tummarello. 2008. Semantic sitemaps: efficient and flexible access to datasets on the semantic web. In *Proc. of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications (ESWC 2008)*, 690–704. Berlin, Heidelberg: Springer-Verlag
21. Cyganiak, R., H. Stenzhorn, R. Delbru, S. Decker, und G. Tummarello. 2008. Semantic Sitemaps: Ecient and exible access to datasets on the Semantic Web. In *Proceedings of the 5th European Semantic Web Conference*, Bd. 5021, 690–704
22. Oren, E., R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, und G. Tummarello. 2008. Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies* 3(1): 37–52

23. Tummarello, Giovanni, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, und Stefan Decker. 1301. Sig.ma: live views on the web of data. In *Proc. of the 19th international conference on World Wide Web (WWW '2010)*. New York, NY, USA: ACM

24. Hausenblas, Michael 2009. *Linked Data Applications – The Genesis and the Challenges of Using Linked Data on the Web, DERI Technical Report 2009-07-26*. http://linkeddata.deri.ie/sites/linkeddata.deri.ie/files/lod-app-tr-2009-07-26_0.pdf

25. Villazón-Terrazas, Boris et al. 2011. Methodological Guidelines for Publishing Government Linked Data. In *Methodological Guidelines for Publishing Government Linked Data*, 27–49. Springer

26. Maali, Fadi, und John Erickson. 2014. *Data Catalog Vocabulary (DCAT), W3C Recommendation*. http://www.w3.org/TR/vocab-dcat/

27. Reynolds, Dave (Hrsg.). 2014 *The Organization Ontology*. W3C Recommendation 16 January 2014, http://www.w3.org/TR/vocab-org/

28. Cyganiak, Richard und Dave Reynolds (Hrsg.). 2014. *The RDF Data Cube Vocabulary*. W3C Recommendation 16 January 2014, http://www.w3.org/TR/vocab-data-cube/

29. Miles, Alistair, und Sean Bechhofer. 2009. *SKOS Simple Knowledge Organization System Reference. W3C Rcommendation*. http://www.w3.org/TR/skos-reference

30. Alexander, Keith, Richard Cyganiak, Michael Hausenblas, und Jun Zhao. 2011. *Describing Linked Datasets with the VoID Vocabulary, W3C Interest Group Note*. http://www.w3.org/TR/void/

31. Miller, Paul, Rob Styles, und Tom Heath. 2008. *Open Data Commons, a License for Open Data* Proc of LDOW2008, April 22

32. Volz, Julius, Christian Bizer, Martin Gaedke, und Georgi Kobilarov. 2009. *Silk – A Link Discovery Framework for the Web of Data* 2nd Workshop about Linked Data on the Web (LDOW2009), Madrid, Spain, April

33. Ngonga Ngomo, Axel-Cyrille, und Sören Auer. 2011. *LIMES – A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data* Proc. of IJCAI

34. Krötzsch, Markus, František Simančík, und Ian Horrocks. 2012. A Description Logic Primer. *CoRR* abs/1201.4089. arxiv.org 2012

Die Bewirtschaftung vernetzter Daten auf Basis von Linked Data Technologien

3

Tassilo Pellegrini

Zusammenfassung

Der Beitrag diskutiert vor allem rechtliche Aspekte der Bewirtschaftung von vernetzten Daten entlang der Content Value Chain. Dies umfasst zum einen die Integration und Verwendung externer Daten im Zuge der Content-Verarbeitung, zum anderen die Wahl des richtigen Lizenzmodells für die Veröffentlichung eigener Daten als Linked Open Data. Ausgehend von unterschiedlichen Asset-Typen, die bei der Generierung von Linked Data anfallen, zeigt der Beitrag welche Asset-Typen durch welches Rechtsinstrument geschützt werden können. Ein besonderes Augenmerk liegt auf der Kombination offener und geschlossener Lizenzinstrumente zu Zwecken der Diversifikation von Geschäftsmodellen.

3.1 Einführung

Mit der zunehmenden Interoperabilität von IT-Systemen und -Plattformen sowie der damit verbundenen Portabilität von Daten rücken neben Content auch technische Artefakte wie Daten, Metadaten, Wissensmodelle und korrespondierende Services ins Vermarktungsportfolio von Unternehmen und öffentlichen Organisationen. Prominente Entwicklungen wie etwa Open (Government) Data, die Renaissance des Data Journalism und der Trend in Richtung Service-Orientierung [14] sind Indizien für einen zunehmenden Wertschöpfungsbeitrag der Datenbewirtschaftung. Dies zeigt sich einerseits in der wachsenden Be-

T. Pellegrini ✉
Institut für Medienwirtschaft, Fachhochschule St. Pölten, Matthias Corvinus Str. 15, A-3100 St. Pölten, Österreich
e-mail: tassilo.pellegrini@fhstp.ac.at

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_3

deutung von Metadaten in der digitalen Wertschöpfung – speziell im Kontext der stark auf Datenanalytik, Automatismen und Co-Produktion basierenden Service-orientierten Logik des eCommerce [10, 9, 19] –, andererseits in der fortwährenden Entwicklung neuer Standards und Methoden zur Bewirtschaftung von Daten unter netzökonomischen Bedingungen, wie etwa die Semantic Web Initiative des World Wide Web Konsortiums[1] oder die Metadateninitiative schema.org[2] belegen. Begriffe wie „Big Data" als konzeptioneller Überbau und „Linked Data" als technologisch-methodische Basis der zunehmenden Bewirtschaftung können hierbei als Ausdruck einer disruptiven Innovation gesehen werden, in der neben dem Beherrschen der Technologie insbesondere die Frage nach der ökonomischen Verwertung der neu geschaffenen, werttragenden Artefakte in der digitalen Wertschöpfung eine zentrale Rolle spielen.

Vor diesem Hintergrund behandelt der Beitrag folgende Fragestellungen:

1. Was sind die technischen Enabler und institutionellen (immaterialgüterrechtlichen) Rahmenbedingungen der Datenbewirtschaftung unter netzökonomischen Bedingungen?
2. Welche Auswirkungen hat dies auf die Lizenzierungspraxis von Unternehmen und öffentlichen Organisationen in Bezug auf die Definition von maschinenlesbaren Licensing Policies?
3. Was ist der aktuelle Status der Linked Data Lizenzierung und welcher Handlungsbedarf erwächst daraus?

Der Beitrag gliedert sich folgendermaßen: Abschnitt 3.2 diskutiert den Innovationsgehalt von Linked Data aus der Perspektive des Enterprise Data Management unter besonderer Berücksichtigung der semantischen Interoperabilität. Abschnitt 3.3 erläutert den Wertschöpfungsbeitrag von Linked Data entlang der Content Value Chain und seine Bedeutung für Service-orientierte Unternehmen. Abschnitt 3.4 diskutiert ausgehend von unterschiedlichen Linked Data Assets die immaterialgüterrechtlichen Aspekte des Einsatzes von Linked Data Technologien. Abschnitt 3.5 geht auf Linked Data Licensing Policies ein und erläutert die Rolle von Rechteauszeichnungssprachen für die Verwertung von Linked Data unter netzökonomischen Bedingungen. Abschnitt 3.6 wirft einen Blick auf die Lizenzierungspraxis von Linked Data und identifiziert aktuelle Problemlagen. Abschnitt 3.7 fasst die Ergebnisse zusammen und gibt einen Ausblick auf weitere Entwicklungen.

[1] Siehe http://www.w3.org/standards/semanticweb/, aufgerufen am 10.03.2014.

[2] Bei dieser Initiative handelt es sich um ein Gemeinschaftsprojekt von Google, Bing, Yahoo und Yandex für die Bereitstellung normierter Metadatenschemata und Markups zur Annotation von Webseiten. Siehe https://schema.org/, aufgerufen am 10.03.2014.

3.2 Linked Data als technikinduzierte Innovation im Enterprise Data Management

3.2.1 Metadaten als Innovationsfeld

Innovationen im Bereich der Metadaten können als Reaktion auf veränderte Umweltbedingen in der Content-Wertschöpfung verstanden werden, deren Ursache wiederum in technologischen und methodischen Innovationen zu finden sind. Eine der radikalen Entwicklungen, die die Medienwertschöpfung in den vergangenen zwei Jahrzehnten massiv beeinflusst hat, war die massenhafte Adaption des World Wide Webs (im Weiteren kurz Web genannt) als multimediale Produktions-, Distributions- und Konsumptionsplattform. Das Web kann hierbei als prototypische Manifestation einer technikinduzierten Medientransformation begriffen werden, die, wie sich mittlerweile vielfach feststellen lässt, nicht nur zu einem intensiven und branchenübergreifenden Substitutionswettbewerb geführt hat, sondern auch das Informationsverhalten im Arbeitsalltag stark verändert hat [18, 5]. Dies spiegelt sich unter anderem in Fragestellungen wider, wie unter diesen neuen Umweltbedingungen Unternehmen und öffentliche Organisationen Information aufbereiten und verfügbar machen sollen, um weiterhin wettbewerbsfähig bleiben zu können. Der daraus entstehende Innovationsdruck führt laut Haase [9] zu einem Metadata-Shift, der sich auf die zentrale Aussage reduzieren lässt: „Mit wachsender Informationsmenge steigt die ökonomische Relevanz wohlstrukturierter Metadaten."

Mittels einer quantitativen Inhaltsanalyse der Library, Information Science and Technology Abstracts (LISTA) Datenbank haben Saumure & Shiri [25] diesen Trend auch empirisch belegt (Tab. 3.1).

Wie aus dieser Gegenüberstellung abgelesen werden kann, haben sich die Forschungsschwerpunkte in der Informationswissenschaft seit dem Jahr 1993 nicht nur verlagert sondern auch ausdifferenziert. Lagen in der Prä-Web Ära die Foki auf Fragestellungen der Indizierung und Simulation kognitiver Modelle (was als Reaktion auf die damals noch dominante Tradition der künstlichen Intelligenz interpretiert werden kann), so liegen die Schwerpunkte in der Post-Web Ära auf Fragen der metadatenbasierten Applikationsentwicklung, der Katalogisierung und Klassifikation (insbesondere von Web-Content), der Interoperabilität und der maschinellen Unterstützung der Wissensorganisation.

Bis auf diesen letzten Aspekt gibt es kaum Überschneidungen zwischen den alten und neuen Forschungsthemen, was als Indiz für die sich wandelnde ökonomische Rolle der Metadaten und ihrer Bewirtschaftung interpretiert werden kann. Gleichzeitig zeigt die konstante Forschung im Bereich der maschinellen Informationsverarbeitung und der Klassifikation, dass es sich hierbei um einen Kristallisationspunkt für eine sich intensivierende Metadatenbewirtschaftung handeln könnte, wie sich mit Marktstudien zu Trends im Enterprise Information Management [6, 15, 16] gut belegen lässt.

Tab. 3.1 Forschungsfelder der Wissensorganisation im Prä- und Post-Web-Zeitalter. (Quelle: Saumure & Shiri 2008)

Forschungsfeld	Prä-Web	Post-Web
Metadata Applications & Uses	–	16 %
Cataloging & Classification	14 %	15 %
Classifying Web Information	–	14 %
Interoperability	–	13 %
Machine Assisted Knowledge Organization	14 %	12 %
Education	7 %	7 %
Digital Preservation & Libraries	–	7 %
Thesauri Initiatives	7 %	5 %
Indexing & Abstracting	29 %	4 %
Organizing Corporate or Business Information	–	4 %
Librarians as Knowledge Organizers of the Web	–	2 %
Cognitive Models	29 %	1 %

3.2.2 Semantische Interoperabilität als Kerninnovation von Linked Data

Konventionelle Datenbereitstellungsstrategien in Form von (semi-)strukturierten Dokumenten (z. B. HTML, CSV-Dateien) oder proprietären APIs werden nur bedingt den Ansprüchen hoch vernetzter und dynamischer Daten-Ökosysteme gerecht. Mit jeder zusätzlichen Quelle steigen die Integrationsaufwände exponentiell, Veränderungen in der Datenbankstruktur gehen oftmals zu Lasten der Systemintegrität und Aktualisierungen der Datenbasis sind meist nur unter hohen Aufwänden in Echtzeit verfügbar. Hier setzt der Linked Data Ansatz an, der eine höchstmögliche Interoperabilität zwischen verteilten Datenquellen anstrebt und so die kosteneffiziente und zeitkritische Integrierbarkeit, eindeutige Interpretierbarkeit und Wiederverwendbarkeit von dispersen Daten ermöglicht. Linked Data bedient sich sogenannter Semantic Web Standards[3] um existierende Datenbestände hoch strukturiert aufzubereiten und plattformunabhängig für die Integration und Syndizierung bereitzustellen. Hierbei werden Daten mittels des normierten Datenmodells RDF (Resource Description Framework)[4] strukturiert und verfügbar gemacht. Die semantisch angereicherten Daten werden im konventionellen Sinne nicht relational sondern als Graph repräsentiert. Sowohl die Knoten als auch die Kanten des Graphen sind über URIs (Uniform Resource Identifiers)[5] eindeutig identifizierbar und referenzierbar. Dieser semantische RDF-Graph kann mittels der normierten Abfragesprache SPARQL

[3] Ein Gesamtüberblick der relevanten Standards findet sich unter http://www.w3.org/standards/semanticweb/, aufgerufen am 20.12.2013.
[4] Siehe http://www.w3.org/RDF/, aufgerufen am 10.12.2013.
[5] Siehe http://www.w3.org/wiki/URI, aufgerufen am 10.12.2013. Siehe auch Berners-Lee [3].

(SPARQL Query Language for RDF)[6] feingranular und in hoher semantischer Tiefe abgefragt werden. Dies erlaubt die leichte Formulierung expressiver Datenbankabfragen, die mit konventionellen Mitteln entweder gar nicht oder nur mit hohen Aufwänden machbar wären.

Die gehobene semantische Interoperabilität erlaubt die kosteneffiziente Zusammenführung verteilt vorliegender Datensets, die Entwicklung von service-orientierten Produkten und ermöglicht eine Bewirtschaftung des digitalen Contents entlang der gesamten Wertschöpfungskette [6, 16].[7]

Tim Berners-Lee, Direktor des World Wide Web Konsortiums, fasst die technologischen Prinzipien von Linked Data folgendermaßen zusammen [4]:

1. Nutze eindeutige Identifikatoren (Uniform Resource Identifiers – URIs) als Name für Dinge.
2. Nutze http-URIs um diese Dinge im World Wide Web auffindbar zu machen.
3. Nutze den RDF-Standard zum Annotieren der URIs mit sinnvoller Kontextinformation.
4. Verknüpfe URIs mit anderen URIs um weitere Informationen auffindbar zu machen.

Das erste Prinzip stellt eine Grundbedingung dar und besagt, dass Ressourcen über einen Uniform Resource Identifier (URI) entsprechend der IETF URI Konventionen [3] ausgezeichnet werden müssen.

Das zweite Prinzip besagt, dass http-URIs als Bezeichnungen für Ressourcen verwendet werden sollen. Denn wie Berners-Lee anmerkt, wird oftmals übersehen, dass http-URIs im eigentlichen Sinne keine Adressen sondern „Namen“ darstellen, auf deren Protokoll eine mächtige und evolvierende Infrastruktur in Form von Schreib- und Leseautomatismen (sog. REST-Services) aufbaut.

Das dritte Prinzip besagt, dass die maschinelle Verarbeitung verfügbarer Datenquellen, die bereits die URI-Konventionen erfüllen, durch Anreicherung mit interoperablen Metadaten verbessert wird. Strukturierte Annotation auf Grundlage von Wissensmodellen bzw. Ontologien[8] setzt hier an. Die Repräsentation der hierbei verwendeten Vokabulare muss der RDF-Norm genügen.

Als viertes und letztes Prinzip sollen die verfügbaren URIs durch gegenseitige Verweise de-referenziert und dadurch vernetzt werden, so wie es im konventionellen „Web of Documents“ auf Basis des Hypertext-Prinzips erfolgt. Daten werden durch Interoperabilität zu Netzwerkgütern und steigern ihren Wert mit dem Grad ihrer Konnektivität

[6] Siehe http://www.w3.org/TR/rdf-sparql-query/, aufgerufen am 10.12.2013.

[7] Für eine differenzierte Diskussion des volkswirtschaftlichen und betriebswirtschaftlichen Wertschöpfungsbeitrages von Big Data im Allgemeinen und Linked Data im Speziellen siehe [20]. Eine Schematisierung der Linked Data Value Chain findet sich bei Latif et al. [13].

[8] Saumure & Shiri definieren Ontologien folgendermaßen: „Ontologies are being considered valuable to classifying web information in that they aid in enhancing interoperability – bringing together resources from multiple sources.“ [25, S. 657].

und Referenzierbarkeit [26] – eine ökonomische Gesetzmäßigkeit, die für den Produktionsfaktor Metadaten bisher unterbelichtet ist. Dieser Aspekt stellt auch den wichtigsten Unterschied zur konventionellen Datenbewirtschaftung dar, wo aufgrund proprietärer Repräsentationsstandards und Schemata kaum Netzeffekte zu erzielen sind.

Im Kern der oben beschriebenen Entwicklungen steht die technische Herstellung von semantischer Interoperabilität zwischen Datenbanken, Repositorien und anderen werthaltigen Informationsquellen. Die Vorteile von Linked Data gegenüber konventionellen Integrationstechnologien lassen sich laut Auer [1] folgendermaßen beschreiben:

De-Referenzierbarkeit: Die Verwendung von URIs erlaubt nicht nur Dinge im Web eindeutig zu identifizieren, sondern auch diese inklusive der angereicherten Zusatzinformation abzurufen.

Kohärenz: Die Verwendung von RDF als universelles Datenmodell erlaubt die kohärente Vernetzung von Informationen aus unterschiedlichen Namensräumen und ermöglicht auf diese Weise die semantische Anreicherung von Information durch sogenannte typisierte Links.

Integrierbarkeit: Das normierte RDF-Datenmodell erlaubt – aus technischer Perspektive – die niedrigschwellige Integration von syntaktischen und semantischen Informationen aus den vernetzten, dispers vorliegenden Datenquellen. Mittels Schema-Mapping (z. B. von RDF Vokabularien) und Instance Matching können in Folge semantisch hoch expressive Informationsbestände aggregiert und abgefragt werden.

Aktualität: Die Datenquellen inklusive ihrer Netzstruktur können aufgrund des geteilten Datenmodells leicht aktualisiert und veröffentlicht werden, ohne daraus resultierender zusätzlicher Integrationsaufwände oder Performanceverluste, wie sie üblicherweise bei konventionellen Extraktionsmaßnahmen oder Datentransformationen entstehen. Dies garantiert unter anderem eine hohe Aktualität der Daten und darauf aufbauender Dienste.

Als konkrete Manifestation des Linked Data Paradigmas lässt sich die seit 2007 stetig wachsende „Linked Data Cloud“[9], eine dezentrale und kollaborativ gewachsene Infrastruktur aus RDF-Daten, anführen. Diese Data Cloud umfasste mit Stand 2013 mehrere hundert Milliarden Fakten aus unterschiedlichsten Themenfeldern und mittlerweile hunderten Datenquellen.[10] Diese Daten sind vorwiegend offen lizenziert und werden bereits aktiv kommerziell genutzt. So veröffentlichen Unternehmen und öffentliche Organisationen ihre Datensets in der Linked Data Cloud und nutzen gleichzeitig deren Daten um hauseigene Datenbestände anzureichern und Rich Content Anwendungen darauf aufzusetzen. Insbesondere Unternehmen aus der Pharma-Industrie (z. B. Roche, Merck, Elly Lilly) und der Medienbranche (z. B. BBC, NY Times, Reuters, Reed Elsevier, Wolters Kluwer, Pearson Publishing, Springer Verlag, ACM, Agence France Press, Google, Facebook) haben sich als Early Adopter von Linked Data Technologien hervorgetan.[11]

[9] Siehe http://linkeddata.org/, aufgerufen am 26.12.2013.

[10] Einen Überblick über verfügbare Datenquellen bietet z. B. http://datahub.io, aufgerufen am 31.12.2013.

[11] Vertiefende Fallbesprechungen siehe z. B. Rayfield [24] für die BBC oder Dodds & Davis [7] für guardian.co.uk.

3.3 Der Wertschöpfungsbeitrag von Linked Data in Service-orientierten Unternehmen

Mit der voranschreitenden Digitalisierung von Geschäftsprozessen agieren viele Unternehmen zunehmend medienartig – d. h. die Bewirtschaftung und Verwertung von Information rücken in den Kern der Geschäftstätigkeit. Dies äußert sich etwa in der zunehmenden Service-Orientierung von Marketingaktivitäten, deren Grundlage oftmals Informationsprodukte mit Service-Charakter – wie etwa personalisierte Produktportfolios, Recommender-Services oder Programmierschnittstellen samt Service-Levels – sind. Als Ursache dieser Entwicklung lassen sich zwei Bedingungen identifizieren, die auch die Adaption von Linked Data Prinzipien begünstigen: zum einen verfügen service-orientierte Unternehmen über ein gehobenes Bewusstsein und Kompetenzen bezüglich der technischen und wirtschaftlichen Bedeutung von wohlstrukturierten Daten für die Wertschöpfung; zum anderen basieren Diversifikationsstrategien in der Service-Orientierung vielfach auf Konzepten der Zweit- und Drittverwertung bestehender Assets – idealerweise mittels Automatisierung. Dies wiederum setzt das Vorhandensein entsprechender, maschinell verarbeitbarer Inhalte voraus, deren Grundlage wiederum prozessunterstützende deskriptive oder strukturelle Metadaten sind.

Vor diesem Hintergrund scheint es nicht verwunderlich, dass ein Zusammenhang zwischen dem Trend zur Service-Orientierung und Linked Data existiert, zumal die zugrunde liegenden technologischen und methodischen Konzepte sowohl traditionelle Geschäftsstrategien unterstützen als auch Optionen für Diversifikation eröffnen. Abbildung 3.1 illustriert diesen Zusammenhang am Beispiel des Einsatzes von Linked Data entlang der Content-Wertschöpfung.

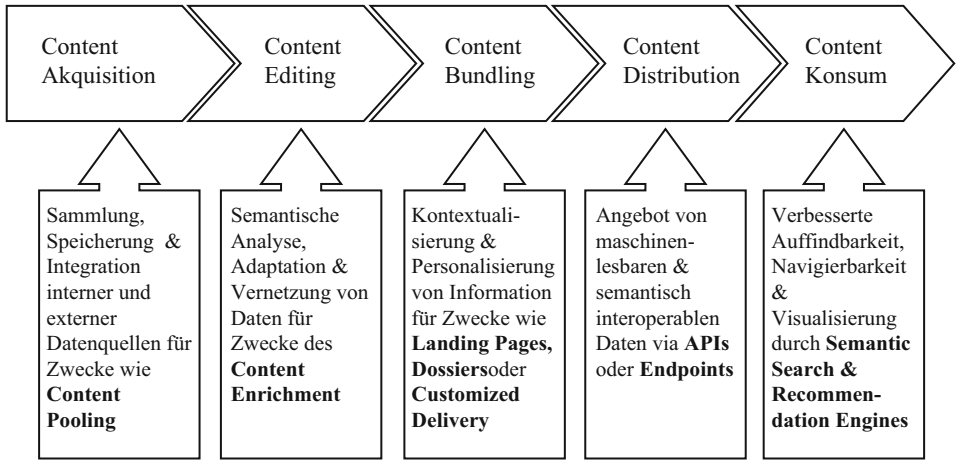


Abb. 3.1 Linked Data in der Content-Value-Chain

1. Die Content-Akquise umfasst alle Aktivitäten der Sammlung, Speicherung und Integration von Daten. Im Zuge dieses Prozesses werden Fakten und Information von internen und externen Quellen für die weitere Verarbeitung in Form semantischer Indices gepoolt.
2. Die Stufe des Content-Editings umfasst die semantische Analyse, Adaption, Verlinkung und Qualitätssicherung von Daten für Zwecke des Content Enrichments.
3. Das Content Bundling beschäftigt sich mit der Kontextualisierung und Personalisierung von Information. Es dient dem maßgeschneiderten, thematisch kohärenten Zugang zu Content-Einheiten z. B. in Form von Landing Pages oder Dossiers.
4. Im Zuge der Content Distribution werden maschinen-lesbare und semantisch interoperable Daten z. B. via Programmierschnittstellen oder SPARQL-Endpoints für den internen und/oder externen Gebrauch technisch zugänglich gemacht.
5. Die Ebene des Content Konsums umfasst alle Maßnahmen der adäquaten Präsentation und Zugänglichmachung von Content in Form von semantischer Suche, Empfehlungs- und Filterdiensten mit dem Ziel die Auffindbarkeit, Navigierbarkeit und Wiederverwendbarkeit zu erhöhen.

Die Betrachtung der Mehrwertpotenziale entlang der Content-Wertschöpfung illustriert, dass Linked Data Technologien sowohl der inkrementellen Weiterentwicklung und Verbesserung bestehender Dienste, als auch der Erschließung neuer Produkt- und Dienstleistungsmärkte, die sich im Zuge der voranschreitenden Verfügbarkeit vernetzter Daten auftun, unterstützen.

Aus der Perspektive der *Produkt- und Service-Diversifikation* schließt dies etwa die Erweiterung bestehender Recherchetools um konzeptbasierte Ansätze wie moderierte Suche, Recommender-Systeme oder andere Filtersysteme ein. Weitere Optionen ergeben sich aus der meist vollautomatischen Granularisierung von Dokumenten in Content-Einheiten, wodurch gezielt Personen, Orte, Produkte, Preise, Aussagen u. v. m. extrahiert und in weiterer Folge kosteneffizient zu personalisierten Dossiers verdichtet werden können. Dies stellt insbesondere in der Verwertung aktualitätskritischer Information einen wichtigen Aspekt dar und spiegelt sich auch im aktuellen Trend des „Roboter-Journalismus“ wieder, wo redaktioneller Content auf Basis hochstrukturierter Informationsbasen de-facto vollautomatisch generiert wird.[12]

Aus der Perspektive der *Marktdiversifikation* führt der Einsatz von Linked Data Technologien unter dem Stichwort *Dynamic Semantic Publishing (DSP)* sowohl zu einer Verlängerung als auch zu einer Verdichtung bestehender Wertschöpfungsketten. Die Verlängerung der Wertschöpfung ergibt sich aus der Möglichkeit, aufgrund der geringen Integrationskosten externer Ressourcen Nischenmärkte kosteneffizienter als bisher zu bewirtschaften. So wird die semantische Anreicherung von Content im Sinne von Verschlagwortung und Strukturierung nicht ex post durch geschulte Spezialisten vollzogen, sondern

[12] Einen weiterführenden Einblick in die wirtschaftlichen und sozialen Implikationen der zunehmenden Automatisierung kreativer Tätigkeiten durch Maschinen auf Basis von Big Data bieten Frey & Osborne [8].

findet bereits on-the-fly im Entstehungsprozess des jeweiligen Content-Produktes statt und wird auf der jeweiligen Wertschöpfungsstufe bloß um bereichsspezifische Aspekte ergänzt. Die Verdichtung der Wertschöpfung rührt daher, dass die Grenzen zwischen den wertschöpfenden Stufen und Akteuren durchlässiger und schwerer abgrenzbar werden, was sich etwa in der voranschreitenden Integration von Produktdaten mit Geschäftsprozessdaten äußert. In Summe bedeutet dies: die Grenzen zwischen B2B- und B2C-Märkten verschwimmen und schaffen neue Potentiale der Markterschließung und Verwertung.

Auf der Ebene der *Verwertungsmodelle* muss Linked Data aus der Perspektive der Asset-Creation diskutiert werden. Als grundlegendes Prinzip gilt, dass auf Basis von Semantic-Web-Standards Metadaten zu Netzgütern werden und sich entsprechend der Funktionslogik von Netzen bewirtschaften lassen [25]. Während Metadaten bisher aufgrund ihrer oftmals proprietären Strukturen und Repräsentationsstandards nicht dazu geeignet waren, auf Basis netzökonomischer Prinzipien bewirtschaftet zu werden, ändert sich dies mit Linked Data radikal. Asset Creation durch Linked Data bedeutet, die bestehenden Metadaten-Ressourcen wie z. B. Schemata, Vokabularien, Datenbank-Modelle, Mapping-Files, Indices, Entitäten-Sammlungen, Content-Snippets etc. zu identifizieren und mit Rechten bzw. Lizenzen zu versehen, welche die skalenökonomischen Effekte von Netzgütern stützen. Dazu ist es notwendig eine Linked Data Licensing Policy zu entwickeln, die sowohl den urheber- als auch datenbankrechtlichen Aspekten gerecht wird.[13] Hierbei ist zu berücksichtigen, dass die traditionell vorherrschende Philosophie starker Eigentumsrechte mit dem viralen, selbstorganisierenden und dezentralen Charakter des Web nur eingeschränkt kompatibel ist. Entsprechend sollte eine Strategie der Rechtediversifikation entwickelt werden, die im Sinne der Versionierung sowohl die Bedienung des auf Offenheit basierenden Web-Ökosystems als auch des Corporate-Marktes ermöglicht. Je nach Marktstruktur und strategischem Ziel könnte dies eine kostengünstige (bis kostenlose) Bereitstellung von Metadaten-Assets auf Basis offener Lizenzen oder eine Paid-Content-Strategie auf Basis von Service Levels und/oder geschlossenen Lizenzmodellen bedeuten, wobei im Falle personalisierter, auf Kundenbedarfe zugeschnittener Produkt- und Dienstleistungen nicht nur die Metadaten-Assets vermarktet werden können, sondern auch die Kompetenz der Metadatenbewirtschaftung. Bildlich gesprochen könnte ein serviceorientiertes Unternehmen seinen Produkt-Index im Netz veröffentlichen und anderen die Möglichkeit geben, auf dessen Basis Applikationen zu entwickeln.[14] Je nachdem, ob der Index für kommerzielle oder nicht-kommerzielle Zwecke genutzt wird, entscheidet sich die zur Anwendung kommende Lizenz, die Version des Index, die Service Levels und das Bepreisungsmodell. Hierbei besteht die Kunst aus dem vollen Spektrum proprietärer und offener Lizenzmodelle zu schöpfen und dies strategisch zur Stützung bestehender Geschäftsmodelle einzusetzen.

[13] Ein Überblick zu Linked Data Licensing findet sich bei [22].

[14] Dies wird etwa durch die Metadateninitiative http://schema.org (bestehend aus Google, Yahoo, Yandex und Bing) bzw. den Geodaten-Anbieter http://geonames.org praktiziert. Der konkrete Nutzungskontext entscheidet über das zur Anwendung kommende Geschäftsmodell.

3.4 Immaterialgüterschutz von Linked Data

3.4.1 Asset-Typen in der Bewirtschaftung vernetzter Daten

Eine differenzierte Betrachtung der technischen Prozessierung von semantischen Metadaten offenbart ein komplexes Gefüge aus Asset-Typen, die als technische Artefakte einen wertschöpfenden Beitrag in der Content-Produktion leisten und durch entsprechende Rechtsinstrumente auch geschützt werden können. Dies ergibt sich im Wesentlichen aus dem gewerblichen Rechtsschutz geistiger Schöpfungen durch Instrumente wie das Urheberrecht, das Datenbankrecht, das Patentrecht u. a.m. Im Kontext von Linked Data treten neben die klassische Vermarktungseinheit „Dokument" auch sogenannte Metadata-Assets ins Zentrum der Schutzwürdigkeit. Je nach Reichhaltigkeit und Expressivität der semantischen Aufbereitung lassen sich folgende Metadata-Assets unterscheiden (Tab. 3.2).

Aus der Prozessierung semantischer Metadaten lassen sich in Folge weitere Asset-Typen ableiten, die als „2nd Order Information" bezeichnet werden sollen. Hierbei lassen sich vier Kategorien unterscheiden: 1) *Referenzen* als Sammlung von Verweisen innerhalb und zwischen Dokumenten (z. B. als Indices); 2) *Inferenzen* als automatische Erschließung impliziter Information aus semantisch verknüpften Datensets (z. B. als Queries); 3) *Präferenzen* als gebrauchsspezifische Muster der Interaktion mit digitalen Artefakten (z. B. als anonymisierte Nutzerprofile); und 4) *Konfidenzen* als personenbezogene Profile aus Transaktionsspuren, Sozial-, Stimmungs- und Meinungsmustern. Tabelle 3.3 erläutert die unterschiedlichen Asset-Typen.

Wie Beer [2] herausstreicht, weckt insbesondere der letzte Asset-Typ in Form von 2nd Order Information Begehrlichkeiten für die Bewirtschaftung von semantischen Metadaten, zumal sich diese Daten hervorragend für Zwecke der nutzerspezifischen Informationsaggregation nutzen lassen. Die sich daraus ergebenden datenschutzrechtlichen Sachverhalte sind bis dato nur unbefriedigend gelöst und institutionalisiert [12, 17].

Tab. 3.2 Metadata Assets in der technischen Bewirtschaftung von Online-Content

Metadata Assets	Strukturelle und technische Artefakte für die Erzeugung von Linked Data
Datensatz	Strukturierte Sammlung und Aufbereitung von Rohdaten
URIs (Uniform Resource Identifiers)	Eindeutige Identifikatoren als Bezeichner und Adresse für Entitäten eines Datensatzes
Namespaces	Eindeutige Namensräume zur Dereferenzierung von URIs
Vokabulare	Eindeutige domänen- und funktionsspezifische Begrifflichkeiten zur Annotation für deskriptive, strukturelle oder administrative Zwecke
Schemata	Formales Modell zur Strukturierung von Daten durch Selektion, Kombination und Mapping von Vokabularen
Ontologien	Formale Modelle um Beziehungen zwischen und Eigenschaften von Metadaten abzubilden
Regeln	Logische Operationen zur automatischen Erschließung von Information aus Ontologien

Tab. 3.3 2nd Order Information in der Prozessierung semantischer Metadaten

2nd Order Information	Information, die aus der Prozessierung semantischer Metadaten entsteht
Referenzen	Aggregation kontextrelevanter Ressourcen in Form von semantischen Indices, diese können sowohl Dokumente als auch Instanzdaten enthalten.
Inferenzen	Queries zur logik-basierten Erschließung impliziter Information im semantischen Graphen
Präferenzen	Gebrauchssensitive Empfehlung und Filterung von Ressourcen auf Basis konstitutiver, regulativer und generativer Regeln (Beer 2009, S. 994)
Konfidenzen	Beobachtung und Analyse nutzerbezogener, bewusster und unbewusster Transaktionsspuren, Interessen und Stimmungsmuster

Tab. 3.4 Rechtsschutz von Linked Data (in Anlehnung an Sonntag [27])

	Urheberrecht	Datenbank-Recht	Wettbewerbsrecht
Dokument	Ja	Ja	Ja
Datensatz	Nein	Ja	Ja
Identifikatoren	Nein	Ja	Nein
Namensräume	Ja	Nein	Ja
Vokabulare	Teilw.	Ja	Ja
Schemata	Teilw.	Ja	Ja
Ontologie	Teilw.	Ja	Ja
Regeln	Teilw.	Ja	Ja
Inferenzen	Ja	Ja	Ja
Referenzen	Ja	Ja	Ja
Präferenzen	Ja	Ja	Ja
Konfidenzen	Ja	Ja	Ja

3.4.2 Rechtsschutz von Linked Data

Die Lizensierungsfrage von Linked Data ist nicht trivial, zumal unterschiedliche Bestandteile eines semantischen Ordnungssystems mit unterschiedlichen Rechtsinstrumenten geschützt werden können. Zur Anwendung kommen in der folgenden Übersicht das Urheberrecht, das Datenbankrecht und das Recht gegen unlauteren Wettbewerb.[15] In Anlehnung an Sonntag (2006) lassen sich folgende Schutzobjekte unterscheiden (Tab. 3.4).

Die Tabelle offenbart ein dicht gewobenes Schutzregime. Während das Urheberrecht den kreativen Werkcharakter schützt, stellt das Datenbankrecht einen Leistungs- bzw. Investitionsschutz dar. Fragestellungen zum missbräuchlichen Gebrauch geschützter Assets

[15] Das Patentrecht wird an dieser Stelle ausgespart, da – zumindest nach europäischer Rechtsprechung – die genannten Assets nur indirekten Schutz in Kombination mit einer technischen Erfindung erlangen können. Ebenfalls nicht Bestandteil der Betrachtung ist das Markenrecht und das Domainrecht.

werden u. a. im Wettbewerbsrecht behandelt. Diese drei Rechtsbereiche spielen in Folge auch die wichtigste Rolle in der Spezifizierung von Licensing Policies für Linked Data und darauf aufbauender Verwertungsmodelle.

3.4.3 Rechtsschutz unter netzökonomischen Bedingungen

Während Metadaten bisher aufgrund ihrer oftmals proprietären Strukturen und Repräsentationsstandards nicht dazu geeignet waren, auf Basis netzökonomischer Prinzipien bewirtschaftet zu werden, ändert sich dies mit Linked Data grundlegend. Asset Creation durch Linked Data bedeutet, die werthaltigen Artefakte differenziert mit Lizenzen zu versehen, welche die skalenökonomischen Effekte von Netzgütern stützen bzw. beschränken. Deshalb kommen vermehrt Commons-basierte bzw. offene Lizenzmodelle – oft in Kombination mit geschlossenen Lizenzmodellen in Form eines Dual Licensings – zum Einsatz.[16]

Im Bereich des Urheberrechts hat sich mit Creative Commons[17] eine tragfähige Alternative für den Schutz von Werken etabliert. So ermöglicht es die CC0-Lizenz, auf alle Urheber- und Urheberpersönlichkeitsrechte sowie all ihre verwandten Schutzrechte an dem betreffenden Werk zu verzichten und auf diesem Weg Nutzungsrechte an den Daten der Öffentlichkeit zu übertragen. Ergänzend steht ein Lizenzbaukasten zur Verfügung, der die feingranulare Definition von Nutzungsrechten mit unterschiedlichen Freiheitsgraden auch für kommerzielle Zwecke zulässt.

Im Bereich des Datenbankrechts arbeiten unterschiedliche Initiativen parallel zu Creative Commons an sogenannten Data Commons[18], einem Set von Lizenzen, das für die Spezifika der Datenbanklizenzierung optimiert ist. Mit aktuellem Stand werden zusätzlich zur GNU Documentation License drei Lizenzmodelle angeboten: Die Lizenz *ODBL* (Open Data Commons Open Database License) bringt vergleichbar der CC0-Lizenz einen völligen Verzicht auf alle Nutzungsrechte mit sich. Die Lizenz *Open Data Commons Attribution License* verlangt nach einer Nennung des Urhebers. Die Lizenz *PDDL* (Open Data Commons Public Domain Dedication and License) erlaubt die offene Definition von Nutzungsrestriktionen.

[16] Ein Blick auf die Lizensierungspraxis der BBC zeigt, dass durch den kombinierten Einsatz offener und geschlossener Lizenzmodelle bestimmte Datenbestände der Öffentlichkeit für die Weiterverwendung zur Verfügung gestellt werden. So bediente sich die BBC (mit Stand Februar 2013) neben dem klassischen Urheberrecht folgender Lizenzmodelle: GNU Free Documentation Licence für Content, der aus der Wikipedia bezogen wird, Creative Commons Public Domain and Attribution-NonCommercial-ShareAlike für Content, der aus der MusicBrainz Datenbank bezogen wird, und Attribution-NonCommercial-ShareAlike 3.0 Unported für die Besprechungen der Musikalben der BBC. Zusätzlich wird die Verwendung der Datenschnittstellen über Geschäftsbedingungen geregelt, die eine uneingeschränkte, nichtkommerzielle Nutzung der BBC Music Beta-Daten erlauben. Siehe http://backstage.bbc.co.uk/archives/2005/01/terms_of_use.html, aufgerufen am 20.02.2013.

[17] Siehe http://creativecommons.org, aufgerufen am 05.12.2013.

[18] Siehe http://www.opendatacommons.org/, aufgerufen am 05.12.2013.

3.5 Licensing Policies und Rights Expression Languages für Linked Data

Entsprechend den diversen Rechtsschutzaspekten sollte eine Linked Data Licensing Policy aus drei Komponenten bestehen: 1) eine maschinenlesbare Lizenz, die die urheberrechtlichen Aspekte abdeckt; 2) eine maschinenlesbare Lizenz, die die datenbankrechtlichen Aspekte abdeckt; und 3) eine Community Norm, welche die verwendeten Lizenzen und Nutzungsrechte für den Human User leicht verständlich aufbereitet und im Sinne des Gesetzes gegen unlauteren Wettbewerb transparente Nutzungsbedingungen und „Good Conduct“ definiert.

3.5.1 Rechteauszeichnungssprachen

Zur maschinellen Auszeichnung von Licensing Policies wurden seit den 1990er Jahren sogenannte Rights Expression Languages (RELs) entwickelt, die dem Bereich der Digital Rights Management Technologien zuzurechnen sind [23]. RELs unterstützen die Identifikation, Filterung, Syndizierung und Modifikation von Content, der sich aus mehreren Quellen unterschiedlicher Rechteinhaber speist, und sie bilden die Grundlage für eine differenzierte automatische Prozessierung und Verwertung von Content. RELs sind damit eine zentrale technologische Komponente in hoch automatisierten und vernetzten Verwertungsstrukturen.

3.5.2 Open Digital Rights Language (ODRL)

Seit dem Jahr 2011 entwickelt die ODRL Arbeitsgruppe der W3C Community and Business Group[19] ein hoch expressives RDF/XML Vokabular zur Auszeichnung von Policies für die automatisierte Interaktion mit Online Content. ODRL baut auf einem Entity-Attribute-Modell auf, das eine feingranulare, maschinenlesbare Definition von Nutzungsrechten für digitale Assets erlaubt. Die Version 2.0 enthält 50 Ausprägungen in Bezug auf Rechte und Pflichten, 27 Verbotstypen und 10 Operatoren. ODRL eignet sich aufgrund seiner hohen Expressivität ideal zur Versionierung eines Datensatzes entlang unterschiedlicher Assets, Nutzertypen und Szenarien.

Die hohe Expressivität und damit verbundene Implementierungskomplexität des ODRL-Vokabulars hemmte bisher die Adaption des Standards für kommerzielle Zwecke. Im Jahr 2013 begann deshalb das International Press and Telecommunications Council (IPTC) unter der Bezeichnung RightsML[20] an einer leichtgewichtigen Adaption von ODRL für Zwecke der Lizenzierung von News-Content zu arbeiten.[21]

[19] Siehe http://www.w3.org/community/odrl/, aufgerufen am 02.01.2014.
[20] Siehe http://dev.iptc.org/RightsML, aufgerufen am 02.01.2014.
[21] Ein Überblick über existierende Use Cases aus der Nachrichtenbranche findet sich unter http://dev.iptc.org/RightsML-Use-Cases, aufgerufen am 02.01.2014.

3.5.3 Creative Commons Rights Expression Language (CCREL)

Komplementär zu ODRL hat sich die Creative Commons Rights Expression Language (CCREL)[22] für urheberrechtsrelevante Schutzaspekte etabliert. Sie ist das Ergebnis einer informellen W3C Arbeitsgruppe, die ihre RDF-Spezifikationen im Jahr 2008 veröffentlichte und seither von der Creative Commons Foundation als Standard für die maschinelle Auszeichnung von Creative Commons Lizenzen empfohlen wird.

CCREL bietet ein kondensiertes, hierarchisch strukturiertes Set an Attributen zur Definition von Nutzungsrechten mit Online-Content, das komplementär zu ODRL steht. Diese Attribute können nahtlos in das ODRL-Vokabular übernommen und mittels ODRL weiter ausdifferenziert und spezifiziert werden. Jedoch eine Kombination von ODRL mit CCREL ist nicht zwingend. Die semantische Expressivität von CCREL ist ausreichend für die simple Annotation von digitalen Assets mit CC Lizenzinformationen.

3.5.4 Open Data Commons

Da Open Data Commons bisher kein eigenes Vokabular zur Auszeichnung von Policies anbietet, ist die Einbindung von datenbankrechtlichen Aspekten in eine Licensing Policy zum aktuellen Stand nur durch eine Adaption des ODRL- bzw. CCREL- Vokabulars möglich, sofern der relevante Datensatz über eine dereferenzierbare URI verfügt. Sämtliche damit verbundenen Interaktionsszenarien lassen sich jedoch einwandfrei durch ODRL bzw. CCREL auszeichnen.

3.5.5 Community Normen

Die Community Norm stellt eine menschenlesbare Version der in den Rechteauszeichnungssprachen formalisierten Nutzungsbedingungen dar. Sie ist die dritte Komponente einer Linked Data Licensing Policy. Die Community Norm gibt entsprechend Auskunft über die Nutzungsbedingungen einer Linked Data Quelle. Damit soll die Nutzungstransparenz offener Daten erhöht und Einhaltung der Nutzungsbedingungen gewährleistet werden.

In einer Minimalvariante sollte eine Community Norm Auskunft geben über alle administrativen Aspekte eines offenen Datensatzes. Darunter fallen z. B. Informationen zu Urheber, zur Anwendung kommende Lizenz sowie Rechte und Pflichten, die mit der Nutzung verbunden sind. Weiters gilt es auch Strukturinformationen zum Datensatz zu explizieren wie z. B. Version, Aktualitätsstand, Anzahl der Entitäten und Relationen, aus denen sich der Datensatz zusammensetzt. Unter Umständen ist es auch noch angebracht im Sinne einer guten Dokumentation Empfehlungen zur Implementierung und weiterführenden Vernetzung des Datensatzes zu geben.

[22] Siehe http://www.w3.org/Submission/ccREL/, aufgerufen am 02.01.2014.

Ein Blick auf die Praxis offenbart, dass zum aktuellen Stand Community Normen sehr unterschiedlich ausfallen und in Umfang, Tiefe und Auffindbarkeit sehr stark voneinander abweichen.

So verpackt die Universität von Southampton ihre Community Norm innerhalb eines Datensatzes als Bestandteil eines sogenannten RDFs Statements in folgender Form:

```
36 rdfs:comment "This data is freely available to
use and reuse. Please provide an attribution to
University of Southampton, if convenient. If you're
using this data, we'd love to hear about it at
webmaster@ecs.soton.ac.uk. For discussion on our RDF,
join
http://mailman.ecs.soton.ac.uk/mailman/listinfo/ecsrdf,
for annoucements of changes, join
http://mailman.ecs.soton.ac.uk/mailman/listinfo/ecsrdf-
announce."^^xsd:string;
```

Das UNIPROT Konsortium[23], eine Forschungsgemeinschaft zum Thema Protein-Sequenzierung, veröffentlicht ihre Community Norm hingegen als HTML-Statement[24] auf dem Datenportal datahub.io in folgender Form:

```
Copyright 2007-2012 UniProt Consortium. We have chosen
to apply the Creative Commons Attribution-NoDerivs
License (http://creativecommons.org/licenses/by-nd/3.0/)
to all copyrightable parts (http://sciencecommons.org/)
of our databases. This means that you are free to copy,
distribute, display and make commercial use of these
databases, provided you give us credit. However, if
you intend to distribute a modified version of one
of our databases, you must ask us for permission first.
All databases and documents in the UniProt FTP directory
may be copied and redistributed freely, without advance
permission, provided that this copyright statement is
reproduced with each copy.
```

Wiederum einen anderen Zugang verfolgt das International Press and Telecommunications Council (IPTC), welches eine sehr umfangreiche Community Norm zur Nutzung ihrer Vokabulars NewsML[25] und dessen Einbettung in Bilder und Bilddatenbaken bereitstellt.[26]

[23] Siehe http://www.uniprot.org/help/about, aufgerufen am 20.01.2014.
[24] Siehe http://datahub.io/dataset/uniprot, aufgerufen am 20.01.2014.
[25] Siehe http://www.iptc.org/site/News_Exchange_Formats/NewsML-G2/, aufgerufen am 20.01.2014.
[26] Siehe http://www.embeddedmetadata.org/embedded-metatdata-manifesto.php, aufgerufen am 20.01.2014.

3.6 Status Quo der Linked Data Lizenzierung – Diskrepanz zwischen Theorie und Praxis

Eine Untersuchung der verwendeten Lizenzen in der Linked Data Cloud [21] offenbart eine aus mehreren Perspektiven unbefriedigende Situation. Tabelle 3.5 veranschaulicht die Lizenzmodelle jener Linked Data Sets, die über das Datenportal http://datahub.io zur Verfügung gestellt werden.[27]

Der Status Quo lässt sich folgendermaßen zusammenfassen: Noch hat sich keine Konvention zur Deklarierung von Policies, die das vollständige Rechtsspektrum von Linked Data abdecken, ausgebildet. Bisher werden hauptsächlich urheberrechtlich relevante Aspekte lizenziert. Die Verwendung von datenbankrelevanten Lizenzmodellen hat sich noch kaum etabliert. Allerdings ist zu berücksichtigen, dass mit der Creative Commons Version 4.0 auch datenbankrechtliche Aspekte stärker in der CC-Lizenzierung Berücksichtigung finden und dadurch eine bessere Rechtsabdeckung gewährleistet ist. Die Situation wird weiters durch regional abweichende Rechtsregime kompliziert. Während

Tab. 3.5 Überblick Lizenzen auf http://datahub.io (Stand 10.07.2013)

License	Number of Datasets
License Not Specified	251
Creative Commons Attribution	135
Creative Commons CCZero	72
Creative Commons Attribution Share-Alike	71
Creative Commons Non-Commercial (Any)	49
Other (Attribution)	38
UK Open Government Licence (OGL)	36
Open Data Commons Open Database License (ODbL)	28
Open Data Commons Public Domain Dedication and Licence (PDDL)	27
Other (Not Open)	26
Other (Open)	25
Other (Public Domain)	25
Open Data Commons Attribution License	14
GNU Free Documentation License	9
Other (Non-Commercial)	9
ukcrown-withrights	6
W3C	1
apache	1
gpl-2.0	1
gpl-3.0	1

[27] Eine vergleichbare Erhebung und kritische Reflexion findet sich auch bei Jain et al. [10].

das Datenbankrecht ein EU-Spezifikum darstellt, werden in den USA datenbankrechtliche Sachverhalte durch den Copyright Act gedeckt. Hinzu kommt, dass im Gegensatz zu Europa Datensätze, die über keine Lizenz verfügen, automatisch der Public Domain zugerechnet werden, wohingegen dies in Europa explizit deklariert werden muss. Ein Blick auf verfügbare Community Normen zeigt, dass diese in Umfang, Formulierung und Zugänglichkeit stark voneinander abweichen. Auch hat sich die Verwendung von Rights Expression Languages kaum etabliert, wodurch nur sehr eingeschränkte Möglichkeiten existieren, Datensätze auf Basis ihrer maschinenlesbaren Lizenzinformation einer automatischen Prozessierung zuzuführen, etwa für Zwecke der Aggregation, Versionierung und Servicierung von Content.

3.7 Conclusio und Ausblick

Während Metadaten bisher aufgrund ihrer oftmals proprietären Strukturen und Repräsentationsstandards nicht dazu geeignet waren, auf Basis netzwerkökonomischer Prinzipien bewirtschaftet zu werden, ändert sich dies durch Linked Data radikal. Der Schlüssel zur gewerblichen Diversifikation ist eine Strategie der Rechtediversifikation, die im Sinne der Versionierung sowohl die Bedienung des Web-Ökosystems als auch des Corporate Marktes ermöglicht und gleichzeitig Rechtssicherheit schafft. Die notwendige Kulturtechnik im Sinne der maschinellen Bereitstellung von interoperablen Lizenzinformationen mittels Rights Expression Languages entlang des Urheber- und Datenbankrechts ist allerdings noch sehr schwach ausgeprägt, was zum Einen auf eine fehlende technische Infrastruktur in Form leicht bedienbarer, systemisch integrierten Tools zur Kompilation und Annotation von Lizenzen, zum Anderen auf fehlende ökonomische Incentives zur Veröffentlichung von Linked Data zurückzuführen ist. Vor dem Hintergrund der weiteren technologischen Ausdifferenzierung von Linked Data Technologien zur maschinellen Datenbewirtschaftung ist jedoch von der inkrementellen Herausbildung einer Kulturtechnik der Linked Data Lizenzierung auszugehen. Die technologischen und juristischen Voraussetzungen dafür sind geschaffen.

Literatur

1. Auer, Sören 2011. Creating Knowledge Out of Interlinked Data. *Proceedings of WIMS'11* 2011(May 25–27): 1–8
2. Beer, David 2009. Power through the algorithm? Participatory web cultures and the technological unconscious. *new media & society* 11(6): 985–1002
3. Berners Lee, Tim. 1998. Uniform Resource Identifiers (URI): Generic Syntax. *IETF Network Working Group*. Request for Comments: 2396, See also: http://www.ietf.org/rfc/rfc2396.txt. Zugegriffen: 20. Februar 2013
4. Berners-Lee, Tim. 2006/2009. Linked Data Design Issues. http://www.w3.org/DesignIssues/LinkedData.html. Zugegriffen: 25. Mai 2013

5. Blumauer, Andreas, und Tassilo Pellegrini. 2009. *Social Semantic Web*. Berlin: Springer Verlag
6. Cranford, Steve 2009. *Spinning a Data Web. In: Price Waterhouse Coopers (Ed.). Technology Forecast, Spring 2009*. http://www.pwc.com/us/en/technology-forecast/spring2009/index.jhtml. Zugegriffen: 20. September, 2013
7. Dodds, L., und Ian Davis. 2009. *MP Data SPARQL Editor*. http://www.guardian.co.uk/open-platform/apps-mp-data-sparql-editor. Zugegriffen: 20. April, 2012
8. Frey, Carl, und Michael Osborne. 2013. The Future of Employment: How Susceptible are Jobs to Computerisation? Working Paper. University of Oxford. Siehe auch http://www.oxfordmartin.ox.ac.uk/downloads/academic/The_Future_of_Employment.pdf. Zugegriffen: 20. Dezember 2013
9. Graube, Markus, Johannes Pfeffer, Jens Ziegler, und Leon Urbas. 2011. *Linked Data as integrating technology for industrial data* Int. Conference on Network-Based Information Systems, 7–9 Sept. 2011., 162–167
10. Haase, Kenneth 2004. Context for Semantic Metadata. In *Proceedings of MM'04* October 10–16, 2004. New York, USA: ACM
11. Jain, Prateek, Pascal Hitzler, Krzysztof Janowicz, und Chitra Venkatramani. 2013. *There's No Money in Linked Data*. http://knoesis.wright.edu/faculty/pascal/pub/nomoneylod.pdf. Zugegriffen: 18. Dezember, 2013
12. Kulathuramaiyer, N., und Hermann Maurer. 2009. Implications of Emerging Data Mining. In *Social Semantic Web*, Hrsg. Andreas Blumauer, T. Pellegrini, 469–484. Berlin: Springer Verlag
13. Latif, Atif, Anwar Us Saeed, Patrick Höfler, Alexander Stocker, und Claudia Wagner. 2009. The Linked Data Value Chain: A Lightweight Model for Business Engineers. In *Proceedings of I-Semantics 2009, the 5th International Conference on Semantic Systems*, 568–577. Graz: Journal of Universal Computer Science
14. Lusch, Robert F., und S.L. Vargo. 2006. Service-dominant logic: reactions, reflections and refinements. *Marketing Theory* 6(3): 281–288
15. McKinsey Global Institute 2011. *Big data: The next frontier for innovation, competition and productivity. Research Report*. http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation. Zugegriffen: 24. Februar 2014
16. Mitchell, Ian, und Mark Wilson. 2012. *Linked Data. Connecting and exploiting big data. Fujitsu White Paper, March 2012*. http://www.fujitsu.com/uk/Images/Linked-data-connecting-and-exploiting-big-data-%28v1.0%29.pdf. Zugegriffen: 12. September, 2013
17. Nagenborg, Michael 2009. Privacy im Social Semantic Web. In *Social Semantic Web*, Hrsg. Andreas Blumauer, T. Pellegrini, 484–506. Berlin: Springer Verlag
18. Pellegrini, T., und Andreas Blumauer. 2006. *Semantic Web. Wege zur vernetzten Wissensgesellschaft*. Berlin: Springer Verlag
19. Pellegrini, T. et al. 2012. Semantic Metadata in the News Production Process. Achievements and Challenges. In *Proceeding of the 16th International Academic MindTrek Conference 2012*, Hrsg. Artur Lugmayr, 125–133. ACM SIGMM
20. Pellegrini, T. 2013. The Economics of Big Data: A Value Perspective on State of the Art and Future Trends. In *Big Data Computing*, Hrsg. R. Akerkar, 343–371. New York: Chapman and Hall/CRC
21. Pellegrini, T., und Ivan Ermilov. 2013. *Guide and Best Practices to Licensing Interlinked Data. Public Deliverable 7.4. EU-Project LOD 2. Grant Agreement No: 257943*. http://svn.aksw.org/lod2/WP7/D7.4/public.pdf. Zugegriffen: 03. Januar 2014

22. Pellegrini, T. 2014. Linked Data Licensing – Datenlizenzierung unter netzökonomischen Bedingungen. In *Transparenz. Tagungsband des 17. Internationalen Rechtsinformatik Symposium IRIS 2014*, Hrsg. Erich Schweighofer, Franz Kummer, Walter Hötzendorfer, 159–168. Wien: Verlag der Österreichischen Computergesellschaft

23. Prenafeta, Javier 2010. Protecting Copyright Through Semantic Technology. *Publishing Research Quarterly* 26(4): 249–254

24. Rayfield, J. 2012. *Sports Refresh: Dynamic Semantic Publishing. In: BBC Internet Blog*. http://www.bbc.co.uk/blogs/bbcinternet/2012/04/sports_dynamic_semantic.html. Zugegriffen: 5. Mai, 2012

25. Saumure, K., und Ali Shiri. 2008. Knowledge organization trends in library and information studies: a preliminary comparison of pre- and post-web eras. *Journal of Information Science* 34(5): 651–666

26. Shy, O. 2001. *The Economics of Network Industries*. Cambridge: Cambridge University Press

27. Sonntag, Michael 2006. Rechtsschutz für Ontologien. In *e-Staat und e-Wirtschaft aus rechtlicher Sicht*, Hrsg. Erich Schweighofer, Doris Liebwald, Matthias Drachsler, Anton Geist, 418–425. Stuttgart: Richard Boorberg Verlag

Teil II
Methoden

Datenintegration im Unternehmen mit Linked Enterprise Data

4

Sören Auer, Rene Pietzsch und Jörg Unbehauen

Zusammenfassung

Datenintegration ist in großen Unternehmen nach wie vor eine zentrale Herausforderung und wird es auch auf absehbare Zeit bleiben. Ein erfolgversprechender Ansatz ist die Verwendung des Linked Data Paradigmas für die Integration von Unternehmensdaten. Ebenso wie inzwischen ein Web der Daten das Dokumenten-zentrierte Web ergänzt, können Daten-Intranets die existierenden Intranet- und SOA-Landschaften in großen Unternehmen erweitern und flexibilisieren. Ein weiterer Vorteil des Linked Data Paradigmas ist die Möglichkeit der Nutzung von Daten aus der inzwischen auf über 50 Mrd. Fakten angewachsenen Linked Open Data (LOD) Cloud. Im Ergebnis kann ein unternehmensinternes Daten-Intranet dazu beitragen die Brücke zwischen strukturiertem Datenmanagement (in ERP, CRM, SCM Systemen) sowie semi- und unstrukturierten Informationen (Dokumente, Wikis, Portale) der Intranetsuche zu schlagen.

4.1 Einführung

Datenintegration in Unternehmen ist ein entscheidendes, aber aufwendiges und herausforderndes Problem. Während operativ geschäftskritische Informationen oft schon in integrierten Informationssystemen wie Enterprise-Resource-Planning (ERP), Customer-Relationship-Management (CRM) und Supply-Chain-Management (SCM) Systemen

S. Auer ✉
Institut für Informatik III, Rheinische Friedrich-Wilhelms-Univ. Bonn, Bonn, Deutschland
e-mail: auer@cs.uni-bonn.de
R. Pietzsch
eccenca GmbH, Leipzig, Deutschland
J. Unbehauen
Universität Leipzig, Leipzig, Deutschland

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_4

verwaltet werden, ist die Integration dieser Systeme selbst sowie die Integration mit der Fülle von Informationen aus anderen Quellen nach wie vor eine zentrale Herausforderung im Enterprise Data Management. Große Unternehmen nutzen oft hunderte oder sogar tausende verschiedene Informationssysteme und Datenbanken. Dies gilt insbesondere für große Konzerne und Hersteller. Es wird zum Beispiel geschätzt, dass im Volkswagen-Konzern ca. 5000 verschiedene Informationssysteme im Einsatz sind. Bei Daimler sind – auch nach einem Jahrzehnt der Konsolidierungsbemühungen – immer noch ca. 3000 unabhängige IT-Systeme im Einsatz.

Mit der zunehmenden Verbreitung von Informationstechnologie in Unternehmen wurde auch eine Vielzahl verschiedener Ansätze, Techniken und Methoden zur Lösung der Datenintegrations-Herausforderungen entwickelt. In den letzten zehn Jahren basierten die gängigen Datenintegrationsansätze in erster Linie auf XML, Web Services und Service-orientierten Architekturen (SOA). XML definiert einen Standard für die syntaktische Datenrepräsentation, Web Services bieten Datenaustauschprotokolle und SOA ist ein ganzheitlicher Ansatz für eine verteilte Systemarchitektur und Kommunikation. Trotz großer Fortschritte wachsen die Herausforderungen durch die Proliferation von IT weiter und es wird zunehmend klar, dass diese Technologien nicht ausreichen, um die Datenintegrationsprobleme in großen Unternehmen zu lösen. Insbesondere ist der mit SOA verbundene Aufwand oftmals zu hoch, um eine flexible, effiziente und effektive Datenintegration in der heutigen dynamischen Unternehmenswelt zu realisieren.

Die Autoren argumentieren, dass sich klassische SOA-Architekturen gut für die Transaktionsverarbeitung eignen, aber effizientere Technologien zur Verfügung stehen um die Daten-Vernetzung und Datenintegration zu unterstützen. Ein erfolgversprechender Ansatz ist die Verwendung des Linked Data Paradigmas für die Integration von Unternehmensdaten. Ebenso wie inzwischen ein Web der Daten das Dokumenten-zentrierte Web ergänzt, können Daten-Intranets die existierenden Intranet- und SOA-Landschaften in großen Unternehmen erweitern und flexibilisieren.

Ein weiterer Vorteil des Linked Data Paradigmas ist die Möglichkeit der Nutzung von Daten aus der inzwischen auf über 50 Mrd. Fakten angewachsenen Linked Open Data (LOD) Cloud. Beispiele für öffentliche LOD-Datenquellen, die für große Unternehmen relevant sind, umfassen *OpenCorporates*[1], eine Wissensdatenbank mit Informationen zu mehr als 50.000 Unternehmen weltweit, *LinkedGeoData*[2], eine von OpenStreetMaps abgeleitete geographische Wissensbasis, die genaue Informationen über alle Arten von räumlichen Entitäten enthält, sowie *Product Ontology*[3], die detaillierte Klassifikationen und Informationen über mehr als 1 Million Produkte umfasst. Für Unternehmen bietet die Erschließung solch großer, frei im Web verfügbarer Wissensbasen ein enormes Potenzial. Es ist jedoch entscheidend die Qualität dieser frei verfügbaren Wissensbasen zu

[1] Siehe http://opencorporates.com/, aufgerufen am 10.04.2014.
[2] Siehe http://linkedgeodata.org/, aufgerufen am 10.04.2014.
[3] Siehe http://www.productontology.org/, aufgerufen am 10.04.2014.

bewerten und mit zusätzlichen nicht-öffentlichen Informationen des Unternehmens (z. B. Unternehmes-Taxonomien, Domain-Datenbanken usw.) zu vernetzen.

Großunternehmen stehen vor der Herausforderung ihre IT-Infrastruktur auf drastische Weise zu flexibilisieren, um den Anforderungen einer modernen Dienstleistungswirtschaft gerecht zu werden und konkurrenzfähig zu bleiben. Das bedeutet unter anderem, dass Informationen und Daten in immer kürzeren Zeitabständen – idealerweise in Echtzeit – integriert werden müssen. Gleichzeitig besteht das Ziel die Integrationskosten stetig zu senken. Flexibilität wird aber auch bei der Transformation von Unternehmen erwartet, zum Beispiel bei Umstrukturierungen im Zuge von Fusionen, Übernahmen oder Rationalisierungsmaßnahmen. Nach interoperablen, anpassungsfähigen und flexiblen Standards konzipierte IT-Systeme können erfolgskritische Faktoren bei der Unternehmenstransformation darstellen und illustrieren die Notwendigkeit entsprechende Paradigmen umzusetzen. Der Einsatz von Linked Data für den Aufbau und die Verwaltung von Unternehmensdaten, Intranets und Wissensbasen wird die digitale Innovationsfähigkeit von Großunternehmen verbessern.

4.2 Evolution vom Web der Dokumente zum Web der vernetzten Daten

Seit der Entstehung des World Wide Web ist eine permanente Weiterentwicklung der zugrunde liegenden Standards und darauf aufbauenden Technologien und Anwendungen beobachtbar. Wurden Anfang der 1990er Jahre vor allem noch statische Webseiten veröffentlicht, so tauchten einige Jahre später bereits dynamische Webanwendungen, Content-Management- und E-Commerce-Systeme auf, die Inhalte aus Datenbanken und anderen strukturierten Datenquellen beziehen und Web-Seiten nutzerspezifisch generieren. Seit der Jahrtausendwende gewinnen vor allem soziale Funktionen in Web-Technologien und -Anwendungen an Bedeutung, die unter dem Oberbegriff Web 2.0 subsumiert werden. Dazu gehören insbesondere Netzeffekte durch soziale Netzwerke, Crowdsourcing und Mashups, die Daten aus verschiedenen Quellen integrieren. Seit einigen Jahren können wir nun einen Wandel zu einem Web der Daten beobachten, bei welchem strukturierte Daten (zusätzlich zu Markup-Text) in Form von RDFa, Microdata oder RDF/Linked Data veröffentlicht werden. Inzwischen kann man diesen Wandel eindrucksvoll an einer Vielzahl von Beispielen beobachten, zu denen insbesondere die von Google, Bing, Yandex und Yahoo vorangetriebene Initiative *schema.org*[4], die Linked Open Data Cloud, Google's Knowledge Graph oder auch Facebook's Open Graph gehören (vgl. Abb. 4.1).

Mit einigem zeitlichen Abstand haben Web-Technologien auch in den Intranets großer Unternehmen und Organisationen Einzug gehalten. Bereits Mitte der 1990er Jahre entstanden erste Intranets mit Web-Seiten und Angeboten für Mitarbeiter. Mit einiger zeitlicher Verzögerung wurden jeweils auch die weiterentwickelten Technologien des World

[4] Siehe http://schema.org, aufgerufen am 10.04.2014.

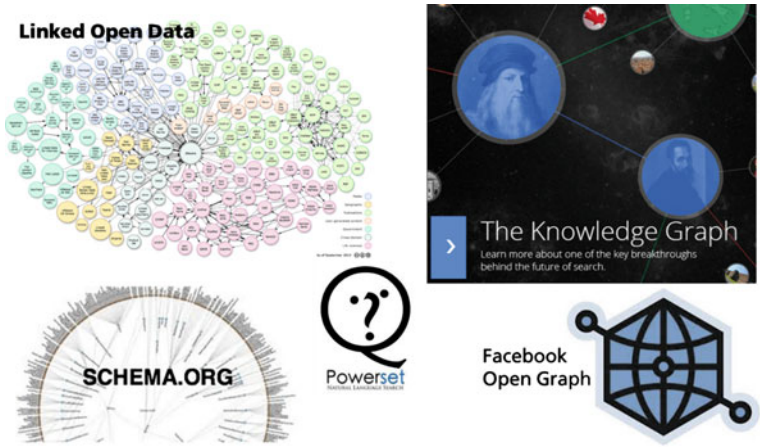


Abb. 4.1 Beispiele für die Erweiterung des Webs um ein „Web of Data“

Wide Web in Intranets angewendet. Inzwischen haben sich auch Portale, Wikis und Mashups fest in Intranets großer Unternehmen etabliert. Es ist daher zu erwarten, dass die derzeitige Evolution des World Wide Web zu einem Web of Data bzw. Daten-Web auch innerhalb von Intranets vollzogen wird und Daten-Web-Technologien verstärkt für die Vernetzung von Daten und Informationssystemen in den Intranets großer Unternehmen und Organisationen genutzt werden (vgl. Abb. 4.2). Dies ist darin begründet, dass Intranets großer Organisationen und das World Wide Web viele Gemeinsamkeiten haben:

Dezentral: Sowohl das Web als auch Intranets großer Organisationen sind oft sehr dezentral und verteilt aufgebaut. Innerhalb von großen Unternehmen liegt das daran, dass oft eine Vielzahl von Organisationseinheiten, Länderorganisationen und Tochterunternehmen existiert und daher eine starke Zentralisierung oft nicht möglich und sogar nicht erwünscht ist. So würde eine Zentralisierung z. B. die Integration gekaufter Unternehmen oder den Verkauf von Unternehmensteilen sehr stark erschweren. Unternehmen gehen daher eher dazu über einen gewissen Grad an Dezentralisierung zu akzeptieren und durch Standards eine Interoperabilität zwischen den betroffenen IT-Systemen sicherzustellen.

Heterogen: Informationsstrukturen in großen Organisationen sind fast so vielfältig wie im WWW selbst. Es existiert eine Vielzahl von Fachanwendungen, spezialisierten Datenbanken, XML-Schemata, Spreadsheets, Wikis, Portalen usw. usf. Für die nachhaltige Integration derselben ist es daher notwendig, geeignete Standards und Technologien zur Verfügung zu haben, von denen Linked Data einen wichtigen methodischen Überbau darstellt.

Eigenverantwortlich: Obwohl Unternehmen mit einem CEO und oft auch einem CIO (Chief Information Officer) ein zentrales Management haben, ist es inzwischen akzeptiert und sogar gewünscht, dass Abteilungen, Unternehmensteile und Tochter-

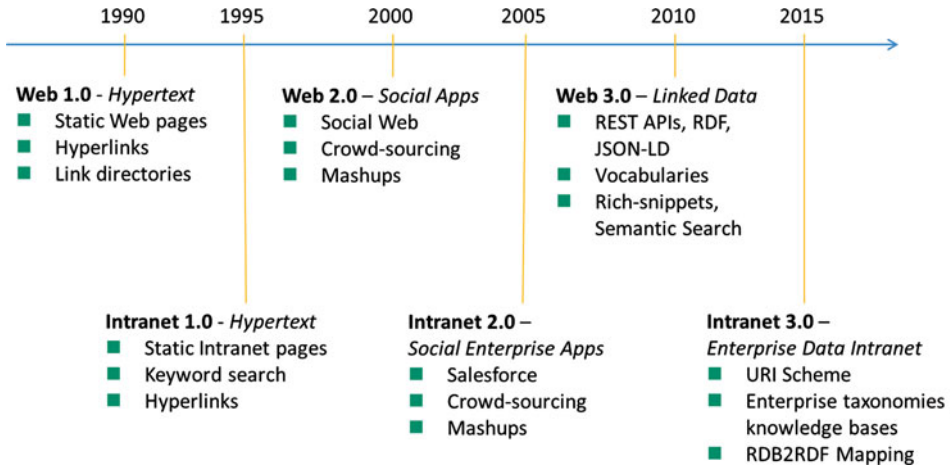


Abb. 4.2 Evolution des Web zu einem Web der Daten und von Unternehmens-Intranets zu Daten-Intranets

unternehmen eigenverantwortlich agieren um bestimmte Erfolgskennzahlen (Key-Performance-Indicator) zu erreichen. Diese Eigenverantwortlichkeit geht damit einher, dass Unternehmensteile oft auch die Möglichkeit haben eigene Konzeptualisierungen, Informationssysteme und Datenstrukturen zu realisieren, die u.U. im Kontrast zu jenen in anderen Unternehmensteilen stehen. Dies ist sehr ähnlich zur Situation im WWW, wo jeder Betreiber eines Angebots frei in der Wahl seiner Informationsstrukturen ist.

Diese Merkmale des Web, als verteiltes und dezentrales Informationsmedium weitgehend ohne zentrale Kontrolle, haben zur Entwicklung der Linked Data Standards[5] geführt:

- Identifikation von Daten und Informationseinheiten über HTTP-URIs
- Rückgabe von standardisiert repräsentierten, strukturierten Daten, wenn diese HTTP-URIs abgerufen werden
- Verweis auf andere, verwandte Daten

Für die Nutzung der Linked Data Prinzipien im Unternehmen sollten diese zu Linked Enterprise Data Prinzipien erweitert werden (vgl. Abb. 4.3):

- Weiterentwicklung von bestehenden Thesauri, Taxonomien, Wikis und Master-Data-Management-Systemen in Unternehmens-Wissensbasen und Wissenshubs
- Etablierung eines unternehmensweiten URI Schemas

[5] Siehe http://www.w3.org/DesignIssues/LinkedData.html, aufgerufen am 10.04.2014.

- Erweiterung der bestehenden Informationssysteme im eigenen Intranet um Linked Data Schnittstellen
- Herstellen von Verlinkungen zwischen in Zusammenhang stehenden Informationen

In den folgenden Abschnitten dieses Kapitels erläutern wir diese Linked Enterprise Data Prinzipien im Detail.

4.3 Etablierung von Unternehmenswissensbasen

Der Erfolg eines Unternehmens ist zentral in der Verfügbarkeit von domänenspezifischem Wissen begründet. Dieses Domänenwissen findet sich in organisationsübergreifenden Quellen wie Fachbüchern und Standards, aber auch in unternehmensspezifischen Quellen wie Glossaren, unternehmensinternen Dokumenten, Datenschemata, Taxonomien, etc. Einige Unternehmen haben bereits begonnen die geschäfts- und prozessrelevanten Begriffe mittels Unternehmensthesauri zu standardisieren und in verschiedenen Sprachen bereitzustellen. Ein solcher Unternehmensthesaurus enthält dabei Begriffe sowie deren Definitionen, Synonyme und Beziehungen zueinander, ähnlich wie *WordNet*[6] oder *Wiktionary*[7] dies domänen-unspezifisch bereitstellen. Mittels Crowdsourcing wurden solche Thesauri bereits effizient für eine Vielzahl von Sprachen erstellt. Ein ähnlicher Ansatz kann in Unternehmen verfolgt werden, indem Mitarbeitern mittels eines Wikis ermöglicht wird Begriffe und Beziehungen zwischen diesen zu erfassen. Ein solcher Unternehmensthesaurus kann sich dann zum Nukleus einer Unternehmenswissensbasis entwickeln. Neben Begriffen und deren Beziehungen zueinander sollte eine Unternehmenswissensbasis auch umfassende relevante Daten und Verknüpfungen zwischen diesen und anderen Daten- und Informationsquellen im Unternehmen enthalten. Ein Beispiel für eine Unternehmenswissensbasis ist Google's Knowledge Graph. Ursprünglich ähnlich wie DBpedia als Wikipedia Wissensextraktion durch Freebase gestartet, hat Google nach der Übernahme von Freebase den Knowledge Graph zu seiner zentralen Unternehmenswissensbasis ausgebaut. Inzwischen enthält der Knowledge Graph neben allgemeinen Daten auch sehr spezifische Themenbereiche wie Musikstücke, Bücher oder Filme.

4.4 Identifikation über ein unternehmensweites URI-Schema

Grundvoraussetzung, um die Integration und Vernetzung von Informationen im Unternehmen voranzutreiben, ist es ein eindeutiges Bezeichnersystem für Dinge wie Personen, Orte, Organisationen aber auch Begriffe, Datenelemente, Produkte, Verträge usw. zu etablieren. Im Web haben sich für die weltweit eindeutige Identifikation von und den Zugriff auf Ressourcen *Universal Resource Locator* (URL) und *Universal/Internationalized Resource Identifier* (URI/IRI) etabliert. Der Unterschied zwischen URLs und URIs ist dabei

[6] Siehe http://wordnet.princeton.edu/, aufgerufen am 10.04.2014.
[7] Siehe http://wiktionary.org/, aufgerufen am 10.04.2014.

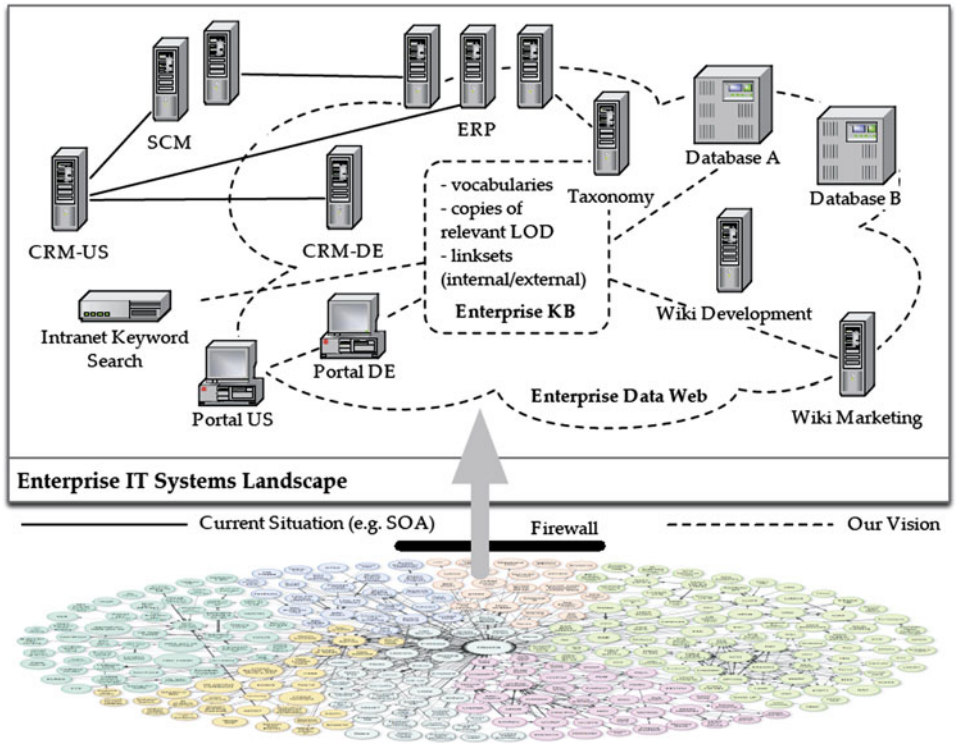


Abb. 4.3 Vernetzung von Daten und Informationen im Intranet mittels einer Unternehmenswissensbasis als Enterprise Knowledge Hub (cf. [7]). Die *durchgezogenen Linien* zeigen, wie IT-Systeme bereits verbunden sind. Die *gestrichelten Linien* visualisieren, wie IT-Systeme mittels eines internen Linked-Data-Web miteinander verknüpft werden. Ein Unternehmens-Daten-Web umfasst eine Enterprise Knowledge Base (EKB), die Wortschatz-Definitionen, Kopien von relevanten Linked Open Datasets sowie interne und externe Linksets zwischen Datensätzen. Daten aus der öffentlichen LOD Cloud können innerhalb des Unternehmens wiederverwendet werden, aber interne Daten sind vor externem Zugriff – wie in üblichen Intranets – gesichert

eher theoretischer Natur und verschwimmt in der Praxis immer weiter. Im Gegensatz zu URLs, die immer über das HTTP-Protokoll abrufbar sind, besteht diese Anforderung an URIs prinzipiell nicht. Es wird aber inzwischen als gute Praxis angesehen auch URIs grundsätzlich über das HTTP-Protokoll abrufbar zu machen. URIs zeichnen sich insbesondere durch folgende Vorteile aus:

Dezentralität: Jeder Besitzer eines Domänennamens, eines über einen Namen erreichbaren Servers oder eines Webspace auf einem Server ist in der Lage eigene URIs zu definieren, die den Domänen-, Servernamen oder die Webspace-Adresse als Präfix beinhalten. Dies ist für größere Organisationen und Unternehmen wichtig, da verschiedene Organisationseinheiten (z. B. Tochterunternehmen, Ländergesellschaften,

Tab. 4.1 Vor- und Nachteile unterschiedlicher Ansätze des URI-Managements

Managementtyp	Vorteile	Nachteile
Zentrales URI-Management: Ein zentraler Dienst vergibt auf Anforderung URIs unter einem einheitlichen Namensschema.	- Überblick über alle vorhandenen Ressourcen - Starke Standardisierung der URI-Struktur	- Single-Point-of-Failure - Mangelnde Flexibilität - Schwierige Sicherstellung der Dereferenzierbarkeit
Zentrale Registratur dezentraler URIs: Neue dezentral vergebene URIs müssen bei einem zentralen Dienst registriert werden.	- Überblick über alle vorhandenen Ressourcen - Ausfallsicherheit und Robustheit gegenüber organisatorischen Veränderungen	- Synchronisation erforderlich
Dezentrales URI-Management: Identifikatoren werden komplett dezentral verwaltet.	- Hohe Flexibilität - Hohe Ausfallsicherheit und Robustheit gegenüber organisatorischen Veränderungen	- Fehlender Überblick - Mangelnde Standardisierung

Abteilungen) bei Bedarf eigene URIs als Bezeichner für ihre Entitäten und Geschäftsobjekte definieren können. Es kann jedoch alternativ bei Bedarf auch ein zentrales URI-Management etabliert werden, bei dem Identifier zentral vergeben werden.

Dereferenzierbarkeit: URIs dienen nicht nur zur Identifikation von Entitäten, sondern auch als Zugriffspfad auf Informationen zu diesen Entitäten.

Herkunft und Nachvollziehbarkeit: Da Domainnamen von Registraren verwaltet werden, kann über diese ihr Eigner ermittelt werden. URIs geben damit Auskunft über den ursprünglichen Autor und können mittels HTTP abgerufen werden, um die Authentizität und Korrektheit der dahinter verborgenen Informationen zu überprüfen.

Tabelle 4.1 gibt einen Überblick über Vor- und Nachteile verschiedener Formen eines unternehmensweiten URI-Managements.

Es sind auch verschiedene Zwischenstufen zwischen diesen drei Paradigmen möglich. So kann z. B. eine föderierte URI-Registratur realisiert werden, bei der Organisationseinheiten eigene URI-Registraturen betreiben. Ebenso können auch bei einem komplett dezentralen Management von Identifikatoren Crawling- und Indizierungstechniken genutzt werden, um sich ähnlich einer Intranetsuche einen Überblick über verschiedene Linked Data Quellen im Unternehmen zu verschaffen. Die jeweilige Lösung sollte abgestimmt auf die Bedürfnisse des Unternehmens gewählt werden. Wenn Länderorganisationen oder Tochterunternehmen eine maximale Unabhängigkeit gewährt werden soll, sollte eine dezentrale Lösung zum Einsatz kommen. Wenn unternehmensweite Standardisierung und Integration zentrale Anforderungen sind, ist eine der zentralen Varianten des URI-Managements zu bevorzugen.

4.5 Etablierung eines Lebenszyklus vernetzter Daten im Unternehmen

Da Daten eine immer zentralere Rolle in modernen Unternehmen spielen, sollte der Lebenszyklus von Daten im Unternehmen möglichst umfassend unterstützt werden. Die verschiedenen Phasen des Linked Data-Lebenszyklus [4] sind in Abb. 4.4 dargestellt und umfassen insbesondere die folgenden Phasen:

RDF Datenmanagement: RDF Daten werden in Triple-Stores oder relationalen Datenbanken mit dedizierten RDF-Schnittstellen gespeichert. Durch den Einsatz von Column-Store-Technologie, dynamische Abfrage-Optimierung, adaptives Anfrage-Caching, optimierte Graph-Daten-Verarbeitung und Skalierbarkeit mittels Cluster- und Cloud-Technologien erreicht RDF Datenmanagement inzwischen eine mit relationalen Datenbanken vergleichbare Leistung bei höherer Daten-Model-Flexibilität.

Authoring: Ein wichtiger Aspekt ist die Erstellung und Bearbeitung von RDF-Daten. Die Erstellung von semantischen Wissensbasen kann insbesondere durch den Einsatz von Semantic Wiki-Technologie und seinem zugrunde liegenden WYSIWYM-Paradigma (What You See Is What You Mean) sowie eine verteilte gemeinschaftliche Pflege der Wissensbasis unterstützt werden.

Verlinkung: Das (teil-)automatisierte Erstellen und Verwalten von Verknüpfungen zwischen verschiedenen Wissensbasen ist ein zentraler Aspekt bei der Realisierung eines Datennetzes im Unternehmen und von entscheidender Bedeutung für Daten-Kohärenz und Datenintegration.

Klassifizierung & Anreicherung: Linked Data aus dem Web aber auch aus unternehmensinternen Quellen besteht in erster Linie vor allem aus Instanz-Daten. Für Daten-Integration, Fusion, Suche und viele andere Anwendungen müssen diese Rohdaten jedoch auf Taxonomien, Vokabulare und Ontologien gemappt und mit diesen integriert werden.

Abb. 4.4 Die Phasen des Linked Data Lebenszyklus

Qualität: Die Qualität von Daten aus dem Daten-Web variiert ebenso wie die von Daten aus unternehmensinternen Quellen. Ein wichtiger Aspekt im Lebenszyklus ist daher die Anwendung von Techniken zur Beurteilung der Datenqualität für einen bestimmten Anwendungsfall mittels Merkmalen wie Herkunft, Kontext, Abdeckung oder Struktur.

Evolution & Reparatur: Daten sind oft dynamisch. Wir müssen die Evolution von Daten angemessen unterstützen und dabei die Nutzung der Daten stabil halten. Änderungen und Modifikationen an Wissensbasen, Vokabularen und Ontologien sollten transparent und beobachtbar sein. Automatisierte Methoden können helfen Probleme in Wissensbasen zu erkennen und passende Reparaturstrategien vorzuschlagen.

Suche & Exploration: Für Anwender müssen Daten und Beziehungen zwischen diesen Daten im Intranet und Daten-Web sichtbar gemacht werden. Dafür können verschiedene Such-, Browsing-, Explorations- und Visualisierungstechniken für verschiedene Arten von Linked Data (z. B. räumliche, zeitliche, statistische Daten) genutzt werden.

Idealerweise wird die Unterstützung dieser Lebenszyklusphasen nicht isoliert betrachtet, sondern Methoden werden angewendet, die sich gegenseitig ergänzen und unterstützen. Beispiele dafür:

- Die Erstellung von Mappings auf Schemaebene beeinflusst direkt die Erstellung von Verlinkungen auf Instanzebene und umgekehrt.
- Diskrepanzen in den Ontologie-Schemata unterschiedlicher Wissensbasen können durch das Erlernen von Äquivalenzen zwischen Konzepten kompensiert werden.
- Feedback- und Rückmeldungen von Endbenutzern (z. B. hinsichtlich Instanz- oder Schema-Verknüpfungen) können als Trainingsdaten für Techniken des maschinellen Lernens genutzt werden, um induktiv Verknüpfungen zu größeren Wissensbasen herzustellen, deren Ergebnisse wiederum von Endbenutzern für eine iterative Verfeinerung beurteilt werden können.
- Semantisch angereicherte Wissensbasen verbessern die Erkennung von Inkonsistenzen und Modellierungsproblemen, die wiederum die Vernetzung, Fusion und Klassifikation verbessern.
- Die Abfrageleistung des RDF-Datenmanagements wirkt unmittelbar auf alle anderen Komponenten, und die Art der Abfragen, die von den Komponenten ausgehen, wirkt auf das RDF-Datenmanagement.

Als Ergebnis dieser Abhängigkeiten, sollten wir die Einrichtung eines Verbesserungszyklus für Wissensbasen des Web of Data verfolgen. Die Verbesserung der Wissensbasis in Bezug auf einen Aspekt (z. B. eine Anreicherung durch die Verknüpfung mit einem neuen Wissens-Hub) löst eine Reihe von möglichen weiteren Verbesserungen (z. B. zusätzliche übereinstimmende Instanz) aus. Die Herausforderung besteht darin Techniken, die die Nutzung dieser gegenseitigen Befruchtungen im verteilten Medium des Web of Data ermöglichen, zu entwickeln. Eine Möglichkeit ist, dass verschiedene Algorithmen ein gemeinsam genutztes Vokabular für Veröffentlichung, Zusammenführen, Reparatur-

oder Anreicherung der Daten verwenden. Zusätzlich existiert ein Dienst, der seine neuen Erkenntnisse in einem allgemeingültigen Vokabular veröffentlicht. Über Benachrichtigungsmechanismen (wie *Semantic Pingback* [10]) können andere Dienste, die Updates für eine bestimmte Datendomäne abonniert haben, oder der ursprüngliche Herausgeber der Daten über Verbesserungsvorschläge informiert werden. Der ursprüngliche Herausgeber hat dann die Möglichkeit die Vorschläge zu prüfen und gegebenenfalls in die originären Daten zu übernehmen. Durch das Mitführen von Herkunftsinformationen (Provenance) ist die Änderungshistorie der Daten und ihrer Autorenschaft stets nachvollziehbar.

Der im Rahmen des FP7 LOD2-Projektes entwickelte Linked Data Stack [2] stellt eine umfassende Werkzeugunterstützung für die verschiedenen Phasen des Lebenszyklus bereit.

4.6 Linked Data Schnittstellen

Von zentraler Bedeutung ist es existierende Informationssysteme und Datenbanken mit Linked Data Schnittstellen auszustatten. Auch da strukturierte Daten größtenteils in relationalen Datenbanken gehalten werden, sind diese ein idealer Ausgangspunkt für einen Einsatz von Linked Data im Unternehmen. Ziel ist es diese Datenbanken im unternehmensinternen Linked Data Intranet verfügbar zu machen um damit Kristallisationspunkte für weitere Anwendungen zu schaffen. Integriert in den Linked-Data Lebenszyklus kann beispielsweise eine Extraktion benannter Entitäten (Named Entity Extraction) oder ein Linking auf diese Daten zurückgreifen und damit weitere Daten integrieren. Im Folgenden erläutern wir: ein Beispielszenario (Abschn. 4.6.1), zeigen welche grundsätzlichen Integrationsszenarien dabei in Frage kommen (Abschn. 4.6.2), Mappingsprachen (Abschn. 4.6.3) und grundlegende Konzepte des Mappings (Abschn. 4.6.4) sowie Tools und weitere Herausforderungen (Abschn. 4.6.5).

4.6.1 Beispielszenario

Zur Veranschaulichung dient als Beispiel eine Thesaurus-Datenbank, wie sie in zahlreichen Unternehmen zum Einsatz kommt und in Listing 4.1 dargestellt ist. Exemplarisch für solch eine Thesaurus-Datenbank stellen wir hier ein stark vereinfachtes System eines fiktiven Werkzeugherstellers vor. Diese Datenbank wurde angelegt um mehrsprachliche Kommunikation zu unterstützen.

Listing 4.1 Beispieldatenbank Thesaurus

```
1
2   FACHGEBIET              KONZEPTE
3                          |========|=======|=============|=============|
4  |================|      |KONZEPT |FACHGEB| BEGRIFF_DE  | BEGRIFF_EN  |
5  |NR   |FACHGEB   |      |========|=======|=============|=============|
6  |=====|==========|      | 1      | 1     | Hammer      | hammer      |
7  |1    |Werkzeuge |      | 2      | 1     | Feile       | file        |
8  |2    |Chemie    |      | 3      | 2     | Oel         | oil         |
9  |================|      |========|=======|=============|=============|
```

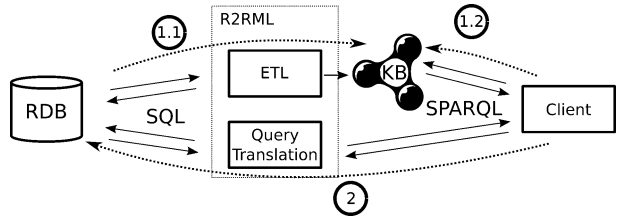


Abb. 4.5 Unterschiedliche Herangehensweisen an das Mapping relationaler Daten nach RDF: Übersetzung von Datenbankabfragen und RDF Extraktion

In dieser Datenbank werden die deutschen (`BEGRIFF_DE`) und englischen (`BEGRIFF_EN`) Begriffe jeweils mit einem Konzept und einem Fachgebiet verknüpft. Das Fachgebiet gruppiert wiederum Konzepte und dient der Zuordnung zu einem Zuständigkeitsbereich innerhalb des Unternehmens. Im Folgenden stellen wir vor, wie diese Datenbank als Linked Data publiziert werden kann. Wir nutzen dazu das *Simple Knowledge Organization System* (SKOS) [25], ein einfaches Vokabular zur Beschreibung von Thesauri oder ähnlichen Konzeptschemata.

4.6.2 Nutzungsszenarien

Bei der Veröffentlichung einer relationalen Datenbank als Linked Data sind zwei grundsätzliche Vorgehensweisen zu unterscheiden. Diese sind in Abb. 4.5 dargestellt und unterscheiden sich im Wesentlichen in der Frage, ob die Daten der relationalen Datenbank als RDF materialisiert werden (1.1 und 1.2) oder einen virtuellen RDF-Graphen (2) darstellen.

Im Falle des materialisierten Graphen wird mit Hilfe eines Mappings zunächst ähnlich eines Extract-Transform-Load (ETL) Verfahrens ein Export erstellt (1.1). Dieser wird in eine RDF-Datenbank geladen (1.2) und kann so beispielsweise als SPARQL-Endpunkt bereitgestellt oder per HTTP dereferenziert werden. Im Falle des virtuellen Graphen wird eine SPARQL-Abfrage oder eine HTTP-Abfrage mit Hilfe eines Mappings in eine oder mehrere SQL-Abfragen umgewandelt (2). Diese SQL-Abfragen werden von der relationalen Datenbank ausgewertet, das Ergebnis transformiert und schließlich an den Client zurückgesendet.

Im konkreten Fall unserer Beispieldatenbank sind beide Szenarien denkbar. Vorteil eines in einer RDF-Triple-Datenbank materialisierten RDF-Graphen ist die Nutzung der speziellen Indizes einer RDF-Datenbank. Vorteil der Veröffentlichung mittels Umwandlung von Abfragen ist der direkte Zugriff auf die Master-Daten ohne den Umweg über Extraktion, Transformation und erneutes Laden.

4.6.3 Mapping relationaler Datenbanken im Lebenszyklus von Linked Data

Neben der technischen Umsetzung muss die Integration in den Lebenszyklus konzipiert werden. Von besonderem Interesse sind in diesem Fall:

URI-Schema: Die in Abschn. 4.4 diskutierten Strategien müssen bei der Gestaltung von sowohl Schema-URIs als auch Instanz-URIs Verwendung finden. Ebenso müssen hier Domänenspezifika berücksichtigt werden. Beispielhaft ist hier die Entscheidung für SKOS für das Schema in Listing 4.1.

Sicherheit: Da die meisten Informationen im Unternehmen Zugriffsbeschränkungen haben, ist zu klären wie diese in einem Mapping zu berücksichtigen sind.

Aktualität: Insbesondere für die Festlegung auf ein Verfahren zur Bereitstellung, siehe Abschn. 4.6.2, ist der Aktualitätsanspruch an die Daten entscheidend. Beispielsweise muss eine versionierte Terminologiedatenbank nicht den gleichen Grad an Aktualität aufweisen wie eine Datenbank, die den aktuellen Status einer Produktionsmaschine abbildet. Im Falle der Terminologiedatenbank wäre es allerdings wichtig, eine solche Versionierung abzubilden und abfragbar zu gestalten.

Feedback: Das Mapping relationaler Datenbanken ist im Regelfall ein rein lesender Zugriff. Durch die Integration können Fehler innerhalb der Daten zu Tage treten. Da hier kein generischer Ansatz existiert, muss ein Feebackmechanismus etabliert werden.

4.6.4 Mappingsprachen, R2RML und Mappingkonzepte

Wesentliche Teile der Linked Open Data Cloud sind Extrakte relationaler Datenbanken. Die Extraktion relationaler Datenbanken wurde schon früh als wichtiges Prinzip zur Bereitstellung von Linked Data erkannt [8]. Daher existiert eine Vielzahl von Werkzeugen, von denen einige exemplarisch im Abschn. 4.6.5 vorgestellt werden. Mit diesen Werkzeugen wurden Sprachen entwickelt, die ein Mapping von relationalen Daten nach RDF ermöglichen. Mit der RDB-to-RDF Mapping Language (R2RML) [6] existiert seit September 2012 ein einheitlicher Standard zur Beschreibung solcher Mappings, der von einer Vielzahl unterschiedlicher Werkzeuge unterstützt wird.

Ein R2RML-Mapping beschreibt eine Transformation eines spezifischen Datenbankschemas. Diese Transformation beschreibt, wie aus den Zeilen und Spalten RDF-Terme und RDF-Tripel erzeugt werden. Im Weiteren wird das Präfix `rr:` verwendet, wenn das Vokabular der R2RML verwendet wird.

In Listing 4.2 stellen wir ein partielles Mapping der `KONZEPTE`-Tabelle unter Verwendung des SKOS-Vokabulars vor.

Listing 4.2 Mapping der Tabelle KONZEPTE

```
:TriplesMapKonzept
    rr:logicalTable [ rr:tableName "KONZEPTE" ];
    rr:subjectMap [
        rr:template "http://example.com/term/Konzept/{KONZEPT}";
        rr:class skos:Concept;
    ];
    rr:predicateObjectMap [
        rr:predicate skos:prefLabel;
        rr:objectMap [ rr:column "BEGRIFF_DE"; rr:language "de"];
    ].
```

Die wichtigsten Konzepte dieses Mappings, die sich auch in anderen Mapping-Sprachen wiederfinden, stellen wir im Folgenden vor.

Triple-Map Eine R2RML Triple-Map beschreibt, wie aus den Zeilen einer Tabelle oder einer Datenbank-Sicht RDF erzeugt wird. Im Beispiel in Listing 4.2 verknüpft die Triple-Map `:TriplesMapKonzept` (Zeile 1) die Datenbank-Tabelle `KONZEPTE` (Zeile 2) mit den Bildungsvorschriften für die RDF-Terme (Zeilen 3–10). Triple-Maps haben dabei immer eine `rr:SubjectMap` und beliebig viele `rr:PredicateObjectMaps`. Diese Bildungsvorschriften werden für jede Zeile der Tabelle ausgeführt. Angewandt auf die oben beschriebene Datenbank entsteht dabei der in Listing 4.3 serialisierte Graph.

Listing 4.3 Auswertung des Mappings auf Tabelle KONZEPTE

```
1 <http://example.com/term/Konzept/1>  skos:prefLabel "Hammer"@de.
2 <http://example.com/term/Konzept/1>  a skos:Concept.
3 <http://example.com/term/Konzept/2>  skos:prefLabel "Feile"@de.
4 <http://example.com/term/Konzept/2>  a skos:Concept.
5 <http://example.com/term/Konzept/3>  skos:prefLabel "Oel"@de.
6 <http://example.com/term/Konzept/3>  a skos:Concept.
```

Term-Map Eine Term-Map im Sinne der R2RML-Spezifikation ist vereinfacht ausgedrückt eine Funktion, die RDF-Terme, also URIs, Literale und Blank Nodes, erzeugt. Diese RDF-Terme bilden die Grundlage zur Definition von Triple-Maps. R2RML definiert dabei drei unterschiedliche Methoden zur RDF-Termerzeugung:

Template Wert: Ein RDF-Term wird durch Zusammenfügen von fixen Strings und Zellwerten erzeugt. In Zeile 4 in Listing 4.2 wird der Ressourcen-identifizierende URI mittels eines solchen Template erzeugt. Hierbei ist darauf zu achten, dass eindeutig identifizierende Spalten verwendet werden. Zudem sind bei der Gestaltung der Templates die Überlegungen zur Gestaltung unternehmensweiter URI-Räume Abschn. 4.4 zu berücksichtigen.

Spalten Wert: Ein RDF-Term wird direkt aus einer Zelle erzeugt. In Listing 4.2, Zeile 9, wird an der Objekt-Position ein Literal, das sich beispielsweise in Listing 4.3 in den ungeraden Zeilen an Position des Objektes wiederfindet.

Konstanter Wert: Ein fixer Wert wird unabhängig von Zellwerten erzeugt. Zeile 8 definiert einen solchen konstanten Wert. Bei einer relationalen Datenbank reflektiert dieser den Spaltennamen der Tabelle, die als Objekt in einem Triple verwendet werden soll. In Listing 4.2, Zeile 8, wird `skos:prefLabel` verwendet.

Durch weitere Annotationen können mit Hilfe dieser drei Methoden Ressourcen-URIs, Literale und Blank Nodes erzeugt werden.

Weitere Konstrukte: Term-Maps und Triple-Maps stellen die Kernkonzepte von R2RML dar.

rr:parentTriplesMap verknüpft Tabellen und erlaubt das Mapping von Tabellenverknüpfungen in RDF darzustellen. Angewandt auf das Beispiel in Listing 4.1 kann damit die Zuordnung der `KONZEPTE` zu `Fachgebieten` abgebildet werden.

rr:language definiert die Sprache eines Literals, wie im Beispiel in Listing 4.2, Zeile 9, dargestellt.

rr:class ermöglicht eine Zuordnung einer Ressource zu einer Klasse. Im Beispiel Listing 4.2, Zeile 5, wird eine Zuordnung zur Klasse `skos:Concept` vorgenommen.

4.6.5 Mapping Werkzeuge und Herausforderungen

Zur Erfüllung der unterschiedlichen Anforderungen an ein Mapping relationaler Daten wurden zahlreiche Werkzeuge für unterschiedliche Anwendungsgebiete entwickelt.

Triplify [3] ist ein leichtgewichtiges Werkzeug, das relationale Datenbanken als Linked Data bereit stellt. Durch die Verwendung von PHP kann es einfach in existierende Anwendungen integriert werden. Ebenso ist es möglich vollständige Abzüge der RDF-Daten zu erstellen.

D2R [5] stellt einen vollwertigen SPARQL-Endpunkt bereit. SPARQL Abfragen können dabei in eine oder mehrere SQL-Abfragen umgewandelt werden.

SparqlMap [25] ermöglicht das Mapping einer SPARQL-Abfrage auf genau eine SQL-Abfrage. Damit ist es möglich auch große Datenbanken als SPARQL-Endpunkt ohne Materialisierung zu exponieren.

Virtuoso [1] integriert eine umzuwandelnde Datenbank in die dem Triple-Store zu Grunde liegende Datenbank. Damit können beide Datenbanken effizient mittels SPARQL abgefragt werden.

4.7 Zusammenfassung und Ausblick

Im letzten Jahrzehnt wurden enorme Fortschritte bei der Konsolidierung und Integration von Daten und Informationen innerhalb von Unternehmen z. B. mittels SOA, ERP und CRM-Technologien erzielt. Dennoch existieren ungelöste Herausforderungen bei Integration produktionskritischer Daten und Informationen, die derzeit in semi-strukturierter oder unstrukturierter Form (Dokumente, Portale, Wikis etc.) vorliegen. Auch der Informationsaustausch zwischen Unternehmen und in Wertschöpfungsketten wurde hingegen nur marginal verbessert. Gerade in einer hochentwickelten arbeitsteiligen Wirtschaft führen die Spezialisierung und Fokussierung von Unternehmen auf Kernaufgaben zu immer komplexeren Liefer- und Wertschöpfungsnetzwerken, deren Beherrschung ein ausschlaggebender Erfolgsfaktor geworden ist. Von zentraler Bedeutung für die robuste, effiziente und effektive Ausgestaltung dieser Wertschöpfungsnetzwerke sind reibungslose Informationsflüsse zwischen den beteiligten Unternehmen. Beispiele für auszutauschende Informationen sind

z. B. Kontaktadressen und Ansprechpartner, Verbrauchs- und Lieferprognosen, Qualitätssicherungsdaten, Logistikinformationen.

Für den reibungslosen und medienbruchfreien Austausch solcher Informationen sind die folgenden zwei Anforderungen von essentieller Bedeutung:

- Informationen müssen flexibel aufgaben- und anwendungsspezifisch feingranular strukturiert werden und die dabei genutzten Informationsstrukturen müssen sich evolutionär geänderten Anforderungen an den Informationsaustausch anpassen.
- Der Informationsaustausch zwischen Unternehmen muss verteilt und dezentral in den Wertschöpfungsnetzwerken erfolgen und somit Datensicherheit, Zugriffskontrolle und Provenienz gewährleisten.

Das im Web bereits etablierte Linked Data Paradigma stellt Methoden, Standards und Technologien bereit um Daten und Informationen in Unternehmensintranets wesentlich effizienter und effektiver zu vernetzen und zu integrieren. Neuer Wert kann aus existierenden Daten und Informationen nur durch eine Vernetzung und Integration entstehen. Dafür ist es insbesondere wichtig gleiche, ähnliche und verwandte Informationen in verschiedenen Quellen zu identifizieren. Der Linked Data Lebenszyklus mit seinen Phasen Datenextraktion, Authoring, Verlinkung, Klassifizierung und Anreicherung, Datenqualität sowie Suche und Exploration unterstützt die inkrementelle Verbesserung und Integration von Daten in Unternehmens-Intranets. Im Ergebnis kann ein unternehmensinternes Daten-Intranet dazu beitragen die Brücke zwischen strukturiertem Datenmanagement (in ERP, CRM, SCM Systemen) sowie semi- und unstrukturierten Informationen (Dokumente, Wikis, Portale) der Intranetsuche zu vernetzen und integrieren.

Literatur

1. Mapping relational data to rdf with virtuoso's rdf views. http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html
2. Auer, S., L. Bühmann, C. Dirschl, O. Erling, M. Hausenblas, R. Isele, J. Lehmann, M. Martin, P.N. Mendes, B. van Nuffelen, C. Stadler, S. Tramp, und H. Williams. 2012. In *Managing the life-cycle of linked data with the LOD2 stack* Proceedings of International Semantic Web Conference (ISWC 2012)., 22. International
3. Auer, S., S. Dietzold, J. Lehmann, S. Hellmann, und D. Aumueller. 2009. Triplify – light-weight linked data publication from relational databases. *18th International World Wide Web Conference* 621–621, April 2009
4. Auer, S., J. Lehmann, und A.-C. Ngonga Ngomo. 2011. Introduction to linked data and its lifecycle on the web. In *Reasoning Web. Semantic Technologies for the Web of Data* Lecture Notes in Computer Science, Bd. 6848, Hrsg. A. Polleres, C. d'Amato, M. Arenas, S. Handschuh, P. Kroner, S. Ossowski, P. Patel-Schneider, 1–75. Berlin Heidelberg: Springer
5. Bizer, C. und R. Cyganiak. 2006. D2r server – publishing relational databases on the semantic web. Poster at the 5th Int. Semantic Web Conf. (ISWC2006)

6. Das, S., S. Sundara, und R. Cyganiak. 2012. R2rml: Rdb to rdf mapping language (w3c recommendation). Technical report
7. Frischmuth, P., S. Auer, S. Tramp, J. Unbehauen, K. Holzweißig, und C.-M. Marquardt. 2013. Towards linked data based enterprise information integration. In *Proceedings of the Workshop on Semantic Web Enterprise Adoption and Best Practice (WASABI) 2013*
8. Lee, T. B. 09 1998. Relational databases on the semantic web. Design Issues (published on the Web)
9. Miles, A. und S. Bechhofer. Aug. 2009. Skos simple knowledge organization system reference. W3C Recommendation, W3C
10. Tramp, S., P. Frischmuth, T. Ermilov, und S. Auer. 2010. Weaving a Social Data Web with Semantic Pingback. In *Proceedings of the EKAW 2010* Knowledge Engineering and Knowledge Management by the Masses, Lisbon, Portugal, 11th October–15th October 2010. Lecture Notes in Artificial Intelligence (LNAI), Bd. 6317, Hrsg. P. Cimiano, H. Pinto, 135–149. Berlin/Heidelberg: Springer
11. Unbehauen, J., C. Stadler, und S. Auer. et al. 2012. Accessing relational data on the web with sparqlmap. In *Proc. 2nd Joint International Semantic Technology Conference* Lecture Notes in Computer Science, Bd. 7774, Hrsg. T. Hideaki. Springer

Methoden der Linked Data Integration

5

Robert Isele

Zusammenfassung

Datenintegration bezeichnet das Zusammenführen unterschiedlicher Datensätze mit dem Ziel der gemeinsamen Abfrage und ist eine essentielle Voraussetzung für den Einsatz von Linked Data im Unternehmenskontext. Dieser Beitrag behandelt die Prozesse, welche notwendig sind um eine globale Sicht auf mehrere Datenquellen herzustellen. Da Linked Data Publisher eine Vielzahl verschiedener Vokabulare verwenden, um Informationen zu repräsentieren, gilt es zunächst die Datensets in ein konsistentes Zielvokabular zu überführen. Desweiteren müssen, in einem zweiten Schritt, Ressourcen in unterschiedlichen Datensätzen, welche dasselbe Realwelt-Objekt repräsentieren, identifiziert und verknüpft werden. Zuletzt müssen die zuvor verknüpften Ressourcen zu einer Entität verschmolzen werden.

5.1 Einführung

Im Unternehmenskontext sind die für neue Aufgaben notwendigen Daten in der Regel nicht in einer einzigen Datenbank vorhanden, sondern über mehrere Datenbanken verteilt. Das Ziel der Datenintegration ist das Herbeiführen einer konsistenten Sicht auf heterogene Daten, indem Daten gleichartig strukturiert und verwandte Ressourcen sinnvoll verknüpft werden.

Ein vollständiger Datenintegrationsprozess umfasst mindestens folgende Schritte:

Data Translation: Die zu integrierenden Datensets sind meist sehr unterschiedlich strukturiert. Das Ziel der *Data Translation* ist die Überführung der Datensets in ein konsistentes Schema.

R. Isele ✉
brox IT-Solutions GmbH, An der Breiten Wiese 9, 30625 Hannover, Deutschland
e-mail: mail@robertisele.com

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_5

Abb. 5.1 Woody Allen in Freebase und DBpedia

Entity Matching: Der Hauptteil des Datenintegrationsprozesses ist die Verknüpfung verwandter Information aus verschiedenen Datensets. Insbesondere die Identifikation verschiedener Entitäten in unterschiedlichen Datensets, welche dasselbe Realwelt-Objekt repräsentieren, ist zentral für den Integrationsprozess. Das Resultat von *Entity Matching* ist eine Menge von Links, welche Entitäten in unterschiedlichen Datensets verknüpfen.

Data Fusion: Das Ziel von *Data Fusion* ist es Entitäten, die im vorhergehenden Entity Matching Schritt verknüpft wurden, zu einer Entität zu verschmelzen. Im Zuge des Integrationsprozesses sollen Konflikte und Inkonsistenzen in den Daten erkannt und behandelt werden.

Wir motivieren die Notwendigkeit für einen Datenintegrationsprozess am einfachen Beispiel zweier Datensets, die Einträge über identische Personen enthalten: Freebase and DBpedia. Während es sich bei DBpedia[1] [2] um ein RDF Datenset handelt, welches automatisch aus Wikipediaartikeln extrahiert wird, ist Freebase[2] [4] eine vorwiegend manuell erstellte, kollaborative Wissensbasis. Abbildung 5.1 zeigt den Eintrag über Woody Allen in beiden Datensets. Obwohl beide Datensets ähnliche Informationen enthalten, sind manche Informationen nur in einem der beiden Datensets enthalten. Somit kann durch die Integration beider Datensets ein Mehrwert geschaffen werden, indem deren gemeinsames Abfragen ermöglicht wird.

Um diesen Prozess zu veranschaulichen, ist dieser Beitrag wie folgt aufgebaut: Der folgende Abschnitt geht auf das Überführen vorliegender Datensets in ein harmonisiertes Schema ein. Abschnitt 5.3 behandelt ausführlich die Integration der Datensets durch die

[1] Siehe http://dbpedia.org, aufgerufen am 21.03.2014.
[2] Siehe http://www.freebase.com, aufgerufen am 21.03.2014.

Verknüpfung verwandter Ressourcen in verschiedenen Datensets. Anschließend behandelt Abschn. 5.4 Strategien, um die zuvor verknüpften Entitäten zu verschmelzen. Zuletzt wird in Abschn. 5.5 ein Überblick über bekannte Tools für die Integration verschiedener Datensets gegeben. Abschnitt 5.6 schließt das Kapitel mit einer Zusammenfassung ab.

5.2 Data Translation

Die zu integrierenden Datensets können aus zwei Quellen stammen: Zum einen können unternehmensinterne Datensets integriert werden, welche in bestehenden relationalen Datenbanken oder anderen Formaten vorliegen und deshalb zunächst nach RDF übersetzt werden müssen. Zum anderen können unternehmensinterne oder öffentliche Datensets integriert werden, welche bereits in RDF vorliegen. Im ersten Fall können die bestehenden Daten mit Hilfe von Linked Data Mappings in das gewünscht RDF Schema überführt werden. Während unternehmensinterne RDF Datensets oft bereits in einem konsistenten Schema vorliegen, verwenden öffentliche RDF Datensets eine Vielzahl verschiedener Schemata, deren Kontrolle sich dem Konsumenten entzieht, und müssen somit erst in das gewünschte Schema überführt werden. Da bereits auf verschiedene Strategien zum Mapping unterschiedlicher Datensets nach RDF eingegangen wurde, wird Data Translation in diesem Kapitel nicht näher erläutert.

5.3 Entity Matching

Die Mehrzahl der Verfahren, die für das Verlinken von Linked Data Quellen entwickelt wurden, lassen sich in zwei Kategorien aufteilen:

- Automatische Ansätze zielen darauf ab, Entitäten ohne eine vom Benutzer bereitgestellte Konfiguration zu verknüpfen [18, 1]. Dafür identifiziert im Allgemeinen ein Mechanismus Duplikate alleinig basierend auf den Eigenschaften des Datensatzes. Dazu existieren automatische Ansätze, welche vom Datensatz unabhängiges Hintergrundwissen, wie zum Beispiel Wörterbücher, verwenden, um die Genauigkeit der Erkennung zu erhöhen [24, 12].
- Im Gegensatz zu automatischen Ansätzen klassifizieren *regelbasierte Ansätze* jedes Paar von Entitäten als Duplikat oder Nicht-Duplikat, basierend auf domänenspezifischen Verknüpfungsregeln [35]. Hierbei legt die Verknüpfungsregel fest, wie die Ähnlichkeit zwischen zwei Entitäten bestimmt wird.

Da für den Unternehmenseinsatz eine hohe Qualität und Nachvollziehbarkeit der generierten Links essentiell ist, beschränkt sich dieses Kapitel vorwiegend auf regelbasierte Ansätze.

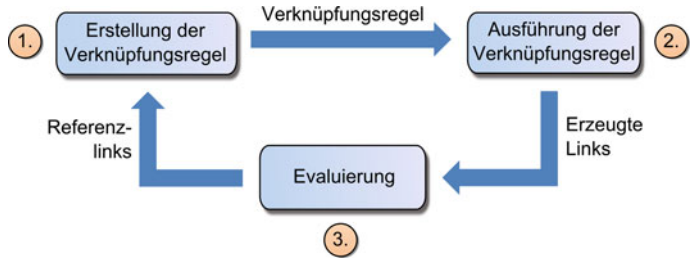


Abb. 5.2 Entity Matching Workflow

5.3.1 Regelbasierter Entity Matching Prozess

Abbildung 5.2 zeigt die wichtigsten Schritte eines regelbasierten Entity Matching Prozesses. Im Folgenden gehen wir für jeden Schritt ins Detail.

Erstellung der Verknüpfungsregel Bevor zwei Datenquellen verlinkt werden können, muss eine Verknüpfungregel definiert werden, welche die Bedingungen angibt, die erfüllt werden müssen, damit ein Link zwischen zwei Entitäten generiert wird. Verknüpfungsregeln können entweder von einem Domänenexperten oder einem Lernalgorithmus erstellt werden. Das Erstellen einer effektiven Verknüpfungsregel von Hand ist ein nicht-triviales Problem und verlangt vom Autor detaillierte Kenntnisse über die Struktur der zu verlinkenden Datensätze. Wir veranschaulichen dies am einfachen Beispiel zweier Datensätze über Filme: Zunächst ist ein Vergleich der Filme alleinig basierend auf ihren Filmtiteln in der Regel nicht ausreichend, da ein Filmtitel unterschiedliche Filme bezeichnen kann, welche z. B. in verschiedenen Jahren veröffentlicht wurden. Daher muss die Verknüpfungsregel zumindest den Filmtitel sowie den Veröffentlichungstermin kombinieren und dabei eine geeignete Aggregationsfunktion wählen, welche die Ähnlichkeiten beider Eigenschaften vereint. In der Regel muss mit Unschärfen und Unregelmäßigkeiten in beiden Datensätzen gerechnet werden. So können die Veröffentlichungsdaten des gleichen Films in verschiedenen Datenquellen um mehrere Tage abweichen. Deshalb muss der Autor der Verknüpfunsregel auch geeignete Distanzmaße zusammen mit adäquaten Schwellwerten auswählen. Die Abdeckung aller in den Datenquellen vorhandener Heterogenitäten durch die Verknüpfungsregel ist oft schwierig und kann meist nur durch mehrere Iterationen erreicht werden.

Ausführung der Verknüpfungsregel Die Ausführung einer Verknüpfungsregel hat das Ziel alle Paare von Entitäten zu identifizieren, für welche die gegebene Verknüpfungsregel erfüllt ist. Das Ergebnis der Ausführung ist eine Menge von Links, worin jeder Link zwei Entitäten verbindet, welche laut Verknüpfungsregel ähnlich sind.

Abbildung 5.3 zeigt einen typischen Ausführungsprozess für zwei Datenquellen. Um zu vermeiden, dass die Verknüpfungsregel für jedes mögliche Paar von Entitäten ausge-

Abb. 5.3 Ausführung einer Verknüpfungsregel

wertet werden muss, kann zunächst eine Indizierung durchgeführt werden. Das Ziel der Indizierung ist es Paare von Entitäten zu identifizieren, welche potentiell ähnlich sind, um dabei andererseits Paare, die definitiv verschieden sind, früh verwerfen zu können. Basierend auf dem Index wird die Verknüpfungsregel anschließend nur für die potentiell ähnlichen Paare von Entitäten ausgewertet.

Verschiedene Indizierungsverfahren wurden entwickelt um die Effizienz des Prozesses zu erhöhen [8, 25, 18, 19]. Allerdings können viele dieser Verfahren dazu führen, dass korrekte Links durch das Indizierungsverfahren fälschlicherweise verworfen werden und damit weniger Links generiert werden als durch die Verknüpfungsregel vorgegeben.

Im vorigen Beispiel zweier Filmdatenbanken könnte eine einfache Indizierungsmethode so aussehen, dass jedem Film basierend auf seinem Erscheinungsjahr ein Index zugewiessen wird. In diesem Fall müssten anschließend lediglich Filme mit dem gleichen Erscheinungsjahr auf ihre Ähnlichkeit mit der Verknüpfungsregel getestet werden. Der Nachteil dieser Methode wäre allerdings der Verlust von Links zwischen Filmen, für welche das falsche Erscheinungsjahr angegeben ist. Die Entwicklung von Indizierungsverfahren, welche die Anzahl der nötigen Vergleiche reduzieren und zugleich den Verlust korrekter Links vermeiden, ist ein wichtiges Forschungsgebiet [7].

Evaluierung Der Zweck des Evaluierungsschrittes ist es, die Qualität der Verknüpfungsregel zu bestimmen und potentielle Fehler in den generierten Links zu finden. Für gewöhnlich erfolgt die Überprüfung basierend auf manuell erstellten Referenzlinks. Eine Menge von Referenzlinks besteht hierbei aus positiven und negativen Referenzlinks. Dabei verknüpfen positive Referenzlinks Paare von Entitäten, welche nachgeprüft das gleiche Real-Welt Objekt beschreiben. Andererseits verbinden negative Referenzlinks Paare von Entitäten, welche verschiedene Real-Welt Objekte beschreiben.

Nach der Ausführung der Verküpfungsregel können mit Hilfe der Referenzlinks zwei Arten von fehlerhaften Links identifiziert werden:

Falsch positive Links: Es wurde ein Link generiert, für den ein negativer Referenzlink existiert.

Falsch negative Links: Für einen positiven Referenzlink wurde kein Link generiert.

Die gefundenen Fehler können genutzt werden, um die Verküpfungsregel iterativ zu verbessern.

5.3.2 Distanzmaße

Die Aufgabe eines Distanzmaßes ist es den Abstand zwischen zwei Zeichenketten zu bestimmen. Ein Distanzmaß gibt einen Wert von 0 zurück, wenn beide Zeichenketten gleichwertig sind, und einen umso größeren Wert, je weiter die Zeichenketten voneinander entfernt sind. Beispielsweise wird ein numerisches Distanzmaß für die beiden Zeichenketten „10“ und „10.0“ eine Distanz von 0 zurückgeben. Handelt es sich bei einem Distanzmaß um ein normalisiertes Maß, werden keine Distanzen größer 1 zurückgegeben.

Analog zu Distanzmaßen ist in der Literatur manchmal auch von Ähnlichkeitsmaßen die Rede. Ähnlichkeitsmaße sind in der Regel normalisiert, d. h. ein Ähnlichkeitsmaß gibt 1 zurück, wenn beide Zeichenketten gleichwertig sind und andernfalls einen Wert kleiner 1.

Zeichenbasierte Distanzmaße Zeichenbasierte Distanzmaße bewerten die Ähnlichkeit zweier Zeichenketten auf der Ebene einzelner Zeichen. Zeichenbasierte Distanzmaße eignen sich gut für den Umgang mit typografischen Fehlern. Als Beispiel für ein häufig verwendetes Distanzmaß führen wir die „Levenshtein Distanz“ ein.

Die *Levenshtein Distanz* [21] (manchmal auch *Editierdistanz* oder engl. *edit distance* genannt) zweier Zeichenketten, ist definiert als die Anzahl von Editieroperationen, die mindestens notwendig sind, um eine Zeichenkette in eine andere zu transformieren. Dafür sind drei Editieroperationen erlaubt:

- **Einfügen** eines einzelnen Zeichens
- **Löschen** eines einzelnen Zeichens
- **Ersetzung** eines Zeichens durch ein beliebig anderes Zeichen.

Wir illustrieren die Berechnung der Levenshtein Distanz an zwei einfachen Beispielen: Die Distanz zwischen „Buch“ and „uch“ beträgt genau 1, denn „Tuch“ kann in „uch“ transformiert werden, indem das erste Zeichen gelöscht wird. Analog beträgt die Distanz zwischen „Buch“ and „Tuch“ ebenfalls genau 1, weil die Ersetzung eines Zeichens ausreichend ist um die Zeichenketten ineinander überzuführen.

Tokenbasierte Distanzmaße Während sich zeichenbasierte Distanzmaße gut für Schreibfehler eignen, erzielen sie schlechte Ergebnisse, wenn die Reihenfolge der Wörter unterschiedlich ist. Werden beispielsweise Personennamen verglichen, verhindert eine Änderung der Reihenfolge des ersten und des letzten Namens (z. B. „John Doe“ und „Doe, John“) oder das Hinzufügen eines Titels bei einer zeichenbasierten Abstandsmessung die Identifizierung eines Duplikats.

Die Idee tokenbasierter Distanzmaße ist es, die Zeichenketten zunächst in ihre Wörter zu zerlegen und anschließend beide Mengen auf Wortebene zu vergleichen. Die Distanz basiert also alleinig auf der Ähnlichkeit der Wörter, während ihre Reihenfolge ignoriert wird. Die Zeichenfolge „Herr Max Mustermann“ würde in die Token

{'Herr', 'Max', 'Mustermann'} aufgeteilt werden. Die Methode, die verwendet wird, um die Zeichenketten in Token aufzuteilen, kann unabhängig der eingesetzten tokenbasierten Metrik ausgewählt werden. Während einfache Ansätze die Strings in Tokens an jedem Leerzeichen aufteilen, berücksichtigen fortgeschrittene Ansätze auch Satzzeichen um Fälle wie „Mustermann, Max" präzise zu verarbeiten.

In reinen tokenbasierten Distanzmaßen werden die einzelnen Token auf Gleichheit verglichen, d. h. zwei Token müssen exakt übereinstimmen. Das hat den Nachteil, dass Token mit typografischen Fehlern, zum Beispiel „Turmstrasse" und „Turmstraße", als unterschiedliche Token betrachtet werden. Im Folgenden werden deshalb auch hybride Distanzmaße eingeführt, welche tokenbasierte und zeichenbasierte Distanzmaße kombinieren, um auch Duplikate von Tokens mit typografischen Fehlern erkennen zu können.

Hybride Distanzmaße Das Ziel von hybriden Distanzmaßen ist es, die Vorteile zeichenbasierter und tokenbasierter Ansätzen zu kombinieren. Wir motivieren die Notwendigkeit für hybride Distanzmaße durch die Erörterung des Hauptnachteils tokenbasierter Distanzmaße: Während tokenbasierte Distanzmaße gut für Zeichenketten funktionieren, die viele Worte teilen, scheitern sie, wenn Schreibfehler in einzelnen Wörtern vorhanden sind. Zum Beispiel erkennt ein tokenbasiertes Distanzmaß problemlos die Namen „John Doe" und „Doe, John" als gleichwertig an, da beide Zeichenketten die gleichen Wörter enthalten. Auf der anderen Seite wird das gleiche tokenbasierte Distanzmaß einen viel höheren Abstand für das Paar „John Doe" und „Jon Doe" erkennen, da aufgrund eines Schreibfehlers beide Zeichenketten nur ein Token teilen. Hybride Ansätze verwenden zeichenbasierte Maßnahmen, um auch Wörter mit Token zu erkennen, die typografische Fehler enthalten.

Eine Reihe von hybriden Distanzsmaßen wurde bei Naumann et al. vorgeschlagen [25]. Ein experimenteller Vergleich von populären hybriden, tokenbasierten und zeichenbasierten Distanzmaßen wurde durch Cohen et al. [10] durchgeführt mit dem Ergebnis, dass das hybride Monge-Elkan Distanzmaß das beste Ergebnis auf den verwendeten Testdaten erzielte.

Phonetische Distanzmaße Die Idee phonetischer Distanzmaße ist es, beim Vergleich von Zeichenketten die Aussprache der Zeichen zu berücksichtigen. Zu diesem Zweck werden bei phonetischen Distanzmaßen die Zeichenketten nicht direkt verglichen, sondern zunächst normalisiert. Im Zuge der Normalisierung werden Zeichen, die ähnlich ausgesprochen werden, durch ein einheitliches Zeichen ersetzt. Zum Beispiel, können die Buchstaben „t" und „d", in Sprachen, in welchen diese ähnlich ausgesprochen werden, durch das Zeichen „t" ersetzt werden. Der resultierende Distanzwert wird anschließend durch den Vergleich der normalisierten Zeichenketten berechnet. Weil die Aussprache verschiedener Zeichen zwischen Sprachen variiert, werden phonetische Distanzmaße meist für eine bestimmte Sprache optimiert.

Eines der bekanntesten phonetischen Distanzmaße ist das *Soundex* [30, 31] Verfahren. Ursprünglich wurde Soundex entwickelt, um Personennamen im United States Census zu vergleichen. Aus diesem Grund ist es für englische Wörter optimiert. Im Laufe der

Zeit wurden allerdings Varianten für andere Sprachen vorgeschlagen. Eine detaillierte Beschreibung des Soundex-Distanzmaßes und ein Überblick über andere weit verbreitete phonetische Distanzmaße kann in [14] nachgeschlagen werden.

Numerische Distanzmaße Die meisten Distanzmaße, die bisher vorgestellt wurden, sind weitgehend unabhängig vom spezifischen Datenformat der zu vergleichenden Zeichenketten. Da aber traditionelle zeichenbasierte Distanzmaße oft schlechte Ergebnisse für numerische Werte erzielen, existiert darüber hinaus eine Reihe von Distanzmaßen, die für Zeichenketten mit numerischen Werten optimiert sind. Der Grund für das schlechte Abschneiden zeichenbasierter Distanzmaße liegt darin, dass Zahlen, die nahe beieinander liegen, nicht notwendigerweise ein Zeichen teilen. Zum Beispiel würden die Zahlen 999,9 und 1.000,0 von einem zeichenbasierten oder tokenbasierten Distanzmaß nicht als ähnlich erkannt. Außerdem existieren unterschiedliche Darstellungen für die gleiche Zahl (z. B. Dezimalschreibweise oder wissenschaftliche Notation). Numerische Ähnlichkeitsmaße decken diese Fälle ab, indem sie zunächst die Zahlen normalisieren.

Basierend auf numerischen Distanzmaßen können Maße für weitere Datentypen definiert werden [8]:

Datumswerte: Datumswerte können verglichen werden, indem jedes Datum zunächst in die Komponenten Tag, Monat und Jahr aufgeteilt wird. Anschließend kann die Distanz in Tagen errechnet werden.

Zeitwerte: Analog zu Datumswerten können Zeitwerte zunächst segmentiert und anschließend als Sekundenwerte vergleichen werden.

Geographische Koordinaten: Distanzen zwischen geographischen Koordinaten können basierend auf ihren Längengraden und Breitengraden berechnet werden.

5.3.3 Verknüpfungsregeln

Im vorigen Abschnitt haben wir verschiedene Distanzmaße eingeführt, um die Ähnlichkeit einzelner Zeichenketten zu bestimmen. So kann in einer Personendatenbank die *Levenshtein Distanz* verwendet werden, um die Ähnlichkeit von Personennamen zu bestimmen, während ein Distanzmaß für Datumswerte verwendet werden kann, um die Ähnlichkeit von Geburtsdaten zu beurteilen. Darüber hinaus können zusätzliche Eigenschaften, wie z. B. die Adresse, der Arbeitgeber, der Ehepartner usw., für den Vergleich herangezogen werden. Um die Ähnlichkeit zweier Entitäten zu bestimmen, ist es oft notwendig, dass mehrere dieser Einzelvergleiche kombiniert werden.

Die Aufgabe einer *Verknüpfungsregel* ist es, einem gegebenen Paar von Entitäten einen Ähnlichkeitswert zuzuweisen. Um den Ähnlichkeitswert zu bestimmen, verbindet eine Verknüpfungsregel einzelne Vergleiche. Im Laufe der Zeit wurden verschiedene Methoden vorgeschlagen die einzelne Distanzen zu einem Ähnlichkeitswert zu kombinieren. Im

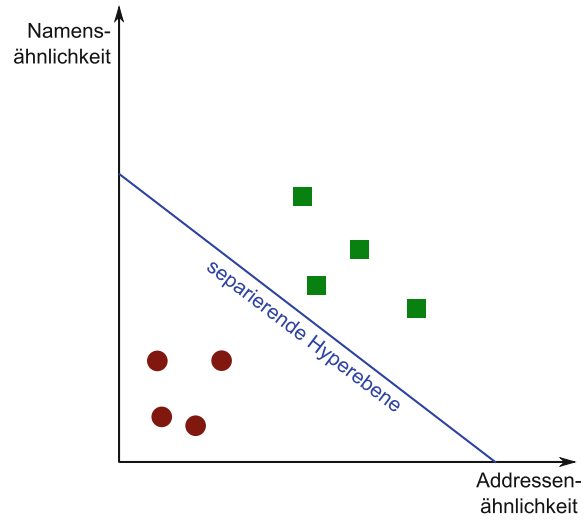


Abb. 5.4 Beispiel eines linearen Klassifikators zur Verlinkung von Personen. Quadrate repräsentieren Beispiele für positive Links, während Kreise Beispiele darstellen, für die kein Link generiert wird

Folgenden geben wir einen Überblick über die meistverwendeten Methoden. Eine umfassendere Übersicht verschiedener Modelle kann in [8] nachgelesen werden.

Lineare Klassifikatoren Ein *linearer Klassifikator* kombiniert mehrere Distanzen, indem die gewichtete Summe der einzelnen Distanzen berechnet wird [11]. Die Größe der Gewichte reguliert hierbei den Einfluss eines bestimmten Vergleichs auf den globalen Distanzwert.

Ein linearer Klassifikator ist definiert als:

► **Definition 5.1 (Linearer Klassifikator)** Gegeben ein Vektor von Distanzen $\vec{d}$ und ein Vektor von Gewichten $\vec{w}$, berechnet sich die Gesamtdistanz als:

$$L_{\text{linear}}(\vec{d}, \vec{w}) = \vec{d} \cdot \vec{w} = \sum_j w_j d_j$$

Wird der Vektor der Distanzen in einen mehrdimensionalen Raum eingetragen, spannt der Vektor der Gewichte bildlich eine Ebene auf. Abhängig davon, auf welcher Seite der Ebene sich der Distanzvektor im Raum befindet, wird der Datenpunkt entweder als Link oder als Nicht-Link klassifiziert.

Ein Beispiel für einen linearen Klassifikator zur Verlinkung von Personen ist in Abb. 5.4 dargestellt. In diesem Beispiel wird jedes Personenpaar durch zwei Ähnlichkeitswerte verglichen: Die Ähnlichkeit der Personennamen sowie die Ähnlichkeit ihrer Kontaktadressen. Quadrate repräsentieren Beispiele für Personenpaare, für die der Klassifikator einen Link generiert, während Kreise Beispiele darstellen, für die kein Link generiert wird.

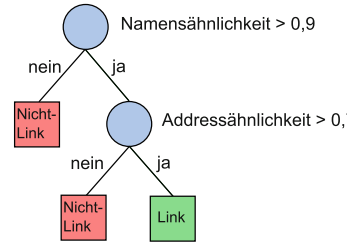


Abb. 5.5 Beispiel eines schwellwertbasierten Klassifikators zur Verlinkung von Personen

Schwellwertbasierte Klassifikatoren Ein *schwellwertbasierter Klassifikator* kombiniert verschiedene Ähnlichkeitstests mit logischen Operatoren [22]. Ein Ähnlichkeitstest besteht dabei aus einem Distanzmaß und einem Schwellwert. Unterschiedliche Modelle schwellwertbasierter Klassifikatoren unterscheiden sich in der Menge von logischen Operatoren, die verwendet werden können. Während das ursprüngliche Modell von [22] nur Konjunktionen (logisches *und*) zulässt, existieren Erweiterungen, welche zusätzlich Disjunktionen (logisches *oder*) und Negationen erlauben [8].

Wir formalisieren nun das ursprüngliche Modell für schwellwertbasierte Klassifikatoren:

► **Definition 5.2 (Schwellwertbasierter Klassifikator)** Gegeben ein Vektor von Distanzwerten $\vec{s}$ und ein Vektor von Schwellwerten $\vec{t}$, berechnet sich die Gesamtdistanz als:

$$L_{\text{schwellwert}}(\vec{s}, \vec{t}) = \bigwedge_j (s_j \geq t_j)$$

Abbildung 5.5 zeigt ein einfaches Beispiel eines schwellwertbasierten Klassifikators zur Verlinkung von Personen.

Expressivere Repräsentationen Während lineare und schwellwertbasierte Klassifikatoren die meistverbreiteten Modelle sind, um Verknüpfungsregeln zu repräsentieren, wurden auch expressivere Modelle vorgeschlagen. Expressive Modelle können klassische lineare und schwellwertbasierte Klassifikatoren in verschiedenen Aspekten erweitern. Beispielsweise kann ein erweitertes Modell auch Datentransformationen erlauben, um Werte vor einem Vergleich normalisieren.

Abbildung 5.6 zeigt eine Verknüpfungsregel für die Verlinkung von geographischen Orten, welche Datentransformationen enthält. In diesem Beispiel vergleicht die Verknüpfungsregel die Namen sowie die geographischen Koordinaten der Entitäten. Die Namen werden zunächst normalisiert, indem sie in Kleinschreibung konvertiert werden. Der anschließende Vergleich erlaubt eine maximale Levenshtein-Distanz der Namen von 1. Die geographischen Koordinaten der beiden Orte können höchstens 10 Kilometer voneinander entfernt sein. Die beiden Schwellwerte normalisieren die Distanzen jeweils auf einen Wert im Intervall $[0, 1]$. Die resultierenden Ähnlichkeitswerte werden dann zu einem einzigen Wert aggregiert. Die im Beispiel verwendete Minimumsaggregation wählt dafür

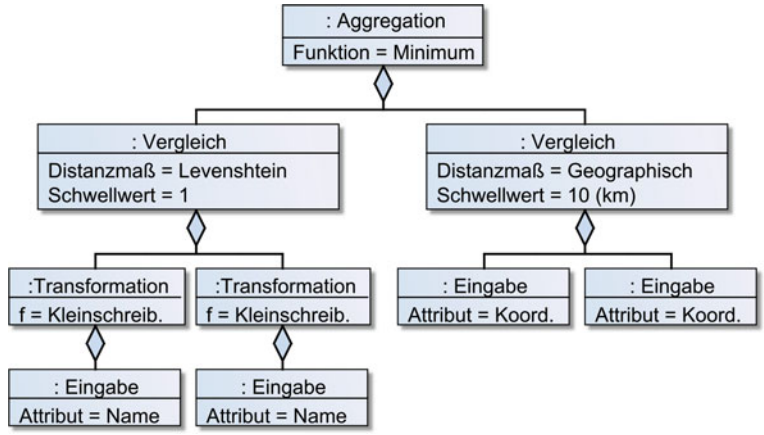


Abb. 5.6 Einfache Verknüpfungsregel für die Verlinkung von geographischen Orten

den minimalen Ähnlichkeitswert aus, d. h. beide Werte müssen über dem globalen Ähnlichkeitsschwellwert liegen, damit ein Link generiert wird.

5.3.4 Evaluierung

In diesem Abschnitt werden die gängigsten Evaluierungsmethoden eingeführt, um die Qualität einer Verknüpfungsregel zu bewerten. Basierend auf einer Menge von positiven und negativen Referenzlinks können zwei Arten von Fehlern auftreten:

- **Fehler 1. Art:** Zwischen zwei Entitäten, für die ein negativer Referenzlink existiert, wurde ein Link generiert.
- **Fehler 2. Art:** Zwischen zwei Entitäten, für die ein positiver Referenzlink existiert, wurde kein Link generiert.

Basierend auf diesen zwei Fehlerarten, kann zwischen vier Fällen unterschieden werden:

	Klassifizierung	
	Link	Kein Link
Positiver Referenzlink	Richtig Positiv (tp)	Falsch Negativ (fn)
Negativer Referenzlink	Falsch Positiv (fp)	Richtig Negativ (tn)

Indem jeder Referenzlink basierend auf der Klassifizierung der Verknüpfungsregel einer der vorgehenden vier Klassen zugeordnet wird, können basierend auf der Anzahl der Links in jeder Klasse sowohl die Genauigkeit als auch die Trefferquote definiert werden.

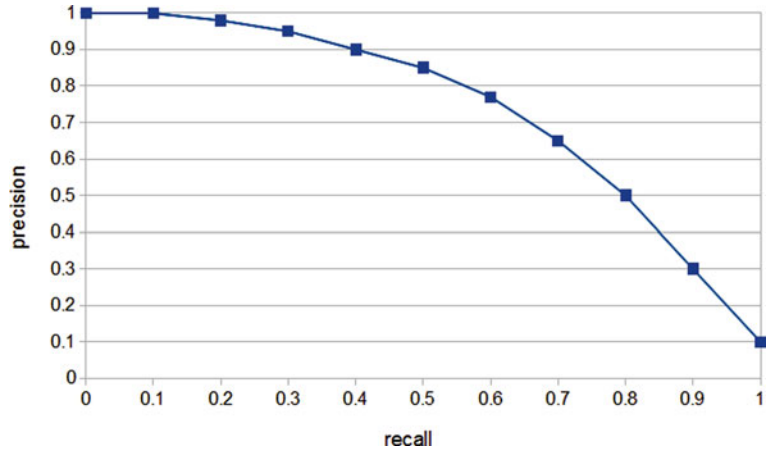


Abb. 5.7 Typisches Recall-Precision Diagramm

Zuerst definieren wir die Genauigkeit einer Verknüpfungsregel:

► **Definition 5.3 (Genauigkeit)** Die Genauigkeit (engl. precision) einer Verknüpfungsregel ist definiert als der Anteil der generierten Links, die korrekt sind:

$$precision = \frac{tp}{tp + fp}$$

Während die Genauigkeit ein Maß für die Korrektheit der Verknüpfungsregel ist, handelt es sich bei der Trefferquote um ein Maß der Vollständigkeit:

► **Definition 5.4 (Trefferquote)** Die Trefferquote (engl. recall) einer Verknüpfungsregel ist definiert als der Anteil der generierten korrekten Links aus allen positiven Referenzlinks:

$$recall = \frac{tp}{tp + fn}$$

In vielen Fällen existiert ein Trade-Off zwischen der Maximierung der Genauigkeit auf der einen Seite und einer Erhöhung der Trefferquote auf der anderen Seite. Eine Erhöhung der Genauigkeit geht oft einher mit einer Reduktion der Trefferquote. Umgekehrt kann eine Erhöhung der Trefferquote zu mehr falschen positiven Links führen und somit die Genauigkeit reduzieren.

Der Zusammenhang zwischen Genauigkeit und Trefferquote für eine bestimmte Verknüpfungsregel kann mit einem Recall-Precision Diagramm visualisiert werden. Abbildung 5.7 zeigt eine typische Verlaufskurve.

Das F-Measure kombiniert die Genauigkeit und die Trefferquote zu einem Wert:

► **F-Measure** Das F-Measure einer Verknüpfungsregel ist definiert als das harmonische Mittel seiner Genauigkeit und seines Trefferwertes:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

5.3.5 Lernverfahren

In Abschn. 5.3.3 haben wir bereits gezeigt, wie Verknüpfungsregeln verwendet werden können, um die genauen Bedingungen zu definieren, die erfüllt sein müssen, damit zwei Entitäten verlinkt werden. Allerdings ist das Schreiben und die Feinoptimierung solcher Verknüpfungregeln von Hand oft schwierig und zeitaufwendig:

1. Der Autor muss diskriminierende Eigenschaften der zu vergleichenden Entitäten identifizieren und für jeden Vergleich ein geeignetes Distanzmaß auswählen.
2. Weil der Vergleich zweier Entitäten an Hand einer einzigen Eigenschaft in der Regel nicht ausreicht, um zu entscheiden, ob beide Entitäten das gleiche Objekt der realen Welt beschreiben, müssen Verknüpfungsregeln gemeinhin mehrere Eigenschaftsvergleiche mit einer geeigneten Funktion zu einem Einzelwert aggregieren.
3. Abhängig vom gewählten Modell müssen geeignete Distanzschwellwerte und/oder Gewichte gewählt werden.

Eine Möglichkeit diesen Aufwand zu reduzieren, stellen überwachte Lernverfahren dar, welche Verknüpfungsregeln aus bestehenden Referenzlinks generieren. Das Erstellen von Referenzlinks ist einfacher als das Schreiben von Verknüpfungsregeln, da es keine Vorkenntnisse über verschiedene Distanzmaße oder das durch das vorliegende Entity Matching System verwendete Modell für die Repräsentation von Verknüpfunsregeln erfordert. Referenzlinks können von Domain-Experten erstellt werden, indem sie die Gleichwertigkeit einer Menge von Entitätenpaaren aus den Datensätzen bestätigen oder ablehnen.

Lineare Klassifikatoren Ein linearer Klassifikator kombiniert eine Menge von Ähnlichkeitsvergleichen durch die Berechnung der gewichteten Summe der Einzelwerte [11]. Das Grundkonzept linearer Klassifikatoren wurde in Abschn. 5.3.3 eingeführt. Zwei verschiedene Ansätze haben sich für das Lernen von linearen Klassifikatoren etabliert [18]: Naive-Bayes-Klassifikatoren und Support-Vektor-Maschinen.

Da das ursprüngliche statistische Modell für das Verlinken unterschiedlicher Datensätze in Datenbanken (engl. record linkage) von Fellegi-Sunter [15] auf Bayes-Statistik beruht, wurden naive-Bayes-Klassifikatoren schon frühzeitig angewendet um Verknüpfungsregeln zu repräsentieren [36]. Allerdings zeigen empirische Vergleiche, dass Support-Vektor-Maschinen für viele Klassifikationsprobleme bessere Resultate erzielen als naive-

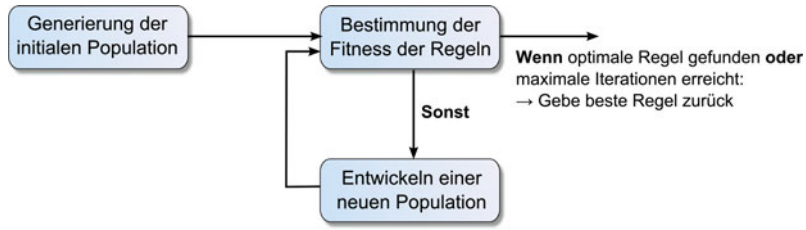


Abb. 5.8 Genetischer Algorithmus

Bayes-Klassifikatoren [6]. Insbesondere sind Support-Vektor-Maschinen geeigneter um Verknüpfungsregeln zu lernen als naive-Bayes-Klassifikatoren [32].

Schwellwertbasierte Klassifikatoren Ein schwellwertbasierter Klassifikator kombiniert mehrere Ähnlichkeitstests mit Hilfe von Booleschen Operatoren [22]. Schwellwertbasierte Klassifikatoren wurden bereits in Abschn. 5.3.3 eingeführt.

Viele bestehende Lernalgorithmen für schwellwertbasierte Klassifikatoren verwenden Entscheidungsbäume [9, 34, 13]. Ein Hauptvorteil von Entscheidungsbäumen ist, dass sie Erklärungen für jede Klassifikation generieren und somit einfach verstanden und manuell verbessert werden können. Unterschiedliche Verfahren zum Lernen von Entscheidungsbäumen können verwendet werden [29]. Beliebte Lernmethoden umfassen CART (Classification and Regression Trees) [5], ID3 [27] und C4.5 [28].

Genetische Programmierung Unter genetischer Programmierung versteht man eine Klasse von Algorithmen des maschinellen Lernens, die auf genetischen Algorithmen basieren und in der Regel die Lösung als Operatorbäume repräsentieren. Genetische Algorithmen erlauben das Lernen von expressiveren Modellen als es mit vorgehenden Lernmethoden möglich ist [20]. Genetische Programmierung ermöglicht es somit Verknüpfungsregeln zu lernen, welche Ähnlichkeitsvergleiche nichtlinear aggregieren oder sogar Datentransformationen enthalten, welche die zu vergleichenden Werte vor dem Vergleich normalisieren [17].

Abbildung 5.8 veranschaulicht den allgemeinen Ablauf eines genetischen Algorithmus. Zu Beginn generiert der genetische Algorithmus eine Population von initialen Verknüpfungsregeln, welche nach einem Zufallsalgorithmus erstellt werden. In jeder Iteration überprüft der genetische Algorithmus die Fitness aller Verknüpfungsregeln in der aktuellen Population, indem für jede Regel überprüft wird, wie viele der aktuellen Referenzlinks korrekt bestimmt werden. Anschließend wird eine neue Population aus der bestehenden erzeugt, indem basierend auf einer Selektionsstrategie Regeln mit höherer Fitness aus der bestehenden Population ausgewählt und zu einer neuen Regel kombiniert werden. Dazu werden zwei Hauptoperationen eingesetzt: 1. Mutation modifiziert eine einzelne Regel zufällig; 2. Crossover rekombiniert zwei Regeln zu einer neuen Regel. Der Algorithmus

ist beendet, sobald entweder eine optimale Verknüpfungsregel gefunden wurde oder eine vorgegebene maximale Anzahl von Iterationen erreicht wurde.

5.4 Data Fusion

Das Ziel des Data Fusion Prozesses ist die Verschmelzung von Entitäten, welche das gleiche Real-Welt-Objekt bezeichnen. Die Verschmelzung basiert auf den Links, die durch den vorhergehenden Entity Matching Prozess erzeugt wurden. Data Fusion verfolgt zwei Ziele:

Vollständigkeit: Durch das Verschmelzen der Attribute aller verlinkten Entitäten werden die Informationen aus verschiedenen Datensätzen kombiniert und damit die Vollständigkeit der Daten erhöht.

Konsistenz: Enthalten verlinkte Datensätze überlappende Informationen über die gleiche Entität, enthalten zum Beispiel zwei Personendatenbanken die Geburtsdaten für jede Person, so können diese Informationen verschmolzen werden. Sind die Informationen der zu integrierenden Datenbanken entweder unvollständig oder teilweise fehlerhaft, werden Strategien für die Konfliktauflösung notwendig.

In der Praxis wird für Linked Data in den meisten Fällen lediglich die Vollständigkeit erhöht, indem Entitäten, die durch sameAs-Links verbunden sind, durch sameAs-Inferenz verschmolzen werden. Dagegen existieren bisher nur wenige Tools, welche das Identifizieren und Auflösen von Konflikten unterstützen.

5.5 Tools

Abschließend soll ein kurzer Überblick über bekannte Tools für die Integration verschiedener Linked Data Quellen gegeben werden.

- Das *Linked Data Integration Framework* (LDIF)[3] [33] unterstützt den vollständigen Datenintergrationsprozess von der Data Translation über das Entity Matching bis zur Data Fusion.
- Das *R2R Framework*[4] [3] unterstützt Data Translation Prozesse. R2R bietet dafür die *R2R Mapping Language* an, welche es ermöglicht benutzerdefinierte Mappings zwischen verschiedenen Schemata auszudrücken.

[3] Siehe http://ldif.wbsg.de/, aufgerufen am 21.03.2014.
[4] Siehe http://r2r.wbsg.de/, aufgerufen am 21.03.2014.

- Das *Silk Link Discovery Framework*[5] [16] und *LIMES*[6] [26] sind Entity Matching Frameworks für Linked Datasets.
- *Sieve*[7] [23] ist ein Linked Data Tool für Quality Assessment und Data Fusion.

5.6 Fazit

In diesem Beitrag wurden die wichtigsten Bestandteile eines Datenintegrationsprozesses im Linked Data Kontext besprochen. Effektive Methoden zur Datenintegration sind eine essentielle Voraussetzung für den Einsatz von Linked Data im Unternehmenskontext.

Der Schwerpunkt dieses Kapitels lag auf der Verknüpfung von Entitäten in verschiedenen Datensets, welche das gleiche Real-Welt-Objekt bezeichnen. Hierzu wurde ausführlich auf populäre Distanzmaße eingegangen, welche benutzt werden können, um die Ähnlichkeit unterschiedlicher Werte zu bestimmen. Aufbauend auf einzelnen Distanzmaßen wurden verschiedene Modelle für Verknüpfungsregeln eingeführt, welche mehrere Distanzmaße kombinieren, um die Ähnlichkeit zweier Ressourcen an Hand ihrer Werte zu bestimmen. Abschließend wurden Lernverfahren eingeführt, die verwendet werden können, um Verknüpfungsregeln basierend auf zuvor erstellten Referenzlinks automatisch zu generieren.

Während im Kontext von Linked Data bereits eine Reihe von Verfahren existieren, um Entitäten, welche das gleiche Real-Welt-Objekt vergleichen, durch das Setzen von Links zu verknüpfen, existierten bisher nur wenige Verfahren, um die dadurch identifizierten Äquivalenzmengen zu einer einzigen Ressource zu verschmelzen. Strategien zur Data Fusion von Linked Data stellen somit ein wichtiges Feld zukünftiger Forschung und Entwicklung dar.

Literatur

1. Aguirre, J.L., B. Cuenca Grau, K. Eckert, J. Euzenat, A. Ferrara, R.W. van Hague, L. Hollink, E. Jimenez-Ruiz, C. Meilicke, A. Nikolov, D. Ritze, F. Scharffe, P. Shvaiko, O. Sváb-Zamazal, C. Trojahn, und B. Zapilko. 2012. Results of the Ontology Alignment Evaluation Initiative 2012. In *Proceedings of the Seventh International Workshop on Ontology Matching (OM)*, 73–115
2. Bizer, C., J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, und S. Hellmann. 2009. DBpedia – a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(3): 154–165
3. Bizer, C., und A. Schultz. 2010. The r2r framework: Publishing and discovering mappings on the web. In *Proceedings of the First International Workshop on Consuming Linked Data*

[5] Siehe http://silk.wbsg.de/, aufgerufen am 21.03.2014.
[6] Siehe http://aksw.org/Projects/LIMES.html, aufgerufen am 21.03.2014.
[7] Siehe http://sieve.wbsg.de/, aufgerufen am 21.03.2014.

4. Bollacker, K., C. Evans, P. Paritosh, T. Sturge, und J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1247–1250
5. Breiman, L., J. Friedman, C.J. Stone, und R.A. Olshen. 1984. *Classification and regression trees*. Chapman & Hall/CRC
6. Caruana, R., und A. Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, 161–168
7. Christen, P. 2011. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering* 24(9): 1537–1555
8. Christen, P. 2012. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer
9. Cochinwala, M., V. Kurien, G. Lalk, und D. Shasha. 2001. Efficient data reconciliation. *Information Sciences* 137(1): 1–15
10. Cohen, W.W., P. Ravikumar, und S.E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the Workshop on Information Integration on the Web*, 73–78
11. Dey, D., S. Sarkar, und P. De. 1998. Entity matching in heterogeneous databases: A distance based decision model. In *Proceedings of the 31st Annual Hawaii International Conference on System Sciences*, 305–313
12. Doan, A., Y. Lu, Y. Lee, und J. Han. 2003. Profile-based object matching for information integration. *IEEE Intelligent Systems* 18(5): 54–59
13. Elfeky, M.G., V.S. Verykios, und A.K. Elmagarmid. 2002. TAILOR: A record linkage toolbox. In *Proceedings of 18th International Conference on Data Engineering*, 17–28
14. Elmagarmid, A.K., P.G. Ipeirotis, und V.S. Verykios. 2007. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering* 19(1): 1–16
15. Fellegi, I.P., und A.B. Sunter. 1969. A theory for record linkage. *Journal of the American Statistical Association* 64(328):1183–1210
16. Isele, R. 2013. Learning Expressive Linkage Rules for Entity Matching using Genetic Programming. Ph.D. thesis, University of Mannheim
17. Isele, R., und C. Bizer. 2012. Learning expressive linkage rules using genetic programming. *Proceedings of the VLDB Endowment (PVLDB)* 5(11): 1638–1649
18. Köpcke, H., und E. Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69(2): 197–210
19. Köpcke, H., A. Thor, und E. Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment* 3(1-2): 484–493
20. Koza, J.R. 1993. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press
21. Levenshtein, V.I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8): 707–710
22. Lim, E.P., J. Srivastava, S. Prabhakar, und J. Richardson. 1993. Entity identification in database integration. In *Proceedings of the Ninth International Conference on Data Engineering*, 294–301

23. Mendes, P.N., H. Mühleisen, und C. Bizer. 2012. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, 116–123. ACM

24. Michalowski, M., S. Thakkar, und C.A. Knoblock. 2004. Exploiting secondary sources for unsupervised record linkage. In *Proceedings of the VLDB Workshop on Information Integration on the Web*, 34–39

25. Naumann, F., und M. Herschel. 2010. *An Introduction to Duplicate Detection*. Morgan & Claypool

26. Ngomo, A.C.N., und S. Auer. 2011. Limes: a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, 2312–2317. AAAI Press

27. Quinlan, J.R. 1986. Induction of decision trees. *Machine Learning* 1(1): 81–106

28. Quinlan, J.R. 1993. *programs for machine learning*. Morgan Kaufmann Publishers

29. Rokach, L., und O.Z. Maimon. 2008. *Data mining with decision trees: theory and applications*. World Scientific Publishing Company Incorporated

30. Russell, R. April 1918. Index, United States patent 1261167

31. Russell, R. November 1922. Index, United States patent 1435663

32. Sarawagi, S., und A. Bhamidipaty. 2002. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 269–278

33. Schultz, A., A. Matteini, R. Isele, P.N. Mendes, C. Bizer, und C. Becker. 2012. LDIF – a framework for large-scale linked data integration. In *21st International World Wide Web Conference, Developers Track*

34. Tejada, S., C.A. Knoblock, und S. Minton. 2001. Learning object identification rules for information integration. *Information Systems* 26(8): 607–633

35. Winkler, W.E. 1995. Matching and record linkage. In *Business Survey Methods*, Hrsg. B.G. Cox, D.A. Binder, B.N. Chinnappa, A. Christianson, M.J. Colledge, P.S. Kott, 355–384. John Wiley & Sons

36. Winkler, W.E. 2002. Methods for record linkage and bayesian networks. Tech. rep., Series RRS2002/05, U.S. Bureau of the Census

Linked Data Kuratierung und Visualisierung mit semantischen Daten Wikis

6

Philipp Frischmuth, Michael Martin, Sebastian Tramp und Sören Auer

Zusammenfassung

Die visuelle Aufbereitung von Linked Data sowohl für Zwecke der Prozessverarbeitung als auch zur Konsumierung durch Endanwender ist ein wichtiges Designelement in der unternehmerischen Aneignung von Semantic Web Technologien. Der Beitrag stellt aktuelle Ansätze in der Linked Data-Visualisierung vor, diskutiert deren Bedeutung im Enterprise Information Management und zeigt Beispiele, wie durch Best Practices unternehmerische Nutzungsszenarien von Linked Data unterstützt werden können.

6.1 Einleitung

Trotz der inzwischen großen Popularität von Linked Data ist es immer noch schwierig für Endanwender mit Linked Data zu interagieren. Es gibt inzwischen eine große Menge von strukturierten Daten, die entsprechend dem RDF-Datenmodell im Web publiziert werden. Die Visualisierung, Bearbeitung und Kuratierung dieser Daten ist für Endanwender jedoch nach wie vor kompliziert. Auch ist das entstehende Daten-Web derzeit noch eher ein schreibgeschütztes „Read-only Web“ statt ein „Read-Write Web“ – die überwiegende Menge der strukturierten Informationen wird nur durch eine relativ kleine Menge von Autoren bereitgestellt [10]. Ein Ansatz zur Realisierung eines semantischen „Read-Write Webs“, bei dem viele Nutzer auf einfache Weise kleine Beiträge leisten können, sind semantische Wikis.

P. Frischmuth ✉ · M. Martin · S. Tramp
Institut für Informatik, AKSW, Universität Leipzig, Augustusplatz 10, 04009 Leipzig, Deutschland
e-mail: frischmuth@informatik.uni-leipzig.de
S. Auer
Enterprise Information Systems, Universität Bonn & Fraunhofer IAIS, 53117 Bonn, Deutschland

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_6

Tab. 6.1 Konzeptuelle Unterschiede zwischen Semantic MediaWiki und OntoWiki

	Semantic MediaWiki	OntoWiki
Basis Entitäten	Artikel	Ressourcen
Bearbeitung	Wiki markup	Formulare
Atomare Elemente	Text blob	Statements

Von Semantic Wikis zu Semantic Data Wikis Semantic Wikis sind eine Erweiterung von herkömmlichen, textbasierten Wikis. Während bei herkömmlichen Wikis ein spezielles Wiki-Markup für die Strukturierung der Seiteninhalte genutzt wird, zielen semantische Wikis auf die Anreicherung der Texte mit maschinenlesbaren semantischen Strukturen ab. Dazu haben sich zunächst zwei orthogonale Ansätze ausgebildet: a) die Erweiterung der Markup-Sprache, um semantische Annotationen und Links mit Bedeutung zuzulassen und b) das direkte Aufsetzen der Wiki-Software auf strukturierten Informationen. Heute haben sich beide Ansätze etwas angenähert, zum Beispiel bietet *Semantic MediaWiki* [16] auch Formulare für die Eingabe von strukturierten Daten. Für die zwei prototypischen Vertreter beider Ansätze – Semantic MediaWiki stellvertretend für (a) und OntoWiki stellvertretend für (b) – wurden die zentralen Merkmale in Tab. 6.1 gegenübergestellt.

Wikis für die Bearbeitung strukturierter Daten Im Gegensatz zu textbasierten Systemen basieren Wikis für strukturierte Daten – auch Daten-Wikis – auf einem normierten Daten-Modell. Die Wiki-Software kann verwendet werden, um Instanzen entsprechend einem bestimmten Daten-Schema hinzuzufügen und (in einigen Systemen) auch zur Modifikation des Daten-Schemas selbst. OntoWiki, als Vertreter dieser Klasse, basiert direkt auf dem RDF-Datenmodell. Auf diese Weise werden sowohl Schema- als auch Instanz-Daten im gleichen Low-Level-Modell (d. h. als Aussagen) repräsentiert und können somit auf gleiche Weise mit dem Wiki bearbeitet werden.

OntoWiki – ein semantisches Daten Wiki OntoWiki begann als RDF-basiertes Daten-Wiki zur Unterstützung verteilter Datenmanagement-Aufgaben und hat sich mittlerweile zu einem umfassenden Framework für die Entwicklung von Semantic Web Anwendungen entwickelt [14]. Dies beinhaltet nicht nur eine umfassende Erweiterungsschnittstelle, die eine Vielzahl von Anpassungen ermöglicht, sondern auch verschiedene Publikations- und Zugriffs-Schnittstellen, sodass OntoWiki-Installationen gleichermaßen als Daten-Anbieter und Daten-Konsument im Web der Daten agieren. OntoWiki ist durch klassische Wiki-Systeme inspiriert, seine Architektur ist jedoch unabhängig und ergänzend zu herkömmlichen Wiki-Technologien. Im Gegensatz zu anderen semantischen Wiki-Ansätzen werden in OntoWiki Textbearbeitung und Knowledge Engineering (d. h. die Arbeit mit strukturierten Daten) getrennt voneinander behandelt. Dabei orientiert sich OntoWiki direkt am Wiki-Paradigma „mache es einfach Fehler zu korrigieren, anstatt zu versuchen Fehler von vornherein zu verhindern“ („making it easy to correct mistakes, rather than making it hard to make them“) [17] und wendet dieses auf kollaborative Bearbeitung von

strukturierten Inhalten an. Dieses Paradigma wird durch die Interpretation von Wissensbasen als *Informationslandkarten* realisiert, in denen jeder Knoten visuell dargestellt und mit verwandten Ressourcen verknüpft wird. Weiterhin ist es möglich das Daten-Schema ebenso wie die zugehörigen Instanz-Daten allmählich zu erweitern. Die folgenden Merkmale kennzeichnen OntoWiki:

Intuitive Anzeige und Bearbeitung der Instanzdaten wird durch generische Methoden bereitgestellt mit der Erweiterungsmöglichkeit um domänenspezifische Darstellungen.
Semantische Sichten erzeugen verschiedene Perspektiven auf die Daten und Aggregationen der Wissensbasis.
Versionierung und Evolution bieten die Möglichkeit Änderungen zu verfolgen, zu bewerten und selektiv rückgängig zu machen.
Semantische Suche ermöglicht eine Volltext-Suche auf allen Daten mit der Möglichkeit Suchergebnisse zu filtern und zu sortieren.
Unterstützung der Zusammenarbeit ermöglicht Diskussionen über kleine Informationseinheiten zu führen und über bestimmte Fakten oder potenzielle Änderungen abzustimmen.
Online Statistiken messen interaktiv die Popularität der Inhalte und Aktivitäten der Nutzer.
Semantische Syndikation unterstützt die einfache Distribution von Daten und deren Integration in Desktop-Anwendungen.

OntoWiki ermöglicht die einfache Erstellung von hochgradig strukturierten Inhalten durch verteilte Communities. Die folgenden Punkte fassen einige Einschränkungen und Schwächen von OntoWiki zusammen und kennzeichnen damit mögliche Anwendungsdomänen:

Umgebung: OntoWiki ist eine Web-Anwendung und kann daher nur in einer verteilten Web-Umgebung genutzt werden.
Anwendungsszenario: OntoWiki fokussiert auf Knowledge-Engineering-Projekte, wo ein präzises Nutzungsszenario entweder zunächst (noch) nicht bekannt oder nicht (leicht) definierbar ist.
Reasoning: Reasoning-Dienste zum Inferieren neuen Wissens und neuer Zusammenhänge werden von OntoWiki nur über externe Dienste bereitgestellt.

Struktur des Kapitels In Abschn. 6.2 geben wir einen Überblick über die Architektur von OntoWiki. Die Visualisierung und Exploration einschließlich der wichtigsten Elemente der Benutzeroberfläche wird in Abschn. 6.3 vorgestellt. In Abschn. 6.4 gehen wir auf die Redaktions- und Content-Management-Funktionalität ein. Ein Anwendungsfall wird in Abschn. 6.5 vorgestellt. Wir schließen mit einem Ausblick auf zukünftige Arbeiten in Abschn. 6.6.

6.2 Architektur

Im folgenden Kapitel beschreiben wir die Architektur sowohl der Wiki Applikation als auch des Frameworks für die Entwicklung von wissensintensiven und agilen Anwendungen. Einen Überblick über die Architektur bietet Abb. 6.1, die im Folgenden detailliert beschrieben wird. Danach gehen wir auf die Schlüssel-Bestandteile *Generic Data Wiki* und *OntoWiki Application Framework* ein, welche in der Übersicht durch einen dicken Rahmen markiert sind.

Die Architektur untergliedert sich in drei Ebenen: *Backend*, *Application* und *Frontend*. Zusätzlich dazu repräsentiert eine durchgezogene vertikale Ebene alle verfügbaren Erweiterungsmöglichkeiten, welche durch Dritte entwickelt werden können: *Plugins*, (komplexere) *Extensions* und spezifische Frontend-Elemente und Services. Sie stellt ein Ökosystem von OntoWiki-Erweiterungen dar.

Die Backend Ebene besteht aus der *Erfurt API*[1] und dem zugrunde liegendem externen *Zend Framework*[2], welches ein stabiles und etabliertes Web Applikations-Framework für PHP ist.

Erfurt ist ein generisches Framework für die Entwicklung von Semantic Web Anwendungen. Erfurt wird parallel zu OntoWiki entwickelt und bietet Entwicklern Zugriff auf RDF Quadrupel (RDF Aussagen + einen Kontext) über API Methoden aber auch über die SPARQL Abfragesprache [13]. Mit Hilfe einer Speicher-Abstraktionsschicht bietet Erfurt Zugriff auf unterschiedliche RDF Triple Stores, wie z. B. *Openlink Virtuoso* [11]. Darüber hinaus können RDF Daten in relationalen Datenbanken gespeichert werden, welche mit Hilfe eines SPARQL-to-SQL-Rewriters angebunden sind. Erfurt bietet außerdem Authentifizierungs- und Access Control Funktionalitäten auf der Basis eines RDF Access Control Schemas, deren Instanzen selbst innerhalb eines Triple Stores gespeichert werden. Weiterhin sind folgende Funktionalitäten in Erfurt enthalten:

- Unterstützung für die Versionierung von RDF Daten,
- ein automatisch invalidierender SPARQL Anfrage Cache [18],
- eine Plugin-Umgebung auf der Basis eines Event-Trigger Systems und
- ein leichtgewichtiger Ressourcen-Abfrage-Mechanismus für den Zugriff und die Konvertierung von beliebigen Ressourcen (beispielsweise um Zugriff auf API Ressource URLs zu gewährleisten).

Auf dem Backend Layer baut das *OntoWiki Application Framework* [14] auf. Dieses nutzt und erweitert sowohl Zend (beispielsweise um die Model-View-Controller Infrastruktur bereitzustellen) als auch die Erfurt API.

Der Frontend Layer besteht aus dem generischen Daten-Wiki, dem RDFa-Form Editor *RDFauthor* und zusätzlichen Zugriffs-Schnittstellen. Das Daten-Wiki arbeitet unabhängig

[1] Siehe http://aksw.org/Projects/Erfurt, aufgerufen am 04.04.2014.
[2] Siehe http://framework.zend.com/, aufgerufen am 04.04.2014.

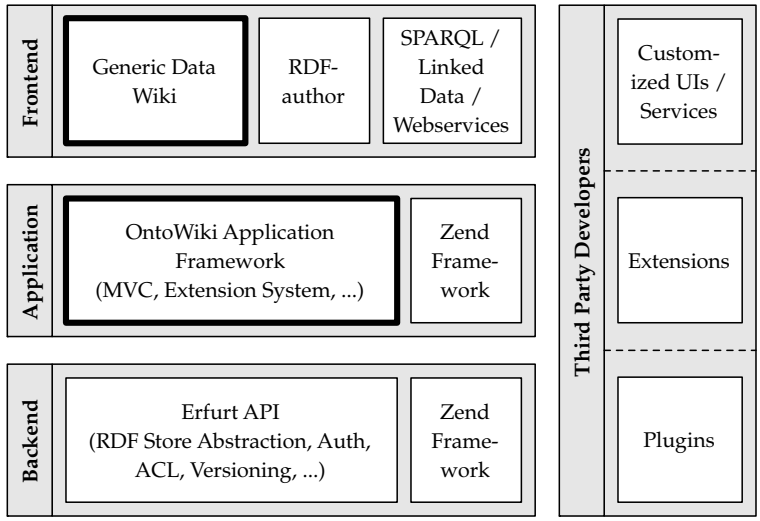


Abb. 6.1 OntoWiki Architektur: Backend, Applikation und Frontend Ebene, zusätzlich die vertikale Ebene für Erweiterungen

von spezifischen Ontologien oder Schemata und kann demzufolge out-of-the-box mit jeder beliebigen Wissensbasis genutzt werden. *RDFauthor* [23] stellt ein vom Daten-Wiki unabhängiges Frontend für eine nutzerfreundliche Editing-Oberfläche für RDF Daten dar. Darüber hinaus wird ein SPARQL Endpunkt [12] und ein Linked Data Endpunkt [6] für maschinellen Zugriff zur Verfügung gestellt.

Externe Entwickler können OntoWiki auf allen drei Ebenen erweitern. Im Backend können Plugins hinzugefügt werden, welche durch spezifische Ereignisse getriggert werden und im Hintergrund spezifische Aktivitäten an den Daten durchführen können. Auf der Applikations-Ebene können gleich mehrere verschiedene Erweiterungsmechanismen genutzt werden, um zusätzliche Funkionen bereitzustellen oder bestehende zu verändern. Auf der Frontend Ebene können darüber hinaus spezielle Darstellungen hinzugefügt werden (beispielsweise für ein spezifisches Vokabular).

6.2.1 Generisches Daten-Wiki

OntoWiki kann out-of-the-box als generisches Werkzeug für die Publikation, Exploration, Bearbeitung und Wartung von beliebigen RDF Wissensbasen verwendet werden. Zu diesem Zweck stellt es generische Methoden und Ansichten zur Verfügung, welche unabhängig von der Domäne der Daten sind und nicht angepasst werden müssen. OntoWiki orientiert sich dazu an folgenden Wiki-Prinzipien [17]:

1. Im Normalfall sind alle Nutzer gleichberechtigt in ihrer Partizipation beim Editieren und Erweitern der Datenbasis. Access Control auf Wissensbasis-Ebene ist zwar möglich, jedoch nicht voreingestellt.
2. Inhalt und Struktur der Wissensbasis können mit den gleichen Methoden und Werkzeugen manipuliert werden. Dies resultiert schon aus dem RDF Daten-Modell, welches Schema- und Instanz-Wissen gleich behandelt.
3. Alle Veränderungen werden versioniert, wodurch es einfach ist Fehler zu beheben.
4. Veränderungen und Aktivitäten können online auf Ressourcen-Ebene diskutiert werden.

OntoWiki ist einzig und allein auf das RDF Daten-Modell ausgerichtet und dadurch konsequent auf strukturierte Informationen fokussiert und nicht auf Texte mit semantischen Annotationen.

Die Detail-Ansicht und die Listen-Ansicht sind zwei allgemeine und Schema-unabhängige Darstellungen, welche im OntoWiki Kern mitgeliefert werden. Die Detail-Ansicht wird verwendet, um eine Beschreibung einer Ressource mit allen bekannten Informationen zu präsentieren. Die Listen-Ansicht stellt eine Anzahl von Ressourcen dar, beispielsweise Instanzen einer bestimmten Klasse. Diese beiden Ansichten sind hinreichend und notwendig um jede beliebige Wissensbasis zu präsentieren. Abbildung 6.2 zeigt exemplarisch sowohl eine Listen- als auch eine Detail-Ansicht.

6.2.2 OntoWiki Application Framework

OntoWiki wurde ursprünglich als generisches Werkzeug für die gemeinsame Arbeit an RDF Wissensbasen konzipiert [3]. Obwohl wir es weiterhin als generisches Daten-Wiki bezeichnen (siehe Abschn. 6.2.1), können die meisten Funktionalitäten unabhängig von der Wiki-Applikation in anderen Anwendungen verwendet werden. Intern werden verschiedene Erweiterungsmechanismen zur Verfügung gestellt, mit denen Entwickler arbeiten können. Tatsächlich ist ein großer Teil des OntoWiki Kernsystems durch Erweiterungen realisiert, auch wenn diese im Normalfall immer mit ausgeliefert und installiert werden. Daher bezeichnen wir OntoWiki häufig auch als *OntoWiki Application Framework (OAF)* [14].

Das OAF besteht prinzipiell aus den Komponenten MVC-Provider (Model-View-Controller), Linked Data Infrastruktur und Erweiterungssystem. Der MVC-Provider stellt sicher, dass alle eingehenden Web-Anfragen auf adäquate Controller mit entsprechenden Funktionalitäten geroutet werden (Erweiterungen können auch Controller zur Verfügung stellen). Die Linked Data Infrastruktur arbeitet auf HTTP Anfrage-Ebene und gewährleistet, dass angefragte Ressourcen (URIs) entsprechend der Anfrage bearbeitet werden. Das heißt es wird in Abhängigkeit vom anfragenden Akteur zwischen verschiedenen Formaten

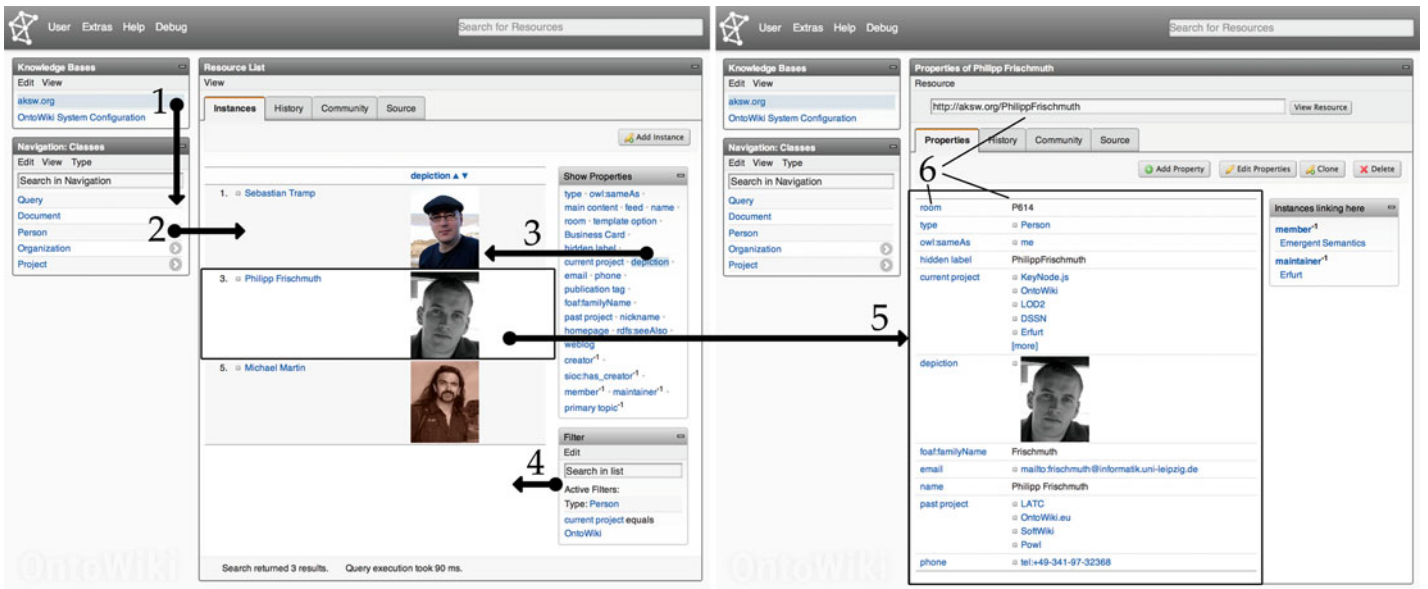


Abb. 6.2 OntoWiki Screenshots mit typischen Arbeitsschritten: 1) Auswahl einer Wissensbasis; 2) Auswahl einer Klasse; 3) Auswahl von zusätzlichen Attributen und Relationen, welche als Spalten in der Tabellen-Ansicht dargestellt werden sollen; 4) Zusätzliche Einschränkung der Ressourcen-Liste durch einen Filter; 5) Auswahl einer Ressource führt den Nutzer zur Detail-Ansicht; 6) Darstellung der RDF Aussagen im User Interface

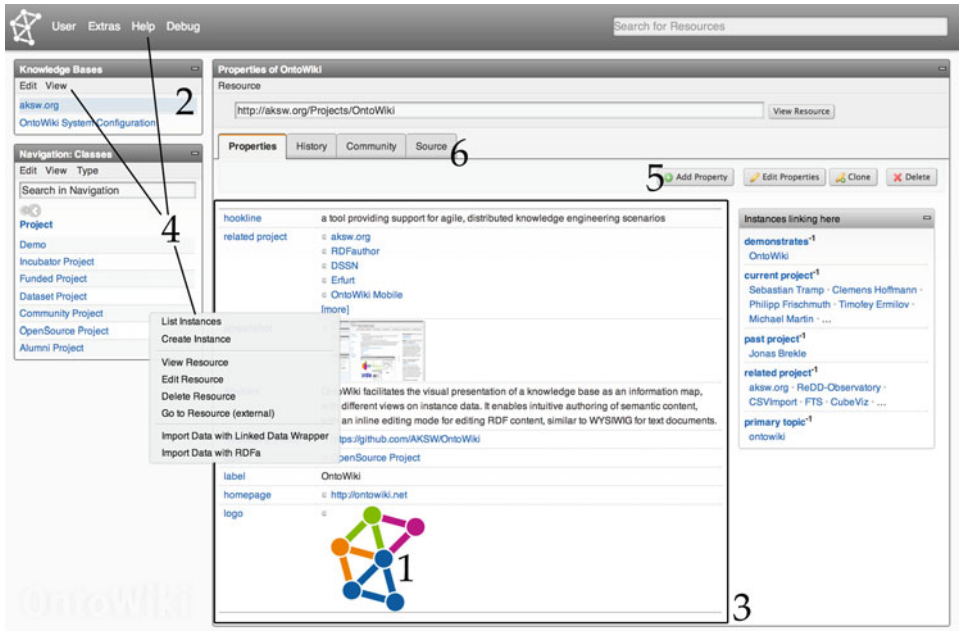


Abb. 6.3 Verschiedene OntoWiki Erweiterungsmöglichkeiten: 1) Bild wird durch ein Plugin zur Verfügung gestellt; 2) Ein Modul-Fenster; 3) Der Haupt-Inhalt wurde von einer Komponente bereitgestellt; 4) Verschiedene Menüs, alle erweiterbar; 5) Erweiterbare Toolbar und 6) Erweiterbares Navigationsmodul

und Visualisierungen unterschieden, auch bekannt als Content Negotiation. So wird ein Nutzer, welcher eine Ressource über den Browser anfragt, eine HTML-Repräsentation bekommen, wohingegen ein Semantic Web Agent, welcher maschinenlesbare Inhalte bevorzugt, z. B. eine XML-Datei erhält.

Das OAF stellt folgende prinzipielle Erweiterungsmöglichkeiten zur Verfügung:

Plug-ins sind die einfachste aber flexibelste Erweiterungsmöglichkeit. Sie bestehen aus beliebigem Code, welcher durch einen bestimmten Event getriggert wird. OntoWiki benutzt Plug-ins beispielsweise zur Präsentation von bestimmten Ressourcen im UI (Videos, Bilder usw. siehe Punkt 1 in Abb. 6.3).

Wrapper sind eine leichtgewichtige Erweiterungsmöglichkeit um RDF Daten für bestimmte Ressourcen zur Verfügung zu stellen. Eine OAF-basierte Applikation, welche Daten von Twitter benutzen möchte, könnte dies sehr einfach mit einem Wrapper ermöglichen, welcher Tweets und Nutzer-Profile ausliest und als SIOC Instanzen [7] ausgibt.

Module sind Erweiterungen, welche kleine Fenster innerhalb des UI darstellen und so sehr einfach zusätzliche Funktionalitäten anbieten können. Die meisten Funktionen von OntoWiki werden durch Module dargestellt (siehe Punkt 2 in Abb. 6.3).

Components sind MVC-Controller, welche Anfragen beantworten können. In den meisten Fällen stellen Komponenten einen Service, den Inhalt von Modulen oder den Hauptinhalt zur Verfügung (siehe Punkt 3 in Abb. 6.3). Eine typische Komponente ist beispielsweise die OntoWiki Karten-Erweiterung[3], welche einen Controller bereitstellt, der Ressourcen auf einer Karte darstellt.

Darüber hinaus gibt es verschiedene weitere Schnittstellen, welche Erweiterungen nutzen können um die Applikation zu manipulieren:

Menüs und Kontext-Menüs existieren über die ganze Applikation verteilt (siehe Punkt 4 in Abb. 6.3) und können durch Erweiterungen beliebig manipuliert werden.

Die Toolbar ist ein zentrales Element, welches eine konsistente UI über alle Views hinweg sichert (siehe Punkt 5 in Abb. 6.3). Über Erweiterungen können hier neue Funktionen hinzugefügt werden.

Die Navigation Bar wird oberhalb des Hauptinhalts dargestellt und dient dem Nutzer zum Umschalten verschiedener Ansichten (siehe Punkt 6 in Abb. 6.3). Erweiterungen können hier eigene Ansichten integrieren und bestehende deaktivieren.

Messages sind Nutzerhinweise, welche einen Text und einen Typ haben (`success`, `info`, `warning`, `error`). Erweiterungen können Messages generieren, welche konzertiert in einem Message-Bereich angezeigt werden.

Zusätzlich zu den genannten Erweiterungsmöglichkeiten ist es möglich das UI grundlegend mit einem angepassten Theme zu verändern und eine eigene Sprach-Lokalisierung zur Verfügung zu stellen. Derzeit existieren Lokalisierungen für Englisch, Deutsch, Russisch und Chinesisch, wobei mittels Erweiterungen weitere Lokalisierungen bereitgestellt werden können.

6.3 Exploration und Visualisierung von RDF-Daten

Zur Untersuchung und Exploration von Wissensbasen können generische aber auch domänenspezifische Werkzeuge zum Einsatz kommen. Je nach Anwender und Aufgabe können die generischen aber auch die spezifischen Werkzeuge vorteilhaft eingesetzt werden. OntoWiki unterstützt diverse Möglichkeiten zur Exploration von RDF-Wissensbasen und ist hochgradig erweiterbar. In Tab. 6.2 ist eine Kategorisierung nach bestehendem Domänenwissen in Kombination mit existierenden Standard-Visualisierungen enthalten.

Im Rahmen der Kategorisierung werden drei Typen von UI-Elementen verwendet, die wie folgt definiert sind:

[3] Siehe https://github.com/AKSW/map.ontowiki, aufgerufen am 04.04.2014.

Tab. 6.2 Kategorisierung anwendbarer UI-Elemente mit Bezug zur Existenz von bestehendem Domänenwissen – Die Erweiterbarkeit von OntoWiki unterstützt dabei alle Ebenen

Ebene	Geeignete UI-Element-Typen	Beispiele
0 (none)	Generisch	Listen, Ressourcenansicht (Abschn. 6.3.1)
1 (partiell)	Normal	Ressourcentitel, Bilder, Weblinks, Karten (Abschn. 6.3.2)
2 (strukturell)	Normal	Hierarchien (Klassen, SKOS, Eigenschaften), statistische Diagramme (Abschn. 6.3.1 und 6.3.2)
3 (voll)	Speziell	Domänenspezifische Seiten (Abschn. 6.3.2) und Formulare

Generisch: UI-Elemente können auf beliebige RDF-Daten angewendet werden.

Normal: UI-Elemente sind in unterschiedlichsten Domänen wiederverwendbar. In Abhängigkeit von der Charakteristik des Elementes wird unterschiedlich detailliertes Wissen über die Daten benötigt.

Speziell: UI-Elemente sind für die Daten einer Domäne maßgeschneidert und können nur auf diese Daten angewendet werden.

Diese UI-Element-Typen stehen in Relation zu den folgenden vier Ebenen über Domänenwissen:

Ebene 0 bedeutet, dass kein Wissen über die Daten vorhanden ist (außer, dass es unter Verwendung von RDF repräsentiert ist). Somit sind nur generische UI-Elemente anwendbar. Betrachtet man OntoWiki als ein generisches Werkzeug, so bietet es zwei wichtige Elemente der Benutzeroberfläche: (a) Ressourcenlisten auf SPARQL-Abfragen und (b) tabellarische Ressourcen-Ansichten.

Ebene 1 erfordert partielles Wissen über die Daten. So enthalten beispielsweise die meisten RDF-Ressourcen einen Titel, der unter Verwendung spezieller RDF-Prädikate kodiert wurde. Sind diese speziellen RDF-Prädikate bekannt, so können Titel statt URIs zur Visualisierung von RDF-Ressourcen verwendet werden (z. B. in einer Ressourcenliste).

Ebene 2 setzt die Existenz von strukturellem Wissen über die Daten voraus. Oft werden Ressourcen unter Verwendung von Klassen gruppiert (z. B. durch Verwendung von RDFS und OWL [19]) oder in Beziehung zu einander gesetzt (z. B. unter Verwendung von SKOS [20]). Ein weiteres Beispiel hierfür sind statistische Daten, die u. a. durch Verwendung des Data-Cube-Vokabulares [9] repräsentiert werden können.

Ebene 3 bedeutet, dass vollständiges Wissen über die Domäne vorhanden ist. Hierfür werden domänenspezifische UI-Elemente zur Verbesserung der Benutzerfreundlichkeit eingesetzt.

OntoWiki bietet mit seiner Erweiterbarkeit vollständige Unterstützung auf allen vier Ebenen durch Bereitstellung von angemessenen UI-Elementen (entweder eigens bereitgestellte oder durch Dritte). Im weiteren Verlauf des Kapitels werden die verschiedenen OntoWiki-Funktionen zur Exploration beschrieben.

6.3.1 Basiselemente der Benutzerschnittstelle

OntoWiki offeriert zwei Basiselemente in der Benutzeroberfläche zur generischen Visualisierung von RDF-Daten:

- **Listenansicht:** Zur Anzeige von Suchergebnissen sowie die
- **Ressourcenansicht:** Zur Anzeige einzelner RDF-Ressourcen.

Beide Ansichten wurden schon kurz in Abschn. 6.2 erwähnt bzw. in Abb. 6.2 dargestellt.

Generische SPARQL-basierende Listen

In OntoWiki sind Ressourcenlisten zentrale UI-Elemente. In den meisten Fällen werden diese verwendet, um Ergebnisse aus einer der folgenden Funktionseinheiten zu präsentieren:

1. Stichwortsuche,
2. Selektion einer RDF-Ressource aus dem Navigationsmodul sowie die
3. Konstruktion und Verwendung einer SPARQL-Abfrage.

In den ersten beiden Fällen wird eine SPARQL-Selektionsabfrage als Grundlage verwendet, die angemessene Filter-Konditionen enthält. Wenn eine SPARQL-Selektionsabfrage direkt vom Benutzer eingegeben wurde, sind zudem zusätzliche Bedingungen zu erfüllen. So müssen beispielsweise in der ersten Spalte der Ergebnisliste URIs enthalten sein, um automatisiert entsprechende Ressourcentitel aufzulösen und die Ergebnistabelle benutzerfreundlich aufzubereiten. Zudem muss die Abfrage weiteren Anforderungen entsprechen, so dass diese durch weitere Komponenten modifiziert werden kann. Somit wird ein sukzessives und facettiertes Filtern der Ergebnisliste an der Oberfläche ermöglicht.

In Abb. 6.4 sind alle beeinflussenden OntoWiki-Komponenten dargestellt, die zur Erzeugung einer als HTML-Tabelle generierten Ergebnisliste beitragen. Initial obliegt dem Administrator die Zugriffsbeschränkung auf verwaltete RDF-Modelle. Existieren Zugriffsbeschränkungen, so werden alle SPARQL-Abfragen, die an die entsprechende OntoWiki-Installation gestellt werden, geprüft und entsprechend manipuliert. Derzeit ist eine Graph-basierte Zugriffskontrolle integriert, so dass Abbildungen von Benutzern auf RDF-Modellen geprüft werden und etwaige Zugriffsverweigerungen in der Entfernung

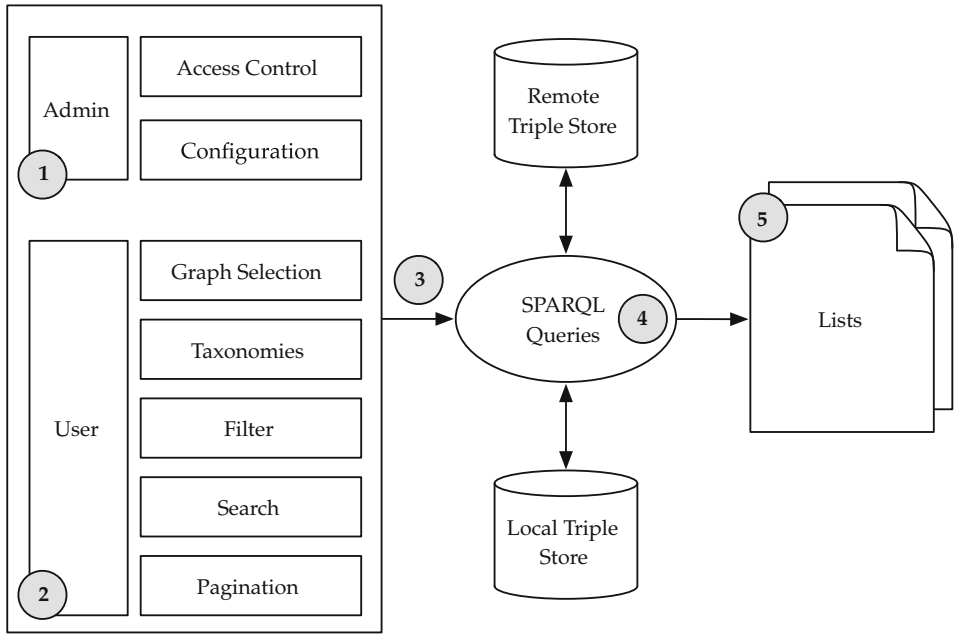


Abb. 6.4 Der Prozess zum Erzeugen einer Liste in OntoWiki: Administrative Facetten (1) und Facetten, die durch den Benutzer bedient werden (2) beeinflussen die Erzeugung einer SPARQL-Abfrage (3), welche gegen lokale resp. entfernte SPARQL-Endpunkte abgesetzt wird und in einer HTML-Tabelle dem Benutzer präsentiert wird (5)

entsprechender `FROM`-Klauseln resultieren. Zudem werden alle dem Benutzer zugänglichen RDF-Graphen den abgesetzten SPARQL-Abfragen automatisiert hinzugefügt.

In erster Linie werden jedoch Filter, welche die zu erzeugende Ergebnisliste beeinflussen, durch Benutzer gegeben:

- Durch die Auswahl einer Wissensbasis wird die `FROM`-Klausel der SPARQL-Abfrage erneut gefiltert.
- Hinzufügen weiterer Filter-Bedingungen schränkt die Ergebnisliste weiter ein (z. B. durch Benutzung des Filter-Moduls). Hierzu werden vornehmlich SPARQL-`FILTER`-Klauseln verwendet.
- Die Benutzung der Stichwortsuche resultiert in der Erzeugung weiterer FILTER-Klauseln nach dem Schema: `FILTER(regex(...))`.
- Durch Selektion spezifischer Teilmengen der Ergebnisliste (Pagination), wird der SPARQL-Abfrage eine Kombination aus `LIMIT` und `OFFSET` beigefügt.

Anschließend wird ein Algorithmus gestartet, welcher die in der Ergebnisliste enthaltenen Ressourcen dereferenziert und dem HTML-Generator zur Verfügung stellt. Dies ermöglicht eine domänen-agnostische Erzeugung von Ergebnislisten.

Generische Ressourcenansicht

Das zweite zentrale UI-Element zur generischen Datenvisualisierung in OntoWiki ist die tabellarische Ressourcenansicht. Diese ist ebenfalls domänen-agnostisch und kann auf alle RDF-Ressourcen angewendet werden. Diese Darstellung enthält alle Fakten, die im selektierten RDF-Model einem ebenfalls selektierten Subjekt zugeordnet werden können. Zur Visualisierung der Fakten (*Statements*) werden die folgenden Tabellenspalten generiert:

- Prädikatsspalte (enthält das Prädikat des Statements),
- Wertspalte (enthält das Objekt des Statements).

Zur Erzeugung dieser Ansicht wird eine triviale SPARQL-Abfrage verwendet, die ausschließlich ein Triple-Pattern mit dem definierten Subjekt (der selektierten RDF-Ressource) enthält.

Zum Erhalt der Übersichtlichkeit werden Werte anhand verwendeter Prädikate gruppiert und ausschließlich eine vordefinierte Anzahl von Werten angezeigt. Sind mehr Werte vorhanden, so können diese durch die entsprechende Interaktion (*More-Link*) vom Benutzer abgerufen werden. Alle in der Ressourcen-Darstellung enthaltenen URIs (an Prädikat- sowie Objektposition) werden als Links generiert und führen bei Verwendung ebenfalls zur jeweiligen Ressourcen-Ansicht.

Werden URIs in Objektposition verwendet, so stellen diese die *ausgehenden* Kanten der jeweiligen Ressource im RDF-Graph dar. Um den Grad der Vernetzung der aktuell selektierten Ressource in Erfahrung zu bringen, sind allerdings auch die *eingehenden* Kanten von Interesse. Auch hierfür enthält die Ressourcenansicht ein Widget, welches diese als Links generierten URIs enthält.

6.3.2 Visuelle Repräsentation semantischer Inhalte

In diesem Abschnitt werden Visualisierungskonzepte von OntoWiki präsentiert. Diese orientieren sich an den in Tab. 6.2 beschriebenen Ebenen.

OntoWiki-Komponenten zur Unterstützung der Visualisierung Der Umgang mit URIs kann Benutzern, die keinen bzw. nur geringes technisches Hintergrundwissen besitzen, nicht abverlangt werden. Würden URIs in der GUI dargestellt werden, so können Darstellungen unübersichtlich werden (bedingt durch Zeichenlänge und die häufig nicht aussagekräftigen Bestandteilen von URIs). Um dies zu verhindern, werden URIs in OntoWiki, wenn möglich, nicht angezeigt und stattdessen Ressourcentitel verwendet.

Title Helper Titel von Ressourcen werden an verschiedensten Stellen im OntoWiki benötigt. Filterformulare, Ergebnislisten nach durchgeführten Suchen, Editierformulare und die Anzeige verlinkter sowie verlinkender Ressourcen sind hierfür Beispiele. Deshalb wurde der *Title Helper* zur Unterstützung der Extraktion von Ressourcentiteln und darauf aufbauender Erweiterungsmöglichkeiten als zentrales Konzept in OntoWiki verankert.

Somit werden die folgenden Vorteile ermöglicht:

- Ressourcen werden in OntoWiki konsistent beschrieben,
- OntoWiki-Erweiterungen müssen das Konzept der Titel-Extraktion nicht selbständig mitliefern,
- Extrahierte Titel können zentral vorgehalten werden, um die Performanz des OntoWikis nicht zu beeinträchtigen.

Zur Repräsentation von RDF-Ressourcen-Titeln existieren Prädikate aus wohl-bekannten Vokabularen wie beispielsweise SKOS, DCTerms[4], FOAF[5], DOAP[6], SIOC und RDFS. Zudem werden diejenigen Titel bevorzugt zur Anzeige gebracht, die der vom Benutzer selektierten Sprache entsprechen. Aufgrund der Tatsache, dass die meisten Wissensbasen zur Repräsentation von Ressourcentiteln Prädikate aus den oben genannten Vokabularen verwenden, funktioniert der *Title Helper* ohne weitere Konfigurationen. Werden allerdings domänenspezifische Prädikate verwendet, so können diese ebenfalls durch Erweiterung der Konfiguration zur Titelextraktion verwendet werden. Auch eine heterogene Verwendung von Prädikaten zur Repräsentation von Titeln ist möglich. Der verwendete Titelextraktor verfügt über einen Algorithmus, welcher die konfigurierte Reihenfolge der zu bevorzugenden Titel beachtet. Ist es dem Algorithmus nicht möglich Titel zu extrahieren, so wird je nach URI-Schema der letzte Teil des URI-Pfades bzw. der Fragmentteil in Kombination mit dem Präfix des Namensraums als Titel verwendet. Insofern diese URI-Teile beim URI-Design selbstsprechend gestaltet wurden, liefert diese Lösung auch brauchbare Ergebnisse.

Plugins Im Standardfall werden URIs als Links in der GUI dargestellt, bei denen der extrahierte Titel zur Anzeige gebracht wird und die Ressourcenansicht als Linkziel definiert wurde. Dieses Verhalten ist in einigen Fällen nicht immer praktikabel, wie die folgenden Beispiele aus dem FOAF-Vokabular verdeutlichen:

- `foaf:homepage` wird benutzt, um RDF-Ressourcen mit Homepage-Ressourcen in Relation zu setzen. Wird eine Homepage-Ressource im OntoWiki als Link generiert, so würden Benutzer bei Verwendung erwarten die entsprechende Homepage zu erreichen. Gäbe es hierfür keine spezielle Lösung, so würde der Benutzer allerdings die OntoWiki-Ressourcenansicht über diese Homepage-Ressource präsentiert bekommen.
- `foaf:mbox` wird zur Kodierung einer Relation zwischen Agenten und Mailboxen (URIs, die dem mailto-Schema folgen) verwendet. Im Standardfall würde bei Verwendung dieser URI als Link die Ressourcenansicht Anwendung finden. Benutzer würden allerdings das Öffnen des lokalen Email-Clients erwarten.

[4] Siehe http://dublincore.org/documents/dcmi-terms/, aufgerufen am 04.04.2014.
[5] Siehe http://xmlns.com/foaf/spec/, aufgerufen am 04.04.2014.
[6] Siehe https://github.com/edumbill/doap/wiki, aufgerufen am 04.04.2014.

- `foaf:depiction` wird verwendet, um RDF-Ressourcen mit Bilder-Ressourcen in Relation zu setzen. Unter Umständen ist es gewünscht Bilder nicht nur zu verlinken, sondern direkt anzuzeigen.

Um derartige Ausnahmen zu regeln, existiert ein Plugin-Konzept in OntoWiki, das für die genannten Beispiele und weitere Fälle verwendet wird. Je nach modelliertem Domänenkonzept kann die Ausnahmeregelung für mehr als nur einzelne Werte notwendig machen.

Geographische Koordinaten beispielsweise werden häufig mit Hilfe des WGS84-Standards kodiert und somit durch mehr als einen Wert repräsentiert. Auch hierfür kann das in OntoWiki eingebaute Plugin-Konzept verwendet werden. So wurde beispielsweise die *Map*-Komponente geschaffen, die neben vielen anderen Funktionen, die Darstellung von Ressourcen mit Geo-Koordinaten auf Karten leistet.

Navigationskomponente

RDF-Wissensbasen sind typischerweise folgendermaßen strukturiert:

1. Gruppierung von Instanzen durch Verwendung von Klassen,
2. Aufbau von Klassen- und Eigenschaftshierarchien unter Verwendung entsprechender Relationen,
3. Beschreibung von Konzepten durch Relationen zu weiteren abstrakteren bzw. konkreteren Ressourcen sowie
4. Benutzung domänenspezifischer Eigenschaften zur Beschreibung von Entitäten.

Die in OntoWiki enthaltene Navigationskomponente dient u. a. als Einstiegspunkt für die Exploration von RDF-Wissensbasen. Die Punkte 1 bis 3 der obigen Liste werden dabei ohne weitere Konfigurationen ermöglicht. Somit werden Klassen- und Eigenschaftshierarchien, die mittels RDFS- und OWL-Konzepten realisiert wurden, standardisiert extrahiert und zur Anzeige gebracht. Auch Hierarchien, die unter Verwendung von SKOS kodiert wurden (Punkt 3), sind vordefiniert explorierbar (Unter vorheriger Selektion der alternativen Hierarchieart). Die jeweils abstraktesten Konzepte werden zu Beginn präsentiert und durch entsprechende Nutzerinteraktion können die jeweils konkreteren Konzepte selektiert werden. Existieren domänenspezifische Hierarchiekonzepte, so können diese ebenfalls konfiguriert werden.

Wurden Konzepte von Interesse durch Benutzung der Navigationskomponente ermittelt, so sind die damit vernetzten Ressourcen anzuzeigen (z. B. die Instanzen einer gefundenen OWL-Klasse). Durch Selektion des Konzeptes werden vernetzte Ressourcen in einer Ressourcenliste dargestellt. Dabei sind in der Regel nicht alle vernetzten Ressourcen von Interesse, sondern nur diejenigen, die mittels eines definierten Prädikates in Relation gebracht wurden. Derartige Prädikate können ebenfalls konfiguriert werden. Am Beispiel von Instanzen einer selektierten OWL-Klasse kommt hierbei die `rdf:type`-Relation zum Einsatz.

Content Management Eine Komponente resp. Funktionseinheit, die in den letzten Jahren häufig nachgefragt wurde, fokussiert auf Content-Management-Methoden. Präzise ausgedrückt wurde eine Methodologie zur Erzeugung einer vollständig anwendungsfallorientierten HTML-Repräsentation von RDF-Ressourcen angefordert. Spezielle Interessengruppen äußerten diese Anforderung, motiviert durch die Notwendigkeit Daten standardisiert zu sammeln, als RDF zu speichern und zu verlinken, Resultate allerdings anwendungsfallspezifisch optisch aufzubereiten und im Web zu präsentieren. In derartigen Szenarien agiert OntoWiki als Content-Management-Backend zur Pflege der Daten, während im Frontend die Daten ausschließlich publiziert und nicht von Benutzern zu editieren sind. Die Unterscheidung zwischen Frontend- und Backend-GUIs ist typisch für die meisten Content-Management-Systeme (CMS)[7]. Diese Anforderung resultierte in der OntoWiki-Site-Extension[8]. Die Site-Extension unterstützt URI-Designs basierend auf wohl-bekannten Dateiendungen wie `ttl` für Turtle [5], `nt` für NTriples[9] und `rdf` für RDF/XML. Dies bedeutet, dass verschiedene Repräsentationsarten einer Ressource `X` (z. B. http://example.com/X) durch die Verwendung des spezifischen Suffixes abgerufen werden können (z. B. `X.ttl` oder `X.nt`). Clients, die auf `X` zugreifen, werden folgend zu den akzeptabelsten Informationsressourcen weitergeleitet. Hierfür wird das `Accept`-Attribut des HTTP-Request-Headers ausgewertet. In den meisten Fällen wird die HTML-Repräsentation angefordert, die durch die Site-Extension erzeugt wird. Zur Erzeugung der HTML-Ausgaben kommen Templates zum Einsatz, die einerseits geschachtelt werden können und deren Erstellung durch Verwendung einer eigens implementierten Wiki-Markup unterstützt wird. Zudem kommen vorgefertigte Erweiterungen zum Einsatz, wie beispielsweise die Annotation der HTML-Repräsentationen mit RDFa.

Konvertierung, Exploration und Visualisierung statistischer Informationen Die Repräsentation von Daten als CSV bzw. in Tabellenform ist, je nach Domäne und Komplexität der Informationen, nicht benutzerfreundlich. Oftmals werden zum Verständnis der dargelegten Tabelleninhalte weitere Dokumente mit Beschreibungen des Inhaltes benötigt. Hierzu bietet sich RDF als Format an, um Daten sowie die dazugehörige Semantik miteinander zu vereinen. Enthalten derartige Daten zudem Statistiken, so können diese durch das RDF-DataCube-Vokabular [9] repräsentiert werden. Dies ermöglicht einerseits, speziell an die Statistik-Domäne angepasste Semantik zu kodieren und andererseits die Verarbeitung dieser Daten durch Software zu ermöglichen, die auf das RDF-DataCube-Vokabular aufbauen.

Eines der Werkzeuge, das zur Exploration und Visualisierung von RDF-DataCubes eingesetzt werden kann, ist CubeViz. Der DataCube-Explorer CubeViz[10] wurde entwickelt, um die Komplexität des RDF-DataCube-Vokabulars vor Interessierten zu verbergen und

[7] Zum Beispiel Drupal (http://drupal.org) und Wordpress (http://wordpress.com).
[8] Siehe https://github.com/AKSW/site.ontowiki, aufgerufen am 04.04.2014.
[9] Siehe http://www.w3.org/2001/sw/RDFCore/ntriples/, aufgerufen am 04.04.2014.
[10] Siehe http://aksw.org/Projects/CubeViz, aufgerufen am 04.04.2014.

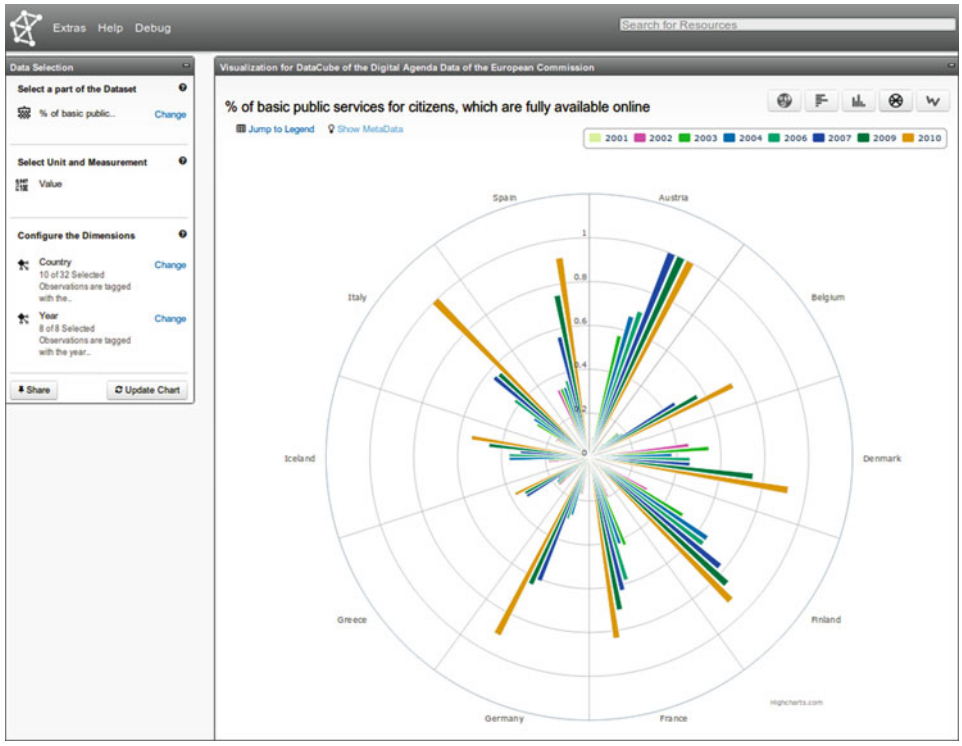


Abb. 6.5 Benutzeroberfläche des Explorers für statistische RDF-Daten CubeViz

die benutzerfreundliche Exploration von derart kodierten Statistiken zu ermöglichen. Dieses Werkzeug wurde als eine Sammlung der folgend gelisteten OntoWiki-Erweiterungen entwickelt:

1. *Komponente zur Analyse auf Integrität vorliegender DataCubes*: Diese Komponente prüft einerseits die Existenz statistischer Daten (repräsentiert mit DataCube) im ausgewählten RDF-Graphen und kann zudem die Qualität der DataCubes anhand empfohlener Integritätsbedingungen testen.
2. *Komponente zur facettierten Selektion statistischer Domänen-Konzepte*: Diese Komponente nutzt SPARQL zur Abfrage von strukturellen Teilen des DataCubes des ausgewählten RDF-Graphen, um darauf aufbauend Facetten zum Filtern zu erzeugen.
3. *Komponente zur Visualisierung selektierter statistischer Beobachtungen*: Diese Komponente dient der Visualisierung von Observationen, die auf Basis der zuvor gegebenen Filterkriterien selektiert wurden.

Alle Komponenten sind in der OntoWiki-GUI integriert (illustriert in Abb. 6.5) und verwenden ein gemeinsames Datenmodell auf Server- sowie auf Clientseite. Folgend werden die zweite und dritte Komponente näher erläutert.

Facettierte Exploration statistischer Daten Resultiert die Ausführung einer essentiellen Introspektionsabfrage in ein positives Ergebnis, so wird CubeViz gestartet und Facetten auf Basis struktureller Informationen des vorliegenden DataCubes erzeugt. Bedingungen an valide DataCubes, die die Existenz von strukturellen Bestandteilen voraussetzen, sind dabei als folgende obligatorische Facetten umgesetzt und bei jeder Exploration vorhanden und zu benutzen:

- *Dataset selection*: Selektion genau einer Instanz vom Typ `qb:DataSet`,
- *Measure property selection*: Selektion genau eines Prädikates vom Typ `qb:MeasureProperty`, das zur Spezifikation des Messwertes von Observationen eingesetzt wird und
- *Dimension element selection*: Selektion von mindestens (bzw. genau) einem Dimensionselement je Dimension.

CubeViz fokussiert ausschließlich die Darstellung der im DataCube kodierten Messwerte und ist nicht mit Aggregatsfunktionen wie `SUM`, `AVG`, `MIN` und `MAX` ausgestattet. Dadurch begründet sich die Selektion von jeweils mindestens einem Element je Dimension, um die Mehrfachauswahl von Observationen der entsprechend selektierten Dimensionselemente zu verhindern. Weiterhin wird die Facettenmenge mit den folgenden fakultativ zu verwendenden Facetten angereichert:

- *Slice selection*: Selektion maximal einer materialisierten Teilmengendefinition, die als Instanz vom Typ `qb:Slice` im RDF-Graphen enthalten ist und mit der obligatorisch selektierten Datenstrukturdefinition in Relation steht, sowie
- *Attribute selection*: Selektion der zu verwendenden Einheit, die in Form einer `qb:AttributeProperty` als Teil der Observation kodiert ist.

Diese fakultativen Facetten stehen nur zur Auswahl, sofern die Existenz entsprechender Konzepte im DataCube gegeben ist. Sind die Konzepte vorhanden, so können diese zum Filtern verwendet werden, zur Auswahl stehende Elemente sind standardmäßig allerdings nicht belegt.

Alle generierten Facetten eröffnen die Möglichkeit, interessante Teilmengen des DataCubes zu extrahieren. Die jeweiligen Elemente können durch Nutzung von Filterdialogen selektiert bzw. deselektiert werden. Je nach Typ der Facette (obligatorisch / fakultativ) wurden GUI-Elemente gewählt und mit JS-Funktionen ausgestattet, um die minimale und maximale Anzahl selektierbarer Elemente zu begrenzen. Diese Dialoge können durch Verwendung des entsprechenden Links in der *Data Selection Box* geöffnet werden. Je nach Facette wird das selektierte Element bzw. die Anzahl selektierter Elemente an die *Data Selection Box* zurückgeliefert und zur Anzeige gebracht. Alle ermittelten strukturellen Elemente inklusive ihrer Metadaten und Titel werden in einer JS-Repräsentation vorgehalten und werden bei der Erzeugung der Diagramme und Legenden wiederverwendet.

Einer der Vorteile des facettierten Explorierens ist die Reduktion möglicher leerer Ergebnismengen. Um dies umzusetzen, wird bei Benutzung einer Facette nicht nur die

Ergebnismenge entsprechend gefiltert, sondern zusätzlich auch jede weitere Facette inklusive ihrer jeweiligen Elemente. SPARQL ist für derartige Funktionalitäten optimal nutzbar. Selektierte Elemente bereits benutzter Facetten werden durch eine Menge an Triple-Patterns repräsentiert und in den SPARQL-Abfragen zur Ermittlung von Elementen aller Facetten integriert. Notwendig ist zudem, dass nicht nur die strukturellen Teile selbst ermittelt werden, sondern auch die Verwendung dieser durch Observationen (ebenfalls durch Triple-Patterns der jeweiligen SPARQL-Abfrage repräsentiert). Zur Erzeugung der Facetten der *Data Selection Box* in CubeViz wurde dies aber nur bedingt umgesetzt, um die Anzahl der aus diesem Vorgehen resultierenden SPARQL-Abfragen weiter zu reduzieren. Gerade bei der Exploration von DataCubes mit vielen Dimensionen (> 10) und vielen Dimensionselementen (> 50 je Dimension) durch viele gleichzeitig agierende Benutzer kann ein derartiges Vorgehen zu hoher Last auf dem Server bzw. der Datenbank führen. Somit wurde die Rückkopplung selektierter Dimensionselemente in die Erzeugung aller weiteren Facetten entfernt.

Visualisierung statistischer Beobachtungen Wurde eine Datenselektion durchgeführt, so können entsprechende Observationen aus dem RDF-Graph extrahiert und zur Anzeige gebracht werden. Hierzu werden die Filterkriterien an die Komponente zur Visualisierung selektierter Observationen übergeben, die auf Basis dieser Filterkriterien eine entsprechende SPARQL-Abfrage generiert. Diese SPARQL-Abfrage wird an den konfigurierten SPARQL-Endpunkt weitergeleitet und die resultierende Ergebnismenge in ein JSON-Format transformiert. Die so kodierte Ergebnismenge wird an die Oberfläche übergeben und durch eine weitere Analyse auf Anzahl disjunkter Dimensionselemente und deren Zuordnung zu Dimensionen untersucht. Wurde festgestellt, wie viele Dimensionen mit jeweils mehr als einem Dimensionselement in der Ergebnismenge vorhanden sind, wird in den JS-seitig repräsentierten Diagrammkonfigurationen die Menge adäquater Diagramme (Charts) bzw. deren Entsprechungen als JS-Implementierungen (Chart-Klassen) ermittelt. Wurde eine Menge brauchbarer Chart-Klassen ermittelt, so wird die erste Klasse dieser Menge instanziiert und die Menge erhaltener Observationen übergeben.

CubeViz enthält eine Abstraktionsebene um zwischen SPARQL (Abfrage von Observationen) und den APIs zur Visualisierung von Charts zu vermitteln. Derzeit sind die APIs *Data Driven Documents*[11] (D3js) und *HighCharts*[12] angebunden sowie diverse Charts implementiert, wie beispielsweise *Pie*-, *Bar*-, *Column*-, *Line*- und *Polar*-Chart. Notwendige Konvertierungsmechanismen zur Transformation der JSON-Repräsentation der Observationsmenge in das Chart-spezifische Eingabeformat werden von der Chart-Implementierung bereitgestellt. Nach erfolgreicher Konvertierung der Observationen in das spezifische Eingabeformat wird die Chart-Erzeugung und Darstellung im Browser gestartet (siehe Abb. 6.5). Dabei werden die während der Datenselektion ermittelten Ressourcen-Titel von Dimensionen und entsprechenden Elementen wiederverwendet.

[11] Siehe http://d3js.org/, aufgerufen am 04.04.2014.
[12] Siehe http://www.highcharts.com/, aufgerufen am 04.04.2014.

Weitere Chart-APIs, wie beispielsweise die *Google Charts API*[13], können durch Existenz dieser Abstraktionsebene lose gekoppelt integriert werden.

Alle geeigneten Chart-Implementierungen, die zur Anzeige abgefragter Observationen ermittelt wurden, werden in Form eines Chart-Selektors offeriert und können vom Benutzer ausgewählt werden. Dazu werden keine weiteren SPARQL-Abfragen durchgeführt, sondern die bereits clientseitig vorhandenen Observationen wiederverwendet und durch die jeweilige Chart-Implementierung in das benötigte Eingabeformat konvertiert. Die implementierten Charts bieten weiterhin konfigurierbare Optionen, um beispielsweise Messwerte im Chart anzuzeigen, Achsen oder die Dimensionen zu tauschen, die zu verwendende Werte-Skala (linear oder logarithmisch) sowie die Kombination von Chart-Typen auszuwählen (z. B. Polar-Chart und Column-Chart, wie dies in Abb. 6.5 zu sehen ist). Um Chart-Optionen zu setzen, wurde ein Widget implementiert, das aus dem Chart-Selektor heraus geöffnet werden kann. Wurden alle Optionswünsche unter Verwendung des Widgets gesetzt, so wird eine entsprechende Aktualisierung des Charts nach einer Bestätigung gestartet.

Interaktive Legende Zusätzlich zur Darstellung von selektierten Strukturelementen in der *Data Selection Box* und entsprechenden Observationen innerhalb des generierten Charts werden alle Werte in Form einer Legende unterhalb des Charts zusammengefasst. Dabei werden zusätzlich auch die Metainformationen von Dimensionen, Dimensionselementen, Datasets etc. aufgeführt. Extrahierte Observationen sind ebenfalls tabellarisch erfasst, so dass eine Übersicht von Messwerten in Kombination mit Kontextinformationen ermöglicht wird. An dieser Stelle ist es möglich, eventuelle Fehler der originalen Messwerte manuell zu beheben. Nach Änderung von Messwerten wird das Diagramm automatisch aktualisiert.

Publikation selektierter statistischer Observationen Nach Selektion gewünschter Observationen, Konfiguration von Chart-Optionen und eventueller Korrektur von Messwerten ist unter Umständen die Erstellung einer dereferenzierbaren URL gewünscht, um die erstellte Ausgabe statistischer Daten zu erreichen (*Shared access*). In frühen Versionen von CubeViz wurden alle Parameter, welche die Ausgabe beeinflussen (URI von DataCube-Elementen, Chart-Typ und Chart-Optionen), unverändert als Teil der URL kodiert. Die Selektion vieler Elemente eines vorliegenden DataCubes ist bei derartigem Vorgehen allerdings durch die Begrenzung der URL-Länge limitiert (Konfiguration des Webservers). Zudem wurde die Kommunikation zwischen Triple-Store und Client später auf AJAX umgestellt, so dass eine automatische Anpassung der Parameter in der URL nicht mehr existierte. Um trotzdem eine Dereferenzierungsfunktionalität zu offerieren, wurde eine entsprechende JS-Funktion integriert, die alle notwendigen Parameter clientseitig sammelt und nach manuellem Aufruf an den Server überträgt. Auf Serverseite werden alle Parame-

[13] Siehe https://developers.google.com/chart/, aufgerufen am 04.04.2014.

ter entsprechend ihrer Funktion gruppiert, jeweils sortiert und auf Basis dieser sortierten Parametergruppen zwei Hashes erzeugt. Der erste Hash wird verwendet, um Datenselektionsparameter (URIs struktureller Ressourcen des DataCubes) identifizierbar zu speichern. Der zweite Hash repräsentiert alle GUI-spezifischen Parameter, die ebenfalls auf dem Server gespeichert werden. In beiden Fällen wird derzeit jeweils eine Datei mit dem Hash als Namen und den Parametern in JSON-Notation als Inhalt der Dateien angelegt (Dateibasierter Cache). Der Server antwortet auf diese Anfrage mit beiden erzeugten Hashes, welche wiederum als Parameter einer generierten URL in der GUI (Permalink) verwendet werden. Dieser Permalink referenziert eine Funktion, die alle Parameter auf dem Server extrahiert und in den entsprechenden SPARQL-Abfragen resp. den GUI-Konfigurationen wiederverwendet, um die ursprüngliche Ausgabe wieder herzustellen. Die dabei abgefragten Observationen können durch Verwendung des adaptiven SPARQL-Caches (vgl. [18]) vorgehalten werden, um die Laufzeit dieser HTTP-Anfrage zu optimieren.

Eine weitere Möglichkeit zur Publikation der mittels CubeViz selektierten Observationen ist der Export von Rohdaten. Diese können in den Formaten CSV und RDF (Turtle) heruntergeladen werden, um eine Weiterverarbeitung mittels anderer Werkzeuge zu ermöglichen. Die URL zu Exportfunktionalitäten werden gleichermaßen, wie oben beschrieben, mit dem Hash zur Repräsentation der Datenselektionsparameter angereichert und sind somit dereferenzierbar.

6.4 Datenerstellung und Datenpflege

Der enorme Erfolg des World Wide Webs resultiert überwiegend daraus, dass gewöhnliche Benutzer befähigt wurden Inhalte sehr einfach zu editieren. Um Inhalte im WWW zu veröffentlichen, mussten Nutzer lediglich Textdateien manipulieren und diese mit wenigen, leicht zu erlernenden HTML-Tags annotieren. Für das Semantic Data Web im Allgemeinen und für Linked Data Autorenwerkzeuge im Besonderen stellt sich die Situation etwas komplizierter dar. Hier müssen Nutzer nicht nur eine neue Syntax erlernen (z. B. Turtle, RDF/XML oder RDFa), sondern sie müssen sich zusätzlich mit dem RDF Datenmodell, diversen Ontologiesprachen (bspw. RDF-S, OWL) und einer stetig wachsenden Anzahl von RDF Vokabularen vertraut machen. OntoWiki erleichtert Nutzern den Einstieg in den Bereich der Erstellung und Pflege von semantischen Inhalten, indem es Nutzerschnittstellen bereitstellt, welche die Komplexität des RDF Datemmodells verbergen aber dieses gleichzeitig vollständig abbilden.

6.4.1 RDFauthor

RDFauthor beruht auf der Idee, beliebige mit RDFa annotierte XHTML Dokumente editierbar zu machen. *RDFa* [1] ist eine W3C Recommendation, mit der es möglich wird,

Repräsentationen für Menschen und Computer in einem einzigen HTML Dokument zu kombinieren. RDFauthor baut auf RDFa auf, indem es mit Hilfe von benannten Graphen Herkunftsinformationen in RDFa Repräsentationen erhält. Zusätzlich erzeugt RDFauthor eine Abbildung von RDFa Sichten auf Editierwidgets. Beim Eintreten von (konfigurierbaren) Ereignissen (bspw. das Klicken auf eine Schaltfläche oder das Bewegen des Mauszeigers über ein bestimmtes Informationsfragment), werden bestimmte Widgets aktiviert und ermöglichen somit die Bearbeitung sämtlicher Informationen auf einer entsprechend annotierten Webseite. Während der Bearbeitung können Widgets im Hintergrund auf Informationsquellen im Data Web zurückgreifen, um die Wiederverwendung von URIs zu erleichtern bzw. die Verlinkung von Ressourcen zu fördern. Das Widget zur Bearbeitung von Ressourcen zum Beispiel schlägt geeignete, bereits eingesetzte URIs vor, die vom Sindice Semantic Web Index [24] abgefragt werden. Sobald die Bearbeitung abgeschlossen ist, werden sämtliche Änderungen an die zugrunde liegenden Datenbanken (Triple Stores) propagiert, wobei SPARQL/Update zum Einsatz kommt. Dies ermöglicht die Integration von RDFauthor Widgets über OntoWiki hinaus. Sie können so beispielsweise in dedizierten Webseiten eingesetzt werden, die von der Site-Erweiterung (siehe Abschnitt zu 6.3.2) erzeugt wurden.

Darüber hinaus ist RDFauthor nicht auf das Bearbeiten von semantischen Daten aus einer einzelnen Quelle beschränkt. Ein RDFa Dokument, welches mit RDFauthor editiert wird, kann Aussagen aus einer Vielzahl von Quellen enthalten. Diese können zeitgleich und in einer für den Nutzer transparenten Art und Weise editiert werden. Basierend auf einer RDFa Erweiterung, die benannte Graphen und Informationen zu SPARQL/Update Endpunkten unterstützt, werden gleichzeitige Änderungen, die verschiedene Graphen betreffen, zu den jeweils zuständigen SPARQL/Update Endpunkten weitergereicht. RDFauthor ist in JavaScript implementiert, so dass es vollständig im Browser agiert und zusammen mit beliebigen Techniken zur Entwicklung von Web-Applikationen eingesetzt werden kann. Als solches ist es nicht an den Einsatz mit OntoWiki gebunden.

RDFa ermöglicht die Anreicherung von Informationen, die in HTML repräsentiert sind, mit RDF. Dadurch können RDF Tripel aus solchen Dokumenten extrahiert werden. RDFauthor sorgt nun dafür, dass solche Tripel bearbeitet werden können. Um Änderungen nun aber dauerhaft im Wiki speichern zu können, benötigt RDFauthor Informationen über die Datenquelle (also Informationen zum SPARQL und SPARQL/Update Endpunkt). Insbesondere muss klar sein, aus welchen RDF Graphen die Tripel extrahiert wurden bzw. in welchen Graphen sie geschrieben werden sollen. Um diese Informationen bereitzustellen, haben wir eine leichtgewichtige Erweiterung zu RDFa definiert.

Um Informationen über die Datenquelle abzubilden, folgen wir dem *named graphs* Ansatz [8]. Wir haben zu diesem Zweck ein kleines Schema erzeugt[14]. Dieses Schema enthält Definitionen für Attribute und Relationen, die wie folgt eingesetzt werden:

[14] Der Namensraum für das RDFauthor Vokabular ist http://ns.aksw.org/update/. Wir nutzen das `update` Präfix im weiteren Verlauf dieses Artikels.

- Um bestimmte RDFa Annotationen in einem Dokument an die jeweiligen SPARQL bzw. SPARQL/Update Endpunkte zu binden, schlagen wir die Benutzung des `link` HTML Tags in Verbindung mit einem `about` Attribut vor, um den Graphen zu benennen. Des Weiteren sollte für das `rel` Attribut der Wert `update:updateEndpoint` verwendet werden und das `href` Attribut sollte die URL des Endpunktes enthalten. Eine weitere Möglichkeit diese Graphmetadaten zu definieren, ist die Verwendung von leeren `span` oder `div` Elementen in Verbindung mit den entsprechenden RDFa Attributen.
- Um festzulegen, welche Tripel zu welchen Graphen gehören, schlagen wir den Einsatz des `update:from` Attributs vor. Der Wert für dieses Attribut enthält den jeweiligen Graphen, dem alle untergeordneten RDFa Annotationen zugeordnet werden sollen. Das `update:from` Attribut und die zusätzlichen Verarbeitungsregeln für RDFa sind von dem Ansatz von [15] inspiriert.

Sämtliche Sichten, die von OntoWiki erzeugt werden (Listen und Detailansichten für Ressourcen) enthalten auch Graphmetadaten. Zusätzlich kann die *Site*-Erweiterung mit der Hilfe von Templates diese Daten ebenfalls einbetten. Die Erzeugung von Widgets kann mittels verschiedener Ereignisse ausgelöst werden. Diese Ereignisse lassen sich in zwei Kategorien aufteilen:

- Ereignisse, die sich auf bestimmte Elemente beziehen (*Element-basiert*) und
- solche, die sich auf das HTML Dokument beziehen (*page-wide*).

Als Folge einer Nutzerinteraktion oder durch einen programmatischen Auslöser, startet RDFauthor die Auswertung der aktuellen Webseite. Dabei werden sämtliche RDF Tripel extrahiert und in einer clientseitigen Datenbank (*rdfQuery databank*) abgelegt, wobei für jeden Graphen eine eigene Datenbank genutzt wird. Tripel, welche die Graphen der Seite beschreiben, indem sie das `update` Vokabular nutzen, werden dabei explizit ignoriert. Für den Fall, dass keine Informationen zum Update-Endpunkt für einen Graphen gefunden werden, wird dieser als nicht editierbar markiert und folglich werden keine Formulare für diese Tripel erzeugt.

Falls der Quellgraph einer Menge von Tripeln mit einem SPARQL Endpunkt verknüpft ist, versucht RDFauthor Metadaten zu den verwendeten Attributen zu beziehen. Dazu wird eine SPARQL-Anfrage gesendet, welche die jeweiligen `rdf:type` und `rdfs:range` Werte für die Attribute ausliest. Mit Hilfe dieser Informationen wählt RDFauthor ein passendes Editier-Widget aus. Alle Widgets, die ausgewählt wurden, werden in einem Bearbeitungs-Formular zusammengeführt und dem Nutzer präsentiert. Abhängig vom auslösenden Ereignis, wird das erzeugte Formular entweder

- als *Overlay*-Fenster dargestellt oder
- in die Webseite integriert.

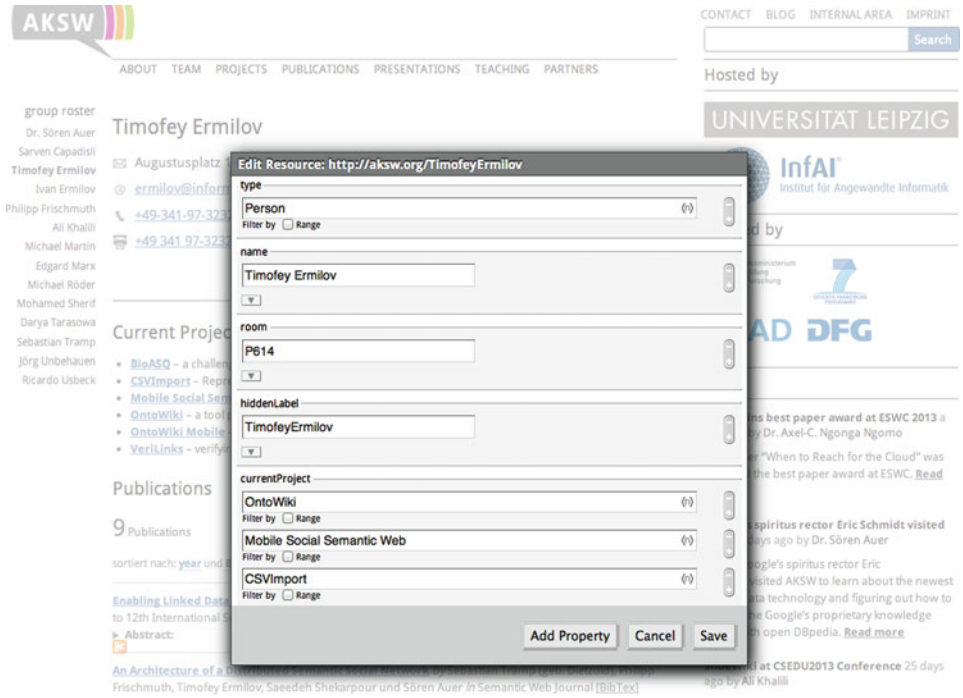


Abb. 6.6 Beispiel eines RDFauthor Widgets: Diese Ansicht zur Bearbeitung wurde durch den *Universal Edit Button* ausgelöst, der in die zugrundeliegende Webseite integriert wurde. Sie wurde als modale Ansicht realisiert, die sich über die ursprüngliche Webseite legt. Nachdem der *Save* Button betätigt wird, werden die Änderungen (hinzugefügte und gelöschte Tripel) mit Hilfe von SPARQL/Update Anfragen zurück an das Wiki kommuniziert

Abbildung 6.6 zeigt ein Beispiel für ein von RDFauthor generiertes Formular in einem *Overlay* Fenster.

Wenn der Nutzer den Bearbeitungsprozess abschließt, werden alle Widgets aufgefordert den jeweils zuständigen Graphen mit den Änderungen zu aktualisieren. Die Unterschiede (*Diff*) zwischen den originalen und den geänderten Graphen werden berechnet (d. h. hinzugefügte Tripel, gelöschte Tripel) und es entsteht ein sogenannter *Diff*-Graph. Für alle Graphen werden anschließend die zugehörigen Tripel-Datenbanken kontaktiert und mit den errechneten *Diff*-Graphen aktualisiert. Dies geschieht mit Hilfe von SPARQL/Update Operationen [22]. Da zu diesem Zweck lediglich alle hinzugefügten und gelöschten Tripel übertragen werden, wobei die `INSERT DATA` und die `DELETE DATA` Syntax verwendet wird, ist nicht zwingend eine vollständige SPARQL/Update Unterstützung erforderlich. Zusätzlich kann RDFauthor mit unterschiedlichen Szenarien zur Zugriffskontrolle umgehen, wozu die Antworten der SPARQL/Update Anfragen ausgewertet werden. Im Falle von `HTTP 401 (Unauthorized)` oder `HTTP 403 (Forbidden)` Status-Codes beispielsweise, wird ein Login-Formular angezeigt.

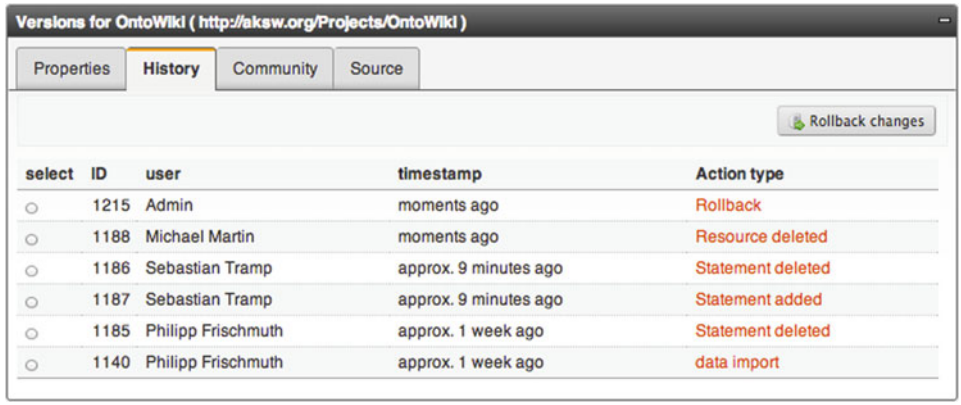


Abb. 6.7 Historie einer Beispiel-Ressource beginnend mit einem Datenimport, gefolgt von kleineren Änderungen und einem versehentlichen Löschvorgang, welcher abschließend rückgängig gemacht wurde

Zusätzlich zur Möglichkeit Tripel in einer Webseite zu bearbeiten, ist es auch möglich neue Tripel hinzuzufügen. Zum einen können dazu existierende Tripel als Templates verwendet werden, zum anderen können auch komplett neue Statements erzeugt werden.

6.4.2 Versionierung

Versionierung von Inhalten ist ein fundamentaler Bestandteil von Wiki-Systemen [17]. Die Möglichkeit vorangegangene Änderungen rückgängig machen zu können, hilft Nutzern Fehler zu beheben und behebt insbesondere die Angst das gesamte System durch eine falsche Eingabe zu zerstören.

Das Versionierungssystem von OntoWiki basiert auf den Informationen zu hinzugefügten und gelöschten Tripeln innerhalb einer Transaktion. Da Änderungen über verschiedenste Kanäle in das System gelangen können (z. B. SPARQL/Update Endpunkt, JSON/RPC Schnittstelle), ist das Versionierungssystem sehr tief in die Triple-Store Zugriffsschicht der Erfurt API integriert.

Dies bedeutet, dass auf jeden einzelnen Aktualisierungsvorgang eine Berechnung der neuen und gelöschten Tripel folgt, um die Änderungen ggf. später rückgängig machen zu können. Mehrere aufeinanderfolgende Aktualisierungsvorgänge können gruppiert und als einzelne Transaktion behandelt werden. Zusätzlich wird jede Änderung mit Nutzer, Zeitstempel und einem optionalen Namen für die Transaktion versehen, um Nutzern eine bessere Übersicht innerhalb des Wikis geben zu können. Die Transaktionsnamen können als IDs für eine Historienansicht verwendet werden, wie in Abb. 6.7 dargestellt.

6.4.3 Import und Triplifizierung von Ressourcen

Eine wichtige Funktion, die bei der täglichen Arbeit mit Daten sehr nützlich ist, ist der Import von Daten verschiedener Größe und aus verschiedenen Datenquellen (auch Datenquellen, die nicht als RDF vorliegen). OntoWiki ist in der Lage Daten mit Hilfe der Linked Data Prinzipien zu beziehen und zu importieren. Dies ist zum Beispiel wichtig, wenn Beschreibungen von Attributen/Relationen oder Klassen aus einem Vokabular importiert werden sollen. OntoWiki kann auch Tripel aus RDFa Webseiten extrahieren und über ein Plugin-System weitere Datenquellen anbinden, die nicht als RDF vorliegen (z. B. EXIF Daten aus Bildern).

Zusätzlich kann OntoWiki eingesetzt werden, um fremde SPARQL Endpunkte anzubinden. Dies ermöglicht einen lesenden Zugriff auf Wissensbasen in der lokalen Umgebung. Solche Endpunkte können von RDB2RDF Werkzeugen bereitgestellt werden (bspw. Sparqlify [25]), um Daten aus relationalen Datenbanken zu integrieren.

6.4.4 Evolution von Datensets

Um komplexere Änderungsaktivitäten zu ermöglichen, haben wir einen Ansatz integriert, der das Schreiben, Nutzen und Verwalten von Evolutionsmustern erlaubt. Der *EvoPat* [21] Ansatz basiert auf der Definition von einfachen Evolutionsmustern, welche deklarativ repräsentiert werden und einfache Evolutions- und Refactoring-Operationen auf Daten- und Schema-Ebene erfassen können. Für komplexere und domänenspezifische Evolutions- und Refactoring-Vorgänge können mehrere einfache Evolutionsmuster zu einer Komposition aus Mustern kombiniert werden. In [21] haben wir eine umfassende Studie von möglichen Evolutionsmustern durchgeführt, inklusive einer kombinatorischen Analyse aller möglichen vorher/nachher Kombinationen. Als Ergebnis daraus haben wir einen umfangreichen Katalog an nutzbaren Evolutionsmustern erhalten. Die Anwendung eines Evolutionsmusters auf eine Wissensbasis führt zu multiplen Änderungen, die als einzelne Transaktion durchgeführt werden. Wenn das Ergebnis nicht dem gewünschten Ergebnis entspricht, können sämtliche Änderungen in einem einzigen Schritt rückgängig gemacht werden.

6.5 Semantische Wikis als Kristallisationspunkte für Enterprise Data Integration

In nahezu allen großen Unternehmen werden heutzutage Taxonomien eingesetzt, um ein gemeinsames linguistisches Modell bereitzustellen mit dem Ziel die große Menge an Dokumenten, E-Mails, Produktbeschreibungen, Unternehmensdirektiven, etc. zu strukturieren, die täglich erzeugt werden. Allerdings werden diese Taxonomien häufig in proprietären Formaten gespeichert, sowie von einer zentralen Abteilung gepflegt und kontrolliert.

Diese Tatsache macht es sehr umständlich solche Daten zu verwenden und insbesondere wiederzuverwenden.

Wir haben OntoWiki in einem Industrieprojekt mit einem großen deutschen Unternehmen eingesetzt, mit dem Ziel diese Situation zu verbessern. Dazu wurden existierende Wörterbücher mit Begriffsdefinitionen in verschiedenen Sprachen triplifiziert. Zu diesem Zweck wurde das standardisierte und populäre SKOS Vokabular eingesetzt. Des Weiteren wurden alle Begriffsdefinitionen mit Hilfe der Linked Data Prinzipien publiziert.

Als ein erstes Resultat dieser Bemühungen waren alle Begriffe, die zuvor über verschiedene Wörterbücher verteilt waren, nun in einer einheitlichen Wissensbasis verfügbar, welche von den Nutzern komfortabel über eine OntoWiki-Instanz exploriert werden konnte.

Um die Vorteile aufzuzeigen, eine solche Unternehmenstaxonomie in RDF zu überführen (insbesondere die Wiederverwendbarkeit), haben wir eine weitere Datenquelle triplifiziert. Diese enthält strukturierte Informationen über die Produkte des Unternehmens (Autos). Wir haben diese Produkte mit den Begriffen der Taxonomie verlinkt und einen dedizierten Suchdienst entwickelt, welcher in Abb. 6.8 abgebildet ist. Der Screenshot auf der linken Seite zeigt OntoWiki mit der Definition des Begriffs *T-Modell* aus dem Taxonomie-Graphen zusammen mit einigen zusätzlichen Informationen. Die URI-Leiste am oberen Rand zeigt die URI für dieses Konzept. Diese kann von anderen Ressourcen genutzt werden, um Links zu diesem Konzept zu erzeugen. Des Weiteren ist es möglich diese URI zu dereferenzieren, um eine Beschreibung der Ressource in einem von Maschinen lesbaren Format zu erhalten. Die Detailansicht für diesen Begriff enthält:

- den Typ der Ressource (`skos:Concept`),
- einen Link zu einem allgemeineren Konzept (hierarchische Ordnung),
- eine textuelle Beschreibung der Bedeutung des Bergriffs,
- bevorzugte Bezeichner für diesen Begriff in verschiedenen Sprachen sowie
- einen alternativen Bezeichner *Combi*.

Zusätzlich zeigt eine kleine Box am rechten Rand des OntoWiki Fensters andere Ressourcen, die einen Link zu diesem Begriff haben. Beispielsweise enthält das allgemeinere Konzept zu dem obigen Begriff auch einen Link, der besagt, dass dieses Konzept eine Spezialisierung darstellt (`skos:narrower`). Des Weiteren verlinken verschiedene Fahrzeugmodelle auf dieses Konzept. Dieser Umstand wird in der Suchapplikation genutzt, die auf der rechten Seite der Abb. 6.8 dargestellt ist. Dieser Screenshot zeigt eine einfache, prototypische Anwendung mit einem Suchfeld. Wenn ein Nutzer den Begriff *Combi* eingibt, wird die Wissensbasis angefragt und das Konzept *T-Modell* wird gefunden, da es den Suchbegriff als Synonym enthält. Anschließend werden alle verlinkten Fahrzeugmodelle abgerufen und dem Nutzer angezeigt. Das hier skizzierte Szenario ist ein gutes Beispiel für eine Wiederverwendung der Taxonomie in einem Kontext, der ursprünglich nicht vorgesehen war.

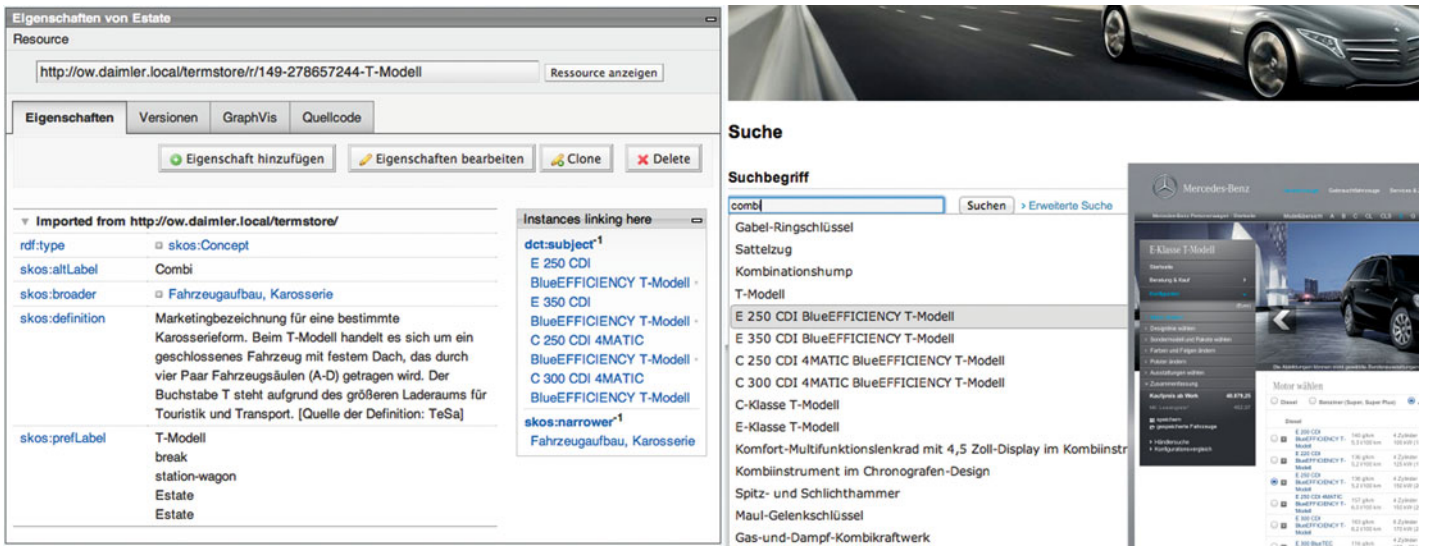


Abb. 6.8 Auf der *linken Seite* ist OntoWiki abgebildet. Es ist die Definition des Begriffs *T-Modell* aus der Taxonomie sichtbar, sowie weitere Ressourcen, die einen Link zu diesem Begriff besitzen. Auf der *rechten Seite* ist eine Suchapplikation sichtbar, nachdem der Begriff *Combi* eingegeben wurde. Diese Applikation nutzt die Metadaten zu dem Begriff und zusätzlich Links zu diesem Konzept, um relevante Inhalte vorzuschlagen

6.6 Zusammenfassung und Ausblick

In diesem Kapitel, haben wir am Beispiel von OntoWiki zentrale Konzepte zur Kuratierung und Visualisierung von Linked Data vorgestellt. Große Herausforderungen sind die *Skalierbarkeit, Funktionalität und Benutzerfreundlichkeit*. Jeder dieser drei Bereiche ist für bestimmte Anwendungen und Anwendungsfälle von entscheidender Bedeutung, aber sie beeinflussen sich gegenseitig. Zum Beispiel gefährden zusätzliche Funktionen oder die Verbesserung der Nutzerfreundlichkeit die Skalierbarkeit. Zusätzliche Funktionen können die Benutzeroberfläche überladen und die Benutzerfreundlichkeit einschränken. Daher ist es von großer Bedeutung, auch wenn Daten-Wikis im Kern generisch und Domänenagnostisch sind, das System flexibel an neue Nutzungsszenarien anpassbar zu machen.

Insbesondere im Unternehmen gibt es vielfältige Einsatzmöglichkeiten für ein semantisches Daten-Wiki: Enterprise Thesauri können damit verwaltet werden. Entitäten und deren Beziehungen im Master Data Management können mit einem semantischen Daten Wiki administriert werden. Stammdaten können mit einem Daten-Wiki dediziert in strukturierter Form für Kooperationspartner, Kunden und Lieferanten zugänglich gemacht werden. Produktdaten können verwaltet werden und für semantische Suchmaschinenoptimierung (semantic SEO) genutzt werden. Im Moment befinden sich diese und andere Einsatzszenarien in vielen Unternehmen noch in der Erprobungsphase, bei einer weiteren Verbreitung in großen Unternehmen und Organisationen ist jedoch damit zu rechnen, dass semantische Wikis zu Kristallisationspunkten in den entstehenden unternehmensinternen Datenintranets und unternehmensübergreifenden Datenextranets werden. Mit dem Linked Data Life-Cycle [4] und dem diesen unterstützenden LOD2 Linked Data Stack [2] (dessen Bestandteil OntoWiki ist) steht eine Vielzahl von Methoden und Technologiekomponenten zur breiteren Unterstützung von Linked Data Management im Unternehmen bereit.

Auch zur Publikation, Visualisierung und Kuratierung von Daten der öffentlichen Hand eignet sich der Einsatz semantischer Wikis. So nimmt die im Internet verfügbare Menge an statistischen Daten ebenso stetig zu wie das Interesse an benutzerfreundlichem Zugriff darauf. Beispielsweise wurden zahlreiche statistische Datensätze im Rahmen des *Open Data Portals* der Europäischen Kommission (ODP)[15] veröffentlicht. In Abhängigkeit vom Format dieser Datensätze (z. B. XLS, CSV) können diese von Interessenten unter Verwendung von Tabellenkalkulationswerkzeugen (z. B. von Google, Document Foundation oder Microsoft) exploriert werden. Werden derartige Daten allerdings unter Verwendung von RDF und, sofern möglich, mit domänenspezifischen Vokabularen wie dem DataCube-Vokabular publiziert, so können Werkzeuge wie der vorgestellte RDF-DataCube-Explorer CubeViz verwendet werden, ohne vorab Anpassungen der Datenrepräsentation vorzunehmen. Werden diese Daten zusätzlich mittels eines SPARQL-Endpunktes offeriert, so ist zudem ein Fern-Zugriff möglich und verringert weiter evtl. Zugangsbarrieren auf die Daten. Im Open Data Portal der Europäischen Kommission wurde eine derartige Infrastruktur integriert und CubeViz zur Exploration ausgewählter statistischer Datensätze eingeführt[16].

[15] Siehe http://open-data.europa.eu/, aufgerufen am 04.04.2014.
[16] Siehe http://open-data.europa.eu/cubeviz/, aufgerufen am 04.04.2014.

Literatur

1. Adida, B., M. Birbeck, S. McCarron, und S. Pemberton. 2008. *RDFa in XHTML: Syntax and Processing. Recommendation, World Wide Web Consortium (W3C), Oct. 2008*. http://www.w3.org/TR/rdfa-syntax/
2. Auer, S., L. Bühmann, C. Dirschl, O. Erling, M. Hausenblas, R. Isele, J. Lehmann, M. Martin, P.N. Mendes, B. van Nuffelen, C. Stadler, S. Tramp, und H. Williams. 2012. In *Managing the life-cycle of linked data with the LOD2 stack* Proceedings of International Semantic Web Conference (ISWC 2012)., 22. International
3. Auer, S., S. Dietzold, und T. Riechert. 2006. OntoWiki – A Tool for Social, Semantic Collaboration. In *The Semantic Web – ISWC 2006* 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006. Proceedings, Bd. 4273, Hrsg. I.F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, L. Aroyo, 736–749. Berlin/Heidelberg: Springer
4. Auer, S., J. Lehmann, A.-C.N. Ngomo, und A. Zaveri. 2013. Introduction to Linked Data and Its Lifecycle on the Web. In *Reasoning Web* Reasoning on the web in the Big Data Era – 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings. Hrsg. M. Koubarakis, B. Giorgos, I. Horrocks, P.G. Kolaitis, G. Lausen, G. Weikum, 1–99. Berlin/Heidelberg: Springer
5. Beckett, D., T. Berners-Lee. 2008. *Turtle – Terse RDF Triple Language*, W3C. http://www.w3.org/TeamSubmission/turtle/
6. Berners-Lee, T. July 2006. *Linked Data Design Issues*, W3C. http://w3.org/DesignIssues/LinkedData.html
7. Breslin, J.G., A. Harth, U. Bojars, und S. Decker. 2005. Towards semantically-interlinked online communities. In *The Semantic Web: Research and Applications*, Hrsg. L. Aroyo, P. Traverso, F. Ciravegna, 500–514. Berlin/Heidelberg: Springer
8. Carroll, J.J., C. Bizer, P. Hayes, und P. Stickler. 2005. Named graphs, provenance and trust. In *WWW2005*. ACM
9. Cyganiak, R., D. Reynolds, und J. Tennison. 2013. *The RDF Data Cube vocabulary. Technical report, W3C*. http://www.w3.org/TR/vocab-data-cube/
10. Demter, J., S. Auer, M. Martin, und J. Lehmann. 2012. LODStats – an Extensible Framework for High-performance Dataset Analytics. In *Proceedings of the EKAW 2012*, Lecture Notes in Computer Science (LNCS) 7603. Springer, 29
11. Erling, O. und I. Mikhailov. 2007. RDF Support in the Virtuoso DBMS. In *The Social Semantic Web 2007, Proceedings of the 1st Conference on Social Semantic Web (CSSW), September 26–28, 2007, Leipzig, Germany* Hrsg. S. Auer, C. Bizer, C. Müller, und A. V. Zhdanova. *LNI* 113:59–68. GI
12. Feigenbaum, L., G. T. Williams, K. G. Clark, und E. Torres. März 2013. SPARQL 1.1 Protocol. Technical report, W3C
13. Harris, S., und A. Seaborne. 2013. SPARQL 1.1 Query Language. Technical report, W3C
14. Heino, N., S. Dietzold, M. Martin, und S. Auer. 2009. Developing Semantic Web Applications with the Ontowiki Framework. In *Networked Knowledge – Networked Media* Studies in Computational Intelligence, Bd. 221, Hrsg. T. Pellegrini, S. Auer, K. Tochtermann, S. Schaffert, 61–77. Berlin/Heidelberg: Springer

15. Inkster, T., und K. Kjernsmo. 2009. *Named Graphs in RDFa (RDFa Quads)*. http://buzzword.org.uk/2009/rdfa4/spec

16. Krötzsch, M., D. Vrandecic, M. Völkel, H. Haller, und R. Studer. 2007. Semantic Wikipedia. *Journal of Web Semantics* 5: 251–261

17. Leuf, B., und W. Cunningham. 2001. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional

18. Martin, M., J. Unbehauen, und S. Auer. 2010. Improving the Performance of Semantic Web Applications with SPARQL Query Caching. In *Proceedings of 7th Extended Semantic Web Conference (ESWC 2010)* Heraklion, Crete, Greece, 30 May – 3 June 2010. Lecture Notes in Computer Science, Bd. 6089, Hrsg. L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, T. Tudorache, 304–318. Berlin/Heidelberg: Springer

19. McGuinness, D. L., und F. van Harmelen. 2004. OWL Web Ontology Language Overview

20. Miles, A., und S. Bechhofer. 2009. SKOS Simple Knowledge Organization System Reference

21. Rieß, C., N. Heino, S. Tramp, und S. Auer. 2010. EvoPat – Pattern-Based Evolution and Refactoring of RDF Knowledge Bases. In *Proceedings of the 9th International Semantic Web Conference (ISWC2010)* Lecture Notes in Computer Science. Berlin/Heidelberg: Springer

22. Seaborne, A., und G. Manjunath. 2008. *SPARQL/Update: A language for updating RDF graphs. Technical Report Version 5: 2008-04-29, Hewlett-Packard*. http://jena.hpl.hp.com/~afs/SPARQL-Update.html

23. Tramp, S., N. Heino, S. Auer, und P. Frischmuth. 2010. RDFauthor: Employing RDFa for collaborative Knowledge Engineering. In *Proceedings of the EKAW 2010 – Knowledge Engineering and Knowledge Management by the Masses* Lisbon, Portugal, 11th October–15th October 2010. Lecture Notes in Artificial Intelligence (LNAI), Bd. 6317, Hrsg. P. Cimiano, H. Pinto, 90–104. Berlin/Heidelberg: Springer

24. Tummarello, G., R. Delbru, und E. Oren. 2007. Sindice.com: Weaving the Open Linked Data. In *ISWC2007* LNCS, Bd. 4825 Springer

25. Unbehauen, J., C. Stadler, und S. Auer. 2012. Accessing relational data on the web with sparql-map. In *Semantic Technology* Second Joint International Conference, JIST 2012, Nara, Japan, December 2-4, 2012. Proceedings, Hrsg. H. Takeda, Y. Qu, R. Mizoguchi, Y. Kitamura, 65–80. Berlin/Heidelberg: Springer

Multilingualität und Linked Data

7

Philipp Cimiano und Christina Unger

Zusammenfassung

Linked Data ist eine flexible Technologie für die Repräsentation und Verlinkung von heterogenen Daten. Solche Daten können offene Daten sein, die im Kontext des Webs verfügbar sind, aber auch interne Unternehmensdaten, die miteinander über Organisationseinheiten und Strukturen hinweg verknüpft sind. In beiden Fällen werden diese Daten oft über Länder- und Sprachgrenzen hinweg erzeugt und benutzt. Das ist insbesondere der Fall für Unternehmen, die international präsent sind und dementsprechend Filialen in mehreren Ländern betreiben, oder für Organisationen, die international operieren. Dabei stellen sich zwei Fragen: Wie können Daten oder Prozesse, die auf Daten zugreifen, über Länder und Sprachen hinweg synchronisiert oder gar integriert werden? Und wie können Daten sprachübergreifend zugänglich gemacht werden? Für beide Zwecke ist es unabdingbar, dass Daten und Prozesse mit Informationen angereichert werden, wie sie in verschiedenen Sprachen verbalisiert werden.

In diesem Kapitel geben wir eine Übersicht über das Themenfeld Multilingualität und Linked Data, vor allem in Hinblick auf neuere Entwicklungen und deren mögliche Anwendungen. Nach einer kurzen Einführung und Motivation zeigen wir die Herausforderungen auf, die sich aus der Nutzung von Linked Data über Sprachgrenzen hinweg ergeben. Dann besprechen wir Verfahren, mit denen Datenschemas, die für verschiedene Länder entwickelt wurden, synchronisiert werden können, um die Aggregation und Integration von Daten über Länder und Sprachgrenzen hinweg zu ermöglichen. Darüber hinaus diskutieren wir, wie Linked Data mit linguistischen Informationen angereichert werden kann, und betrachten einige Anwendungen, die zeigen, wie solche Informationen für die Generierung und die Interpretation natürlicher Spra-

P. Cimiano ✉ · C. Unger
CITEC, Universität Bielefeld, Inspiration 1, 33619 Bielefeld, Deutschland
e-mail: cimiano@cit-ec.uni-bielefeld.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_7

che verwendet werden können, um einen sprachübergreifenden Zugang zu Linked Data zu ermöglichen.

7.1 Einführung

Die Internationalisierung nimmt in vielen Bereichen zu. Unternehmen sind immer stärker international tätig und pflegen Kundenbeziehungen oder Beziehungen zu anderen Unternehmen oder Lieferanten in verschiedenen Ländern. Auch im Rahmen der Politik nimmt die Notwendigkeit von internationalen Absprachen und die Einigung auf gemeinsame Regeln immer weiter zu, z. B. in Bereichen wie Import und Export, Einwanderung oder Transaktionssteuern.

In solchen Szenarien und Anwendungen müssen Prozesse und Regeln über Sprachgrenzen hinweg harmonisiert werden, Daten in verschiedenen Sprachen integriert und möglicherweise gemeinsame sprachübergreifende Taxonomien oder Terminologien entwickelt werden.

Das Überbrücken von Sprachbarrieren ist eine nicht-triviale Aufgabe. Durch die konsequente Verwendung von URIs und RDF als Datenmodell unterstützt zwar Linked Data prinzipiell die Integration von Daten, es bedarf aber dennoch Algorithmen, die Entsprechungen (*Alignments*) zwischen Daten in verschiedenen Sprachen finden können, sowie Verfahren, die Daten oder Vokabulare in eine bestimmte Sprache übersetzen. Letzteres wird in Analogie zur derselben Problematik in der Softwareentwicklung *Lokalisierung* genannt. Lokalisierung ist essentiell für Anwendungen, in denen Linked Data als Datenbasis verwendet wird und Sprecher unterschiedlicher Kulturen oder sprachlicher Kontexte mit diesen Daten interagieren oder sie abfragen können sollen.

In diesem Kapitel geben wir einen Überblick über Probleme und Fragestellungen, die sich aus der Notwendigkeit ergeben, auf Linked Data über Sprachen hinweg zugreifen zu können, sowie verlinkte Datensätze oder Vokabulare und Taxonomien, die zur Beschreibung von Daten verwendet werden, über Sprachgrenzen hinweg zu harmonisieren. Zuerst geben wir einen Überblick über die Behandlung von Multilingualiät in Linked Data. Wir besprechen dann insbesondere Verfahren zur Lokalisierung bzw. Übersetzung von Vokabularen, Ontologien und Taxonomien sowie für das sprach-übergreifende Alignment von Ontologien oder Vokabularen, die in unterschiedlichen Sprachen vorliegen. Insbesondere präsentieren wir das Lexikonformat *lemon* und diskutieren einige Anwendungen für *lemon*-angereicherte Linked-Data-Quellen.

7.2 Multilingualität in Linked Data

Für viele Anwendungen ist es wichtig, Linked Data mit Informationen darüber anzureichern, wie bestimmte Vokabularelemente in verschiedenen Sprachen ausgedrückt werden, insbesondere Individuen, Klassen und Eigenschaften.

Betrachten wir als Beispiel den fiktiven Fall eines Weinhändlers, der die von ihm angebotenen Weinsorten auf seiner Webseite veröffentlichen möchte. Er hat Kunden in Deutschland, im Vereinigten Königreich und in Spanien und möchte die Informationen daher in drei Sprachen anbieten. In RDF können Bezeichnungen für eine Klasse durch die Eigenschaft `label` ausgedrückt werden (Beispiele werden hier und im Folgenden immer in Turtle-Syntax angegeben, wobei `onto` ein Präfix ist, das die URI der Ontologie abkürzt):

```
onto:Rotwein rdf:type rdfs:Class ;
             rdfs:label "Rotwein"@de ;
             rdfs:label "red wine"@en ;
             rdfs:label "vino rojo"@es .
```

Gleiches gilt natürlich für Eigenschaften, die in einem Vokabular verwendet werden. Betrachten wir z. B. die Eigenschaft `hatPreis`, die den Preis eines Weines angibt. Auch diese Eigenschaft kann in mehreren Sprachen ausgedrückt werden:

```
onto:hatPreis rdf:type rdfs:Property ;
              rdfs:label "Preis"@de ;
              rdfs:label "price"@en ;
              rdfs:label "precio"@es .
```

Die URIs `onto:Rotwein` und `onto:hatPreis` sind dabei sprachunabhängige Identifikatoren, die für die Klasse der Rotweine bzw. für die Eigenschaft, einen bestimmten Preis zu haben, stehen. Der Einfachheit halber haben sie hier mnemonische Namen, es sollte aber beachtet werden, dass URIs nur beliebige Zeichenfolgen sind. Um auf Daten, die diese Identifizierer verwenden, in verschiedenen Sprachen zugreifen zu können, ist es daher wichtig, dass Labels in diesen Sprachen zur Verfügung stehen. Leider ist das Hinzufügen von multilingualen Labels bisher keine gängige Praxis im Semantic Web. Bisherige Untersuchungen haben gezeigt, dass lediglich 21 % der RDF-Literale einen Sprach-Tag haben. Darüber hinaus ist Englisch eindeutig die führende Sprache im Linked-Data-Web – in dem Sinne, dass sie 85 % aller Sprach-Tags ausmacht (siehe [14]).

Zusätzlich ergeben sich in der Praxis zwei Herausforderungen. Die erste entsteht dadurch, dass zuweilen über Länder hinweg verschiedene Taxonomien und Ontologien mit verschiedenen URIs verwendet werden. Es könnte zum Beispiel sein, dass ein weiterer fiktiver Weinhändler eine ähnliche Weinontologie wie die obige entwirft und dabei den Identifikator `RedWine` für die Klasse der Rotweine benutzt sowie eine Eigenschaft `price` definiert, die der Eigenschaft `hatPreis` entspricht. Um Daten, die in beiden Ontologien ausgedrückt werden, aggregieren zu können, müssen die entsprechenden Vokabulare aufeinander abgebildet werden. Dazu will man festhalten, dass bestimmte Klassen und Eigenschaften, obwohl sie verschiedene URIs haben, ein und dasselbe Konzept darstellen. Das ist möglich mit Hilfe der Eigenschaften `owl:equivalentClass` und `owl:equivalentProperty` (und analog `owl:sameAs` für Individuen):

```
onto:RedWine owl:equivalentClass    onto:Rotwein .
onto:price   owl:equivalentProperty onto:hatPreis .
```

In Abschn. 7.3 werden Verfahren genauer betrachtet, die Ontologien bzw. Vokabulare automatisch in mehrere Sprachen übersetzen, sowie Verfahren, die Vokabulare in verschiedenen Sprachen aufeinander abbilden.

Die zweite Herausforderung ergibt sich aus der Tatsache, dass die Konzeptualisierung von Daten in einer Ontologie und die natürlichsprachliche Konzeptualisierung der Welt nicht immer übereinstimmen. Dadurch sind in manchen Fällen linguistisch komplexe Informationen nötig, die nicht allein über Labels ausgedrückt werden können. Wenn man zum Beispiel angeben will, dass die Eigenschaft `Erzeuger`, die den Erzeuger eines Weines angibt, mit Hilfe des Verbes „anbauen“ verbalisiert werden kann, ist zusätzlich hilfreich anzugeben, dass das Tripel

```
onto:Eiswein onto:Erzeuger onto:Weingut_Grau .
```

durch die Sätze „Das Weingut Grau baut Eiswein an“ oder „Eiswein wird vom Weingut Grau angebaut“ ausgedrückt werden kann, nicht aber durch „Das Weingut Grau wird von Eiswein angebaut“.

Als weiteres Beispiel nehmen wir an, obiger Weinhändler möchte auf seiner Webseite Weine mit dem Adjektiv „preiswert“ beschreiben, die über die Eigenschaft `hatPreis` mit einem Wert bis zu 12 EUR (oder 10 Pfund für die britischen Kunden) verbunden sind. Die Klassen der preiswerten Weine lässt sich zwar als Restriktionsklassen (`owl:RestrictionClass`) definieren, aber gerade wenn man eine Ontologie nicht selber entwickelt und pflegt, sondern eine schon vorhandene nutzt, kommt es vor, dass man sprachliche Ausdrücke für Konstrukte braucht, die in der Ontologie nicht explizit definiert und benannt sind. Labels sind dann nicht ausreichend, da man das Label „preiswert“ weder der Eigenschaft `hatPreis` noch den Zahlenwerten 1–12 zuordnen kann.

In Abschn. 7.4 stellen wir daher *lemon* vor, ein Lexikonmodell für Ontologien, das es erlaubt, komplexe linguistische Informationen zu erfassen und (einfachen oder komplexen) Ontologiekonzepten zuzuordnen.

7.3 Ontologie-Lokalisierung und sprachübergreifendes Ontologie-Alignment

In diesem Abschnitt führen wir kurz in die Probleme der Ontologie-Lokalisierung und des Ontologie-Alignments ein. Im Bereich der Ontologie-Lokalisierung besprechen wir die unterschiedlichen Typen von Lokalisierungaktivitäten und verweisen auf aktuelle Entwicklungen und Prototypen, ohne jedoch auf technische Details einzugehen. Anschließend gehen wir auf das Ontologie-Alignment ein und beschreiben als Beispiel ein System,

das im Kontext des Monnet-Projektes entwickelt wurde, um Vokabulare für die Finanzberichterstattung aus unterschiedlichen Ländern Europas aufeinander abzubilden.

7.3.1 Ontologie-Lokalisierung

Laut Cimiano et al. [6] kann die Lokalisierung einer Ontologie wie folgt definiert werden:

> Ontologie-Lokalisierung bezeichnet den Prozess der Anpassung einer gegebenen Ontologie an die Anforderungen einer bestimmten Gemeinschaft, die durch eine gemeinsame Sprache, Kultur oder geo-politische Umgebung charakterisiert ist.

Die Aufgabe, eine Ontologie zu lokalisieren, ist analog zum Problem der Software-Lokalisierung zu verstehen. In der Software-Industrie müssen Software-Produkte an die kulturellen Gegebenheiten und auch an die Sprache der Nutzer angepasst werden. In erster Linie muss die Dokumentation und auch die graphische Oberfläche übersetzt werden. In einigen Fällen muss aber auch die Funktion der Software angeglichen werden, zum Beispiel wenn sich die Anforderungen von Nutzern verschiedener sprachlicher und kultureller Kontexte unterscheiden. Im ersten Fall bleibt die eigentliche Funktion der Software unberührt und die Anpassung findet nur oberfächlich statt, nämlich nur an den Stellen, wo eine Interaktion mit dem Nutzer stattfindet. Im zweiten Fall sind tiefergehende Anpassungen an einem Softwareprodukt notwendig.

Ähnlich ist die Situation bei der Lokalisierung von Ontologien. In einigen Fällen ist es ausreichend, die *lexikalische Ebene* zu übersetzen. Praktisch bedeutet das, dass lediglich die Labels der entsprechenden Klassen, Individuen oder Eigenschaften übersetzt werden. In anderen Fällen ist eine tiefergehende Anpassung der *konzeptuellen Ebene* notwendig, d. h. es müssen Begriffe neu definiert werden oder gar neu eingeführt werden, um die Ontologie an die Gegebenheiten einer anderen Kultur anzupassen.

Eine weitere wichtige Dimension ist der Zweck, für den die Ontologie angepasst wird. Man unterscheidet hier zwischen einer *funktionalen Anpassung* und einer *beschreibenden Anpassung*. Im Falle der funktionalen Anpassung muss die angepasste Ontologie für die Zielgemeinschaft den gleichen Zweck bzw. die gleiche Funktion erfüllen, welche die ursprüngliche Ontologie für die kulturelle Umgebung erfüllt, für die sie entwickelt wurde. Im Falle einer Anpassung zu beschreibenden Zwecken ist das Ziel, die Ontologie für die Zielgemeinschaft zugänglich zu machen, indem die Begriffe in der Sprache der Zielgemeinschaft beschrieben werden. Betrachten wir als Beispiel den Fall einer Ontologie, die politische Ämter modelliert. Eine deutsche Ontologie würde Begriffe wie „Staatsoberhaupt“ oder „Regierungschef“ definieren sowie die entsprechenden Unterbegriffe „Bundespräsident“ und „Bundeskanzler“. Falls nun einer anderen kulturellen Gemeinschaft, zum Beispiel Sprechern des angelsächsischen Sprachraums, Zugang zu dieser Ontologie gegeben werden soll, zum Beispiel um das politische System Deutschlands zu verstehen, würde es ausreichen die obigen Begriffe wörtlich zu übersetzen, wie in folgender Tabelle angegeben:

DE	EN	ES
Staatsoberhaupt	Head of State	Jefe de Estado
Regierungschef	Head of Government	Jefe de Gobierno
Bundespräsident	President	Presidente
Bundeskanzler	Federal Chancellor	Canciller Federal

Falls aber die Ontologie zu funktionalen Zwecken übersetzt wird, müssen die Begriffe durch funktional äquivalente Begriffe aus der Zielgemeinschaft ersetzt werden, zum Beispiel wie folgt:

DE	EN	ES
Staatsoberhaupt	Head of State	Jefe de Estado
Regierungschef	Head of Government	Jefe der Gobierno
Bundespräsident	Queen/King	Rei/Reina
Bundeskanzler	Prime Minister	Presidente

Dabei reicht es in der Regel nicht aus lediglich den Konzepten Labels in einer anderen Sprachen zu geben, da sich die Konzepte inhaltlich unterscheiden. Die Begriffe „Bundeskanzler“ und „Prime Minister“ zum Beispiel beschreiben Ämter, zwischen denen es länderspezifische Unterschiede hinsichtlich der Befugnisse, der Rolle und des Amtsverständnisses gibt. Also wird bei der Lokalisierung das Konzept „Bundeskanzler“ durch ein neues Konzept „Prime Minister“ ersetzt, welches der Welt der Zielgemeinschaft entspricht.

In einigen Fällen müssen Konzepte nicht nur durch neue ersetzt, sondern auch verfeinert oder verallgemeinert werden. Betrachten wir den Begriff „Fluss“ im Deutschen bzw. „river“ im Englischen. Bei der Anpassung dieses Begriffes für eine französischsprechende Gemeinschaft muss beachtet werden, dass im Französischen die Unterscheidung gemacht wird zwischen „rivière“, einem Fluss, der in einen anderen Fluss mündet, und einem „fleuve“, einem Fluss, der in ein Meer mündet. Diese Unterscheidung führt dazu, dass bei der Lokalisierung einer deutschen oder englischen Ontologie der Begriff „Fluss“ bzw. „river“ entsprechend dadurch verfeinert werden muss, dass Unterklassen für „fleuve“ und „rivière“ eingeführt werden. Die entgegengesetzte Situation besteht dann bei der Lokalisierung einer französischen Ontologie für einen deutschsprachigen oder englischsprachigen Kontext. In diesem Fall können die Unterklassen für „fleuve“ und „rivière“ entfernt werden. Alternativ dazu kann man auch die Unterscheidung belassen und entweder keine Lexikalisierung der Unterklassen in der Zielsprache angeben oder die allgemeineren Begriffe „Fluss“ und „river“ als Labels für beide Unterklassen verwenden.

In den letzten Jahren sind verschiedene Methoden und Werkzeuge entwickelt worden, welche die Lokalisierung einer Ontologie unterstützen. Im Rahmen des NeOn-Projektes[1], zum Beispiel, wurde der *LabelTranslator* entwickelt [19], ein regelbasiertes System, das eine Vielzahl von Übersetzungsquellen (auch solche, die im Web verfügbar sind) konsul-

[1] Siehe: http://www.neon-project.org, aufgerufen am 17.03.2014.

tiert um passende Übersetzungen zu ermitteln und verschiedene Übersetzungsalternativen zu gewichten. Der *LabelTranslator* wurde in das *Neon Toolkit*[2] integriert. Im Rahmen des Monnet-Projektes wurden desweiteren Werkzeuge entwickelt, die Verfahren der statistischen maschinellen Übersetzung anwenden, wie sie zum Beispiel auch in *Google Translate* oder *Bing Translate* verwendet werden, um Wahrscheinlichkeiten für verschiedene Übersetzungskandidaten zu ermitteln. Dieses Übersetzungsmodul wurde in das *Be Informed Studio*[3] integriert (siehe [8]), ein Modellierungswerkzeug der niederländischen Firma *Be Informed*.

7.3.2 Ontologie-Alignment

In vielen Anwendungsfällen sind tatsächlich verschiedene Ontologien für unterschiedliche sprachliche Gemeinschaften vorhanden. Für viele Zwecke ist es nötig, diese unterschiedlichen Ontologien aufeinander abzubilden, um zum Beispiel die Integration und Interoperabilität von Daten zu gewährleisten. Nehmen wir zum Beispiel einen Analysten, der den Jahresumsatz, das Kapital und die Liquidität verschiedener IT-Firmen in Europa vergleichen möchte. Zwar sind in den meisten Ländern Unternehmen verpflichtet ihre Kennzahlen jährlich offenzulegen, allerdings verwenden verschiedene Länder unterschiedliche Konzeptualisierungen und Vokabulare – eine Situation, die den Vergleich, Integration und Aggregation der Daten deutlich erschwert. Zum Beispiel wird in Deutschland die GAAP-Taxonomie des Handelsgesetzbuches verwendet und in Italien die *Tassonomia relativa ai Principi Contabili Italiani*. Ohne eine Abbildung der verschiedenen Taxonomien aufeinander können Daten nur sehr schwer integriert und verglichen werden. Das Ontologie-Alignment hat daher zum Ziel, Begriffe verschiedener Taxonomien aufeinander abzubilden. In multilingualen Kontexten kann man nach Spohr et al. [22] folgende Fälle unterscheiden:

- **Einsprachiges (monolinguales) Ontologie-Alignment:** Die Ontologien, die aufeinander abgebildet werden sollen, benutzen eine gemeinsame Sprache, die für das Alignment genutzt werden kann.
- **Mehrsprachiges (multilinguales) Ontologie-Alignment:** Die Ontologien, die aufeinander abgebildet werden sollen, haben mehrere Sprachen gemein, so dass auch Übereinstimmungen zwischen den Labels in verschiedenen Sprachen für das Alignment verwendet werden können.
- **Sprachübergreifendes (crosslinguales) Ontologie-Alignment:** Die Ontologien, die aufeinander abgebildet werden sollen, teilen keine Sprache. In diesem Fall werden die Labels der einen Ontologie in die Sprache der anderen Ontologie übersetzt, oder die Labels beider Ontologien werden in eine dritte Sprache (eine sogenannte *Pivot-Sprache*) übersetzt.

[2] Siehe: http://neon-toolkit.org/wiki/Main_Page, aufgerufen am 17.03.2014.
[3] Siehe: http://www.beinformed.nl/BeInformed/website/en/EN/Studio, aufgerufen am 17.03.2014.

Im Rahmen des Monnet-Projektes haben Spohr et al. [22] ein Verfahren für das Alignment von verschiedenen Ontologien entwickelt, das in den drei oben genannten Szenarien eingesetzt werden kann. Dazu werden einerseits statistische maschinelle Übersetzungsdienste wie *Bing Translate*[4] benutzt, um Labels in verschiedene Sprachen zu übersetzen. Andererseits basiert das Verfahren auf maschinellen Lernverfahren, die anhand gegebener Beispielabbildungen eine lineare Gewichtung verschiedener Merkmalsindikatoren lernen. Mit Hilfe dieser kann dann, gegeben ein Konzept aus einer Taxonomie, das passendste Konzept aus einer anderen Taxonomie bestimmt werden. Dabei werden folgende Indikatoren verwendet:

- **Ähnlichkeit der Labels auf der Ebene der Zeichenkette**: Hier werden zum einen Ähnlichkeitsmaße verwendet, welche die Reihenfolge der Wörter in einem Label betrachten (z. B. Levensthein-Distanz), und zum anderen solche, die die Reihenfolge ignorieren (z. B. Kosinusähnlichkeit). Die Ähnlichkeitswerte werden über die verschiedenen Labels in den verschiedenen Sprachen aggregiert, da Konzepte sogar innerhalb einer Sprache verschiedene Labels haben können.
- **Strukturelle Merkmale** nutzen die taxonomische Struktur, insbesondere die Ober- und Unterbegriffe des abzubildenden Begriffes, um diese mit den Ober- bzw. Unterbegriffen eines Kandidatenkonzeptes zu vergleichen. Im Falle der Reporting-Taxonomien, die im Rahmen der beschriebenen Fallstudie betrachtet wurden, werden taxonomische Beziehungen verwendet, um rekursiv zu spezifizieren, wie bestimmte Größen, zum Beispiel die Liquidität eines Unternehmens, aus anderen Größen berechnet werden. Denn die Information darüber, wie bestimmte Kennzahlen aus anderen Kennzahlen berechnet werden, ist entscheidend bei der Frage, ob zwei Begriffe wirklich äquivalent sind. Um solche Informationen zu vergleichen, wird sowohl die Anzahl der verschiedenen Kennzahlen verglichen, aus denen sich beide Begriffe zusammensetzen, als auch die Labels dieser Kennzahlen.

Diese Indikatoren und ihre Berechnung sowie das Verfahren für das Training der *Support Vector Machines* ist detailliert in Spohr et al. [22] beschrieben. Wir abstrahieren an dieser Stelle von technischen Details und gehen nur auf die Anwendung im Kontexts des Alignments von Business-Reporting-Taxonomien ein. Insbesondere wurde das Verfahren auf die folgenden drei Taxonomien angewendet:

- Die *XEBR Kerntaxonomie* wurde von der *XBRL Europe Business Registers Working Group*[5] entwickelt und umfasst 269 Buchhaltungskonzepte, die in vielen nationalen Taxonomien vorkommen und mit englischen Labels versehen sind.
- Die Taxonomie *Principi Contabili Italiani*[6] (ITCC) aus dem Jahr 2011 umfasst 444 Begriffe mit englischen, italienischen, französischen und deutschen Labels.

[4] Siehe: http://www.bing.com/translator, aufgerufen am 17.03.2014.
[5] Siehe: http://www.xbrleurope.org/working-groups/xebr-wg, aufgerufen am 17.03.2014.
[6] Siehe: http://www.xbrl.org/it, aufgerufen am 17.03.2014.

- Die *GAAP-Taxonomie des Deutschen Handelsgesetzbuches*[7] (HGG) aus dem Jahre 2011 umfasst 3146 Begriffe mit englischen und deutschen Labels.

Die Ergebnisse für die verschiedenen Szenarien (monolingual, multilingual und sprachübergreifend), die Spohr et al. [22] berichten, lassen folgende Schlussfolgerungen zu: In den meisten Fällen verbessert die Verwendung verschiedener Sprachen die Ergebnisse. Auch die Verwendung von strukturellen Informationen verbessert die Qualität der Ergebnisse, im Allgemeinen um ungefähr 5 %. Und selbst für den sprachübergreifenden Fall, in dem keine Labels in einer gemeinsamen Sprache vorhanden sind und die Übersetzung automatisch in verschiedenen Sprachen erfolgt, sind die Ergebnisse positiv: In 50 % der Fälle ist das korrekte Konzept an der ersten Position des Rankings, in über 70 % der Fälle unter den ersten 5, und in fast 80 % der Fälle unter den ersten 10. Ein Experte müsste also pro Begriff eine kleine Anzahl von Vorschlägen manuell überprüfen, was den Aufwand und die benötigten Ressourcen, um zwei Taxonomien aufeinander abzubilden, deutlich senkt.

Eine Übersicht über das Problem des Ontologie-Alignments sowie den gängigen Ansätzen dafür geben Euzenat und Shvaiko [10]. Einen genaueren Überblick über Verfahren im Bereich des cross-lingualen Ontologie-Alignments geben Trojahn et al. [24]. Ein umfassende Einführung in das Thema bietet außerdem das Buch *Ontology Matching* [10].

7.4 Das *lemon*-Modell und Anwendungen

In diesem Abschnitt wollen wir eine kurze Einführung in das Lexikonmodell *lemon* geben, das erlaubt, lexikalische Informationen mit Ontologieelementen zu verknüpfen. Anschließend werden wir einige Anwendungen betrachten, die zeigen, wie ein solches Lexikon für die Generierung und die Interpretation natürlicher Sprache verwendet werden kann.

7.4.1 Das *lemon*-Modell

Das Lexikonmodell *lemon* (*Lexicon Model for Ontologies*) ist ein Modell für die Repräsentation lexikalischer Informationen in Bezug auf eine Ontologie. Das umfasst zum einen morphologische und syntaktische Informationen zu Wortklasse, Wortformen sowie Anzahl und Art der Argumente, zum anderen aber auch semantische Informationen zur Bedeutung von Wörtern und Phrasen in Bezug auf eine Ontologie. Diese Informationen werden in RDF ausgedrückt, so dass ein Lexikon als Linked Data veröffentlicht und geteilt werden kann.

Der Kern des *lemon*-Modells, dargestellt in Abb. 7.1, umfasst folgende Elemente:

[7] Siehe: http://www.xbrl.de, aufgerufen am 17.03.2014.

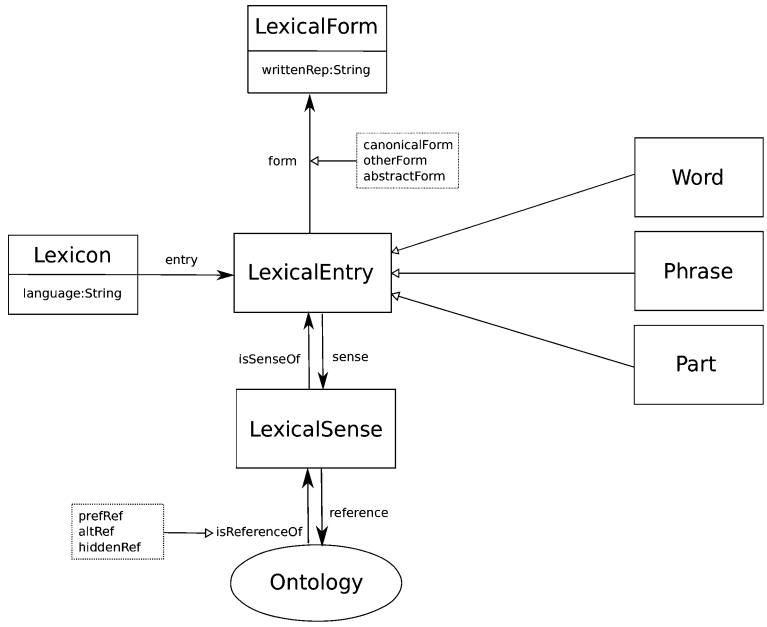


Abb. 7.1 Der Kern des *lemon*-Modells: Pfeile mit gefüllter Spitze repräsentieren Eigenschaften und Pfeile mit leerer Spitze geben Unterklassen und Untereigenschaften an

- Ein *Lexikon* (*lexicon*) ist eine Menge lexikalischer Einträge in einer bestimmten Sprache. Für eine Ontologie können natürlich mehrere Lexika in verschiedenen Sprachen entwickelt werden.
- Ein *lexikalischer Eintrag* (*lexical entry*) ist in der Regel ein Wort (z. B. „Wein“) oder eine Phrase (z. B. „nicht-physischer Vermögenswert im Eigentum des Unternehmens“).
- Eine *Form* (*form*) stellt die sprachliche Realisierung eines lexikalischen Eintrags dar, üblicherweise die geschriebene Form (*written representation*), aber möglicherweise auch eine phonetische Darstellung.
- Eine *Referenz* ist eine Entität in der Ontologie, d. h. eine Klasse, Eigenschaft oder Entität, und der Kern der Bedeutung eines lexikalischen Eintrags in Bezug auf die Ontologie.
- Die *Bedeutung* (*sense*) eines Eintrages besteht aus einer Referenz, möglicherweise zusammen mit Angaben zu Einschränkungen im Gebrauch eines Wortes oder anderen pragmatischen Informationen.

Zum Beispiel kann man für die Weinhändler-Ontologie aus Abschn. 7.2 ein deutsches Lexikon wie folgt definieren:

```
1  @prefix onto:  <http://example.org/ontology#> .
2  @prefix lex:   <http://example.org/lexicon#> .
3  @prefix lemon: <http://www.lemon-model.net/lemon#> .
4
5  lex:lexicon a lemon:Lexicon ;
6                lemon:language "de" ;
7                lemon:entry lex:Rotwein,
8                            lex:Preis,
9                            lex:anbauen,
10                           ...  .
```

Das Tripel in Zeile 5 legt die URI für das Lexikon fest und definiert es als Individuum vom Typ `Lexicon`, in Zeile 6 wird die Sprache des Lexikons angegeben, Deutsch, und schließlich werden in den Zeilen 7 und 10 die einzelnen lexikalischen Einträge aufgelistet.

Die Bedeutung eines lexikalischen Eintrags wird durch Referenz auf eine spezifische Klasse, Eigenschaft oder ein bestimmtes Individuum in der Ontologie definiert. Um zum Beispiel eine sprachliche Realisierung der Klasse `Rotwein` anzugeben, kann man folgenden lexikalischen Eintrag anlegen:

```
1  lex:Rotwein a lemon:Word ;
2   lemon:canonicalForm [ lemon:writtenRep "Rotwein"@de ] ;
3   lemon:sense         [ lemon:reference  onto:Rotwein ] .
```

Er spezifiziert, dass die Grundform (*canonical form*) von dem lexikalischen Eintrag die geschriebene Form „Rotwein“ hat, eine Zeichenkette, die mit dem Sprachtag `de` versehen ist, und dass die Bedeutung des Eintrags auf die ontologische Klasse `Rotwein` referiert.

Wichtig ist, dass *lemon* ein Modell ist, dass die Struktur eines Lexikons beschreibt, jedoch keinerlei linguistische Kategorien vorschreibt, also zum Beispiel kein Vokabular bereitstellt, um morphosyntaktische Eigenschaften wie Wortklasse, Wortformen usw. zu erfassen. Zu diesem Zweck kann das Vokabular einer beliebigen linguistischen Ontologie importiert werden. In diesem Kapitel verwenden wir die Ontologie LexInfo [4][8], die über 600 spezifische linguistische Kategorien und Eigenschaften umfasst. Damit kann der Eintrag für Rotwein zum Beispiel wie folgt erweitert werden:

```
1  lex:Rotwein a lemon:Word;
2    lexinfo:partOfSpeech lexinfo:noun;
3    lemon:canonicalForm [ lemon:writtenRep "Rotwein"@de;
4                          lexinfo:number lexinfo:singular ];
5    lemon:otherForm     [ lemon:writtenRep "Rotweine"@de;
6                          lexinfo:number lexinfo:plural ];
7    lemon:sense         [ lemon:reference onto:Rotwein ].
```

[8] Siehe: http://www.lexinfo.net/ontology/2.0/lexinfo, aufgerufen am 17.03.2014.

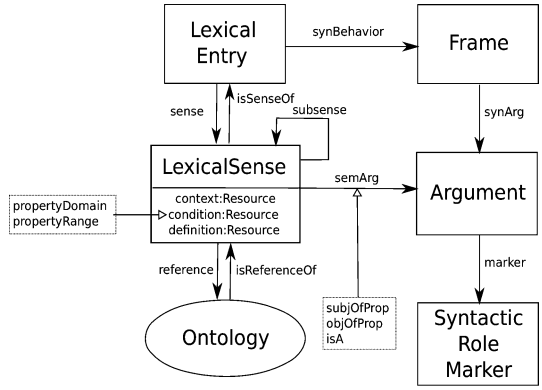


Abb. 7.2 Das Syntax-und-Argumentrollen-Modul von *lemon*

Zusätzlich zu den Kernklassen und -eigenschaften enthält *lemon* verschiedene Module, die weitere Aspekte der Lexikon-Ontologie-Schnittstelle beschreiben, aber nicht für jedes Lexikon relevant sind. Eines dieser Module ist das *Syntax-und-Argumentrollen-Modul*, dargestellt in Abb. 7.2, das zusätzliche Klassen und Eigenschaften definiert, welche die Verbindungen zwischen syntaktischen Konstruktionen und semantischen Prädikaten erfassen. Ein sogenannter *Frame* gibt den syntaktischen Kontext an, in dem ein Eintrag vorkommen kann, insbesondere die erforderlichen syntaktischen *Argumente*. Argumente sind häufig markiert, wobei ein *Argumentrollenmarker* (*syntactic role marker*) entweder ein eigener lexikalischer Eintrag sein kann, zum Beispiel eine Präposition, oder eine syntaktische Eigenschaft wie Kasus.

Zum Beispiel würde ein transitiver Verbrahmen festlegen, dass ein Subjekt und ein direktes Objekt erforderlich sind. Da die spezifischen Frames und Argumentrollen nichts mit der Struktur des Lexikoneintrags zu tun haben, sondern bestimmte linguistische Entscheidungen betreffen, wird das Vokabular dafür nicht von *lemon* zur Verfügung gestellt, sondern muss erneut aus einer geeigneten linguistischen Ontologie importiert werden.

Nehmen wir als Beispiel die Eigenschaft `Erzeuger` der Weinhändler-Ontologie. Dafür hatten wir in Abschn. 7.2 das Label „anbauen“ angegeben und die Schwierigkeit entdeckt, dass aus dem Label selber nicht ersichtlich ist, welchen semantischen Argumenten die syntaktischen Argumente entsprechen. Genau das lässt sich nun einfach in einem Lexikoneintrag erfassen:

```
lex:anbauen a lemon:Word;
  lexinfo:partOfSpeech lexinfo:verb;
  lemon:canonicalForm [ lemon:writtenRep "anbauen"@de ];
  lemon:synBehavior   [ a lexinfo:TransitiveFrame;
                        lexinfo:Subject     _:arg1;
                        lexinfo:DirecObject _:arg2;
  lemon:sense         [ lemon:reference onto:Erzeuger;
                        lemon:subjOfProp    _:arg2;
                        lemon:objOfProp     _:arg1 ].
```

Der Eintrag hat die Wortform Verb, die kanonische Form „anbauen“ und eine Bedeutung, die auf die Eigenschaft `Erzeuger` in der Ontologie referiert. Zusätzlich zu der Referenz auf die Eigenschaft werden die semantischen Argumente dieses Prädikats benannt, `_:arg2` als semantisches Subjekt und `_:arg1` als semantisches Objekt. Dieselben Argument-URIs werden auch bei der Angabe des syntaktischen Kontextes verwendet, in dem der Eintrag vorkommen kann. Der Rahmen wird als `TransitiveFrame`, also der eines transitiven Verbes, festgelegt, mit zwei Argumenten: einem Subjekt, `_:arg1`, das damit dem semantischen Objekt entspricht, und einem direkten Objekt, `_:arg2`, das damit dem semantischen Subjekt entspricht. Damit ist festgelegt, dass das Tripel *x* `onto:Erzeuger` *y* sprachlich als „*y* baut *x* an“ und nicht als „*x* baut *y* an“ realisiert wird.

Ein weiteres Beispiel, das wir im Abschn. 7.2 nicht mit Hilfe von Labels erfassen konnten, ist die Bezeichnung „preiswert“ für Weine, deren Preis im Bereich von bis zu 12 EUR bzw. 10 Pfund liegt. Die Klasse aller Entitäten mit einem Preis-Wert zwischen 0 und 12 lässt sich in OWL[9] als Restriktionsklasse wie folgt definieren:

```
lex:PreisBis12 a owl:RestrictionClass;
  owl:onProperty onto:Preis;
  owl:allValuesFrom [ a rdfs:Datatype;
                      owl:onDatatype xsd:double;
                      owl:withRestrictions
                      ( [ xsd:minInclusive  "0.00" ]
                        [ xsd:maxInclusive "12.00" ] ) ].
```

Diese Klasse ist möglicherweise nicht in der Ontologie definiert, kann aber mit Hilfe des Vokabulars der Ontologie definiert werden. Man kann also sagen, dass sie implizit Teil der Ontologie ist, jedoch in der Ontologie nicht explizit konstruiert und benannt ist. Dieses explizite Konstruieren und Benennen kann nun im Lexikon passieren, was uns erlaubt, einen einfachen Adjektiveintrag für „preiswert“ anzugeben, der auf obige Restriktionsklasse referiert:

```
lex:preiswert a lemon:Word;
  lexinfo:partOfSpeech lexinfo:adjective;
  lemon:canonicalForm [ lemon:writtenRep "preiswert"@de ];
  lemon:synBehavior   [ a lexinfo:AdjectiveFrame;
                        lemon:synArg _:arg;
  lemon:sense         [ lemon:reference lex:PreisBis12;
                        lemon:isA     _:arg ].
```

Das semantische Argument `isA` steht dabei für ein beliebiges Element der referierten Klasse.

[9] OWL ist eine Semantic-Web-Sprache zur Repräsentation von Ontologien. Für eine Beschreibung der aktuellen Version, OWL 2, siehe http://www.w3.org/TR/owl2-primer/, aufgerufen am 17.03.2014.

Zusätzlich können in *lemon* Einschränkungen für den Gebrauch eines Eintrages angegeben werden. Zum Beispiel kann es sein, dass der Weinhändler nicht nur Weine, sondern auch Weinzubehör wie Gläser und Dekanter anbietet, und die Ontologie eine einzige Eigenschaft `Hersteller` sowohl für den Erzeuger eines Weines als auch den Hersteller von Weinzubehör benutzt. In diesem Fall sollte diese Eigenschaft als „anbauen“ (oder „erzeugen“) lexikalisiert werden, falls es sich um Wein handelt, aber als „herstellen“, falls es sich um Zubehör handelt. Dazu kann man die Bedeutung eines lexikalischen Eintrags mit einer Bedingung wie `propertyDomain` bzw. `propertyRange` einschränken, die jeweils festlegen, dass das Subjekt bzw. Objekt einer Eigenschaft von einem bestimmten Typ sein muss, in unserem Fall vom Typ `Wein` oder `Zubehoer`.

```
lex:anbauen a lemon:Word;
    lemon:sense [ lemon:reference       onto:Hersteller;
                  lemon:propertyDomain  onto:Wein ].

lex:herstellen a lemon:Word;
    lemon:sense [ lemon:reference       onto:Hersteller;
                  lemon:propertyDomain  onto:Zubehoer ].
```

Weiterhin würde nun die Bedeutung des Wortes „preiswert“ variieren, je nachdem, ob es sich um Wein oder ein bestimmtes Zubehör handelt. Man kann also parallel zu `PreisBis12` weitere Restriktionklassen `PreisBis30` usw. definieren sowie für den Adjektiveintrag mehrere Bedeutungen angeben, deren Gebrauch jeweils eingeschränkt ist:

```
lex:preiswert a lemon:Word ;
    lemon:sense [ lemon:reference onto:PreisBis12;
                  lex:usedFor onto:Wein ],
                [ lemon:reference onto:PreisBis30;
                  lex:usedFor onto:Dekanter ].

lex:usedFor rdfs:subProperty lemon:condition.
```

Die Einschränkung `usedFor` ist in diesem Fall eine von uns definierte. Ähnliche Einschränkungen sind nützlich, wenn man Lexikalisierungen personifizieren will, wenn man also zum Beispiel bei der Generierung von Texten für fortgeschrittene Endnutzer Fachbegriffe verwenden, für neue Nutzer hingegen einfachere Begriffen verwenden will (siehe z. B. [5]).

Details zu *lemon* sowie weitere Informationen zu den verschiedenen Modulen und zur Modellierung verschiedener lexikalischer Aspekte können im *lemon Cookbook*[10] nachgelesen werden. Außerdem dient *lemon* derzeit als Basis für die Aktivitäten der W3C

[10] Siehe: http://lemon-model.net/learn/cookbook.php, aufgerufen am 17.03.2014.

Ontology Lexica Community Group[11], die ein Standard-Modell für die Anreicherung von Ontologien mit lexikalischen Informationen entwickelt.

7.4.2 Lexika als Linked Data

Da Ontologielexika in RDF, also einer Standard-Semantic-Web-Sprache, repräsentiert werden und auf Elemente einer Ontologie verweisen, stellen sie selber Linked Data dar und können genauso wie Ontologien veröffentlicht und geteilt werden. Außerdem kann *lemon* als eine Art Austauschformat dienen, um lexikalische Ressourcen wie zum Beispiel *WordNet*[12] und *Wiktionary*[13] als Linked Data verfügbar zu machen (siehe [18]).

Schließlich kann man sich vorstellen, dass sich ein Ökosystem von Ressourcen entwickelt, das Ontologien, Lexika für diese Ontologien in unterschiedlichen Sprachen, lexikalische Ressourcen und möglicherweise auch Werkzeuge für die semi-automatische Entwicklung von Lexika sowie für die Generierung von verschiedenen Grammatikformaten aus Lexika umfasst.

7.4.3 Automatisches Erzeugen von Grammatiken

Die manuelle Entwicklung von Grammatiken ist ein ressourcenintensiver Prozess. Besonders bei großen Domänen ist es aufwendig Grammatiken zu konstruieren, sie weiterzuentwickeln und schließlich auch auf andere Sprachen zu portieren.

Ontologielexika können dabei helfen den Prozess der Grammatikgenerierung zu automatisieren. Zum einen erlauben sie sehr reichhaltige linguistische Informationen auszudrücken, zum anderen repräsentieren sie diese Informationen auf eine kompakte und theorieneutrale Weise und abstrahieren damit von bestimmten Grammatiktheorien. *lemon*, zum Beispiel, ist weitestgehend unabhägig von bestimmten syntaktischen und semantischen Theorien und erst mit der Wahl einer importierten linguistischen Ontologie legt man sich auf bestimmte linguistische Kategorien fest. Darüber hinaus wird die Bedeutung lexikalischer Einheiten in Bezug auf eine Ontologie angegeben, wodurch das Generieren von semantischen Repräsentationen erleichtert wird, die sich mit der Struktur und dem Vokabular einer bestimmten Ontologie im Einklang befinden.

In früheren Arbeiten haben wir die Generierung von Grammatiken aus *lemon*-Lexika für verschiedene Arten von lexikalisierten Grammatikformaten implementiert, unter anderem für *Lexicalized Tree Adjoining Grammars* (LTAG) und *Grammatical Framework* (GF). Dazu werden zuerst für jeden lexikalischen Eintrag alle nötigen Informationen extrahiert: die Wortklasse, die verschiedene Wortformen, der syntaktische Frame sowie die

[11] Siehe: http://www.w3.org/community/ontolex/, aufgerufen am 17.03.2014.
[12] Siehe: http://wordnet.princeton.edu, aufgerufen am 17.03.2014.
[13] Siehe: https://www.wiktionary.org, aufgerufen am 17.03.2014.

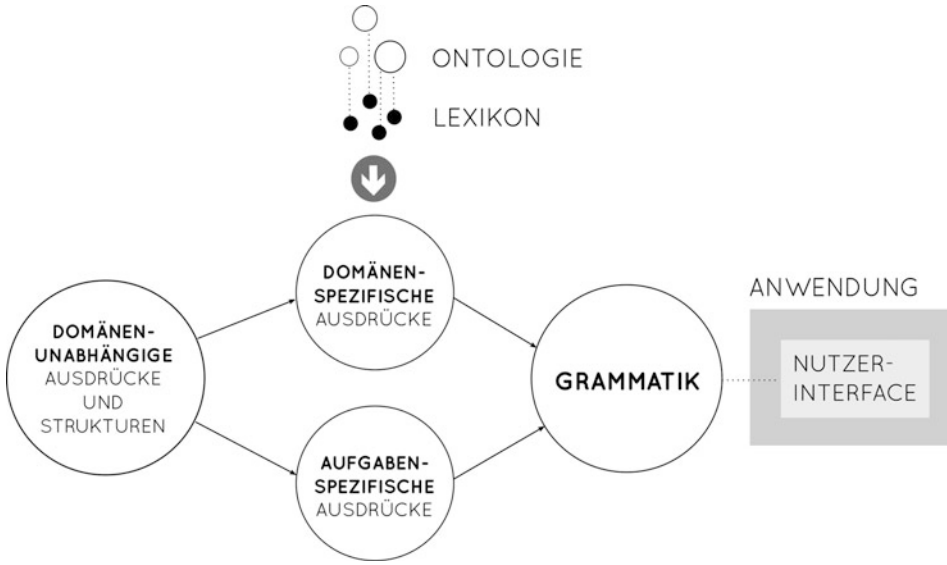


Abb. 7.3 Grammatikmodularität

Anzahl und Art der syntaktischen Argumente, die Bedeutung mit vorhandenen semantischen Argumenten sowie deren Entsprechung zu den syntaktischen Argumenten. Da ein *lemon*-Lexikon in RDF-Format vorliegt, können diese Information mit Hilfe der Abfragesprache SPARQL[14] abgefragt werden. Danach werden basierend auf dem syntaktischen Frame oder der Wortklasse Templates gefüllt, welche die allgemeine Form des Eintrags in dem bestimmten Grammatikformat spezifizieren.

Eine Implementierung dieser Methoden ist auf Bitbucket[15] verfügbar. Details zu den Grammatikformalismen sowie zur Rolle von Ontologielexika und aus Lexika generierten Grammatiken für die Interpretation natürlicher Sprache sind in dem Buch *Ontology-based interpretation of natural language* [7] zusammengefasst.

Wichtig anzumerken ist, dass die generierten Grammatiken nur domänenspezifische Ausdrücke umfassen, aber keinerlei domänenunabhängige Ausdrücke, wie Determinierer, Pronomen, Hilfsverben und Ausdrücke für Negation oder Koordination, da diese keine Entsprechung in der Ontologie haben und damit auch nicht Bestandteil des Lexikons sind. Eine aus einem Lexikon erzeugte Grammatik stellt also nur ein Grammatikmodul dar, das in eine umfassendere Grammatik eingebettet werden muss, um in Anwendungen nützlich zu sein. Eine solche umfassendere Grammatik könnte wie in Abb. 7.3 organisiert sein. Das heißt, es gibt ein Modul, das domänenunabhängige Ausdrücke und Satzstrukturen umfasst. Dieses Modul wird üblicherweise manuell konstruiert und kann dann für jede

[14] Siehe: http://www.w3.org/TR/rdf-sparql-query/, aufgerufen am 17.03.2014.
[15] Siehe: https://bitbucket.org/chru/lemongrass, aufgerufen am 17.03.2014.

Domäne wiederverwendet werden. Darauf aufbauend gibt es zum einen ein Modul mit domänenspezifischen Ausdrücken, das automatisch aus dem Ontologielexikon generiert wird, und gegebenenfalls ein Modul mit aufgabenspezifischen, zum Beispiel dialogrelevanten, Ausdrücken. Zusammen bilden die Module eine vollständige Grammatik, die in einer Anwendung verwendet werden kann, zum Beispiel in einer natürlichsprachlichen Schnittstelle zum Endnutzer.

Will man die Grammatik auf eine neue Domäne portieren, so genügt es, die Ontologie und das Ontologielexikon auszutauschen und daraus automatisch ein neues domänenspezifisches Modul zu erzeugen. Da mit einer Ontologie mehrere Lexika in verschiedenen Sprachen verknüpft sein können, können außerdem für eine Domäne Grammatikmodule in mehreren Sprachen erzeugt werden. Solch ein multilingualer Kontext setzt allerdings voraus, dass auch die anderen Grammatikmodule in den verwendeten Sprachen vorliegen.

In den folgenden Abschnitten zeigen wir zwei Anwendungsmöglichkeiten für ontologiebasierte Grammatiken: natürlichsprachliche Generierung, zum Beispiel um Linked Data in verschiedenen Sprachen zu verbalisieren, und die Interpretation natürlicher Sprache in Bezug auf eine Ontologie, zum Beispiel um das Abfragen von Linked-Data-Quellen in unterschiedlichen Sprachen zu ermöglichen.

7.4.4 Natürlichsprachliche Generierung

Eine der Stärken von Linked Data ist, dass Daten in einem strukturierten, maschinenlesbaren Format vorliegen. Aber während auf der einen Seite Maschinen mit diesen Daten operieren, will man sie auf der anderen Seite auch für Menschen zugänglich machen. Unternehmen, die Linked Data benutzen, wollen zum Beispiel ausgehend von diesen Daten Produktbeschreibungen generieren, diese Beschreibungen in verschiedenen Sprachen lokalisieren, usw. Solche Beschreibungen sollen dabei stets im Einklang mit den Daten sein, auch wenn diese sich ändern. Dazu braucht man Methoden, um Linked Data in ein für Menschen leicht verständliches Format wie natürliche Sprache umzuwandeln. Natürlichsprachliche Generierung beschäftigt sich daher mit dem automatischen Erzeugen von Texten aus strukturierten Daten.

In den letzten Jahren ist eine Reihe von Systemen entwickelt worden, die sich dieser Aufgabe annehmen [3]. Was alle diese Systeme gemeinsam haben, ist, dass sie Wissen brauchen, wie die Elemente einer Ontologie oder Wissensbasis, d. h. die Klassen, Eigenschaften und Individuen, sprachlich ausgedrückt werden. Zum einen können dafür die vorhandenen Labels genutzt werden [20, 23], zum anderen bieten Ontologielexika eine reiche Quelle solcher Informationen.

Die meisten Systeme zur natürlichsprachlichen Generierung unterscheiden zwei Phasen: die Auswahl des zu verbalisierenden Inhalts (Was soll gesagt werden?) und die sprachliche Realisierung desselben (Wie soll es gesagt werden?). Systeme folgen dabei typischerweise einer Pipeline, die von Reiter & Dale [21] vorgeschlagen wurde, und er-

zeugen Text in den folgenden drei Etappen. Zuerst werden der Inhalt und die Struktur des Textes geplant, d. h. welche Fakten versprachlicht werden sollen und in welcher Reihenfolge bzw. in welchen Gruppen. Im nächsten Schritt wird für jedes dieser Fakten ein Satzmuster festgelegt, das dann schließlich im letzten Schritt mit spezifischen Ausdrücken gefüllt wird.

Vor allem im letzten Schritt, wenn Entscheidungen auf lexikalischer Ebene nötig sind, kann ein Ontologielexikon den Generierungsprozess maßgeblich unterstützen. Der deutlichste Fall ist, dass ein Lexikon üblicherweise eine oder mehrere Lexikalisierungsalternativen für ein Ontologieelement angibt. Diese können mit Informationen darüber verknüpft werden, in welchem Kontext welche Lexikalisierung zu bevorzugen ist. Zum Beispiel können bestimmte Lexikalisierungen als Fachbegriffe markiert werden, die dann nur verwendet werden, wenn die Zielgruppe des Textes Experten sind, nicht aber, wenn ein Text für Anfänger oder Nichtexperten erzeugt wird. Ein System, das ein Ontologielexikon zu diesem Zweck benutzt, ist in [5] beschrieben.

Ein System, das Ontologien verbalisiert, die, ähnlich zu Ontologielexika, mit RDF-Annotation von linguistischen Informationen auf lexikalischer Ebene, aber auch auf Satzebene und in Bezug auf Nutzermodellierung angereichert sind, ist *NaturalOWL* [12]. Der einzige Nachteil von diesem System ist, dass lexikalische Information und Information über Satzmuster sowie Text- bzw. Dialoginformationen nicht voneinander getrennt sind. Es ist also schwierig, schon vorhandene lexikalische Informationen für die Generierung verschiedener Textformen wiederzuverwenden. Das ist genau das Szenario, das wir im vorigen Abschnitt beschrieben haben: ein Ontologielexikon erfasst lexikalische Informationen, während domänenunabhängige Grammatikmodule Satzmuster sowie Text- und Dialogformen festlegen, die dann variabel miteinander kombiniert werden können.

7.4.5 Multilinguales Question Answering

Question Answering ist die Aufgabe, automatisch Antworten zu natürlichsprachlichen Fragen aufzufinden. Die Quellen, in denen nach Antworten gesucht wird, sind dabei unterschiedlicher Natur. Traditionell liegt ein starker Fokus auf Textdaten, also dem Auffinden von Antworten aus Zeitungsartikeln, Webseiten usw. Aber bereits in den 1960ern und 1970ern wurde begonnen auch strukturierte Daten zu berücksichtigen, insbesondere aus Datenbanken [1]. Heutzutage spielt vor allem Linked Data eine zunehmend wichtige Rolle. Mit der wachsenden Menge semantischer Daten wächst auch das Interesse an Methoden, diese Daten Endnutzern zugänglich zu machen. Eine Möglichkeit ist *Question Answering*, dessen Aufgabe nun darin besteht, ausgehend von einer natürlichsprachlichen Frage eine formale Abfrage zu konstruieren, welche die Antwort(en) aus einer gegebenen Linked-Data-Quelle extrahiert. Das erlaubt Endnutzern ihren Informationsbedarf auf eine einfache und intuitive Art und Weise auszudrücken – einerseits ohne mit Semantic-Web-Sprachen wie RDF und der dazugehörigen Abfragesprache SPARQL vertraut sein

zu müssen, und andererseits ohne das den Daten zugrundeliegende Schema kennen zu müssen.

Im Falle unseres Weinhändlers könnte eine Endnutzeranfrage zum Beispiel folgende sein: „Gib mir alle Weine, die im Breisgau angebaut werden und unter 20 EUR kosten.“ Die entsprechende SPARQL-Abfrage würde dann wie folgt aussehen:

```
SELECT DISTINCT ?w WHERE {
  ?w rdf:type onto:Wein .
  ?w onto:Erzeuger ?x .
  ?x onto:Ort onto:Breisgau .
  ?w onto:Preis ?p .
  FILTER (?p < 20)
}
```

Das Abbilden von natürlichsprachlichen Fragen auf formale Abfragen stellt einige Herausforderungen bereit. Zuerst einmal müssen natürlichsprachliche Ausdrücke auf URIs der den Daten zugrundeliegenden Ontologie abgebildet werden. In einigen Fällen ist das unkompliziert, zum Beispiel entspricht das Wort „Wein“ der Ontologieklasse `Wein` und der Name „Breisgau“ dem Individuum `Breisgau`, in anderen Fällen aber ist das schwieriger, zum Beispiel muss „angebaut“ auf die Eigenschaft `Erzeuger` und „kosten“ auf die Eigenschaft `Preis` abgebildet werden, während die Präposition „unter“ einem Filter über dem Objekt dieser Eigenschaft entspricht.

Noch einmal komplizierter wird es, wenn die Abfrage kürzer formuliert wird als „Gib mir alle Weine aus dem Breisgau für unter 20 EUR“. In diesem Fall sind die relevanten Eigenschaften der Ontologie nicht explizit benannt, sondern hinter semantisch leichten Ausdrücken wie „aus“ und „für“ versteckt. Gerade die Interpretation von solchen Präpositionen hängt sehr stark vom Kontext und von der zugrundeliegenden Domäne ab. Betrachten wir einen Satz wie „die Veröffentlichung der jährlichen Kennzahlen ist Pflicht für Unternehmen aus dem Dienstleistungssektor“, ist leicht ersichtlich, dass die Präpositionen „für“ und „aus“ hier anders gebraucht werden als in unserem Weinbeispiel und sich auf ganz andere ontologische Elemente beziehen würden.

Hinzu kommt, dass oft nicht nur die Begriffe und die entsprechenden URIs andere sind, sondern dass sich auch die Struktur der natürlichsprachlichen Frage von der der formalen Abfrage unterscheidet. In der Phrase „Weine aus dem Breisgau“, zum Beispiel, gibt es eine linguistische Relation zwischen „Weine“ und „Breisgau“, in der Ontologie hingegen sind Weine nicht direkt mit einem Ort verbunden, sondern mit einem Erzeuger, der wiederum an einem Ort verankert ist. Die Relation in der Frage entspricht also zwei Tripeln in der SPARQL-Abfrage, `?w onto:Erzeuger ?x` und `?x onto:Ort onto:Breisgau`, wohingegen für die Interpretation der Phrase „Weinbauern aus dem Breisgau“ nur letzteres Tripel wichtig wäre.

Es gibt eine Reihe von Ansätzen zu *Question Answering* über Linked Data und das Forschungsinteresse wächst stetig. Einen guten ersten Überblick bietet [17]. Es gibt sowohl Ansätze, die rein auf Ontologielexika aufbauen, zum Beispiel *Pythia* [26], als auch

Ansätze, die versuchen, Fragen unabhängig von solchen Ressourcen in Bezug auf ein beliebiges Ontologievokabular zu interpretieren, und zu diesem Zweck verschiedene Strategien anwenden, um die Lücke zwischen natürlicher Sprache und dem zugrundeliegenden Ontologieschema zu schließen. Neuere Arbeiten umfassen zum Beispiel [11], [15] und [25].

In vielen Fällen kann, unabhängig vom genauen Ansatz eines Systems, ein Ontologielexikon helfen, die Diskrepanz zwischen natürlicher Sprache und der Ontologie zu überbrücken. Eine solche Brücke wird vor allem dann unverzichtbar, wenn Daten in verschiedenen Sprachen vorliegen und abgefragt werden. Multilingualität rückt zunehmend in das Interesse der Semantic-Web-Community, da sowohl die Zahl der Daten, die in anderen Sprachen als Englisch veröffentlicht werden, als auch die Zahl der Nutzer, die auf diese Daten zugreifen wollen und nicht Englisch als Muttersprache sprechen, erheblich wächst. Das stellt eine Herausforderung für die meisten aktuellen Ansätze zu *Question Answering* über Linked Data dar, die oft für das Englische entwickelt wurden und bis auf wenige Ausnahmen nicht multilingual sind. Um diese Richtung in der Forschung zu betonen, konzentriert sich die Evaluationskampagne *Question Answering over Linked Data* [16] (QALD), Teil einer größeren *Question-Answering*-Initiative[16], u.a. auf Multilingualität und bietet einen Benchmark mit Fragen in sieben europäischen Sprachen an (Englisch, Deutsch, Spanisch, Italienisch, Französisch, Niederländisch und Rumänisch).

7.4.6 Weitere Anwendungen

Weitere Anwendung für lexikalisiertes Linked Data sind zum Beispiel folgende:

- Generierung von natürlichsprachlichen Fragen, die ein gegebener Datensatz beantworten kann, um dem Anweder das Verständnis des Inhaltes der Daten zu vereinfachen (ähnlich zu Mathieu d'Acquin et al. [9], allerdings in natürlicher Sprache)
- Informationsextraktion von Relationen aus textuellen Daten (wie zum Beispiel in [13] beschrieben)
- Validierung von RDF-Fakten anhand textueller Quellen (wie ebenfalls in [13] beschrieben)
- Generierung von Formularen für die Modellierung oder Abfrage von Daten (siehe zum Beispiel [27])
- Generierung von natürlichsprachlichen Zusammenfassungen für verlinkte Datensätze (wie zum Beispiel in [2] beschrieben)

In all diesen Anwendungen wird Wissen darüber benötigt, wie Klassen, Eigenschaften und Individuen eines Datensatzes sprachlich ausgedrückt werden. Demnach können Ontologielexika in all diesen Anwendungen von Interesse und Nutzen sein.

[16] Siehe: http://nlp.uned.es/clef-qa/, aufgerufen am 17.03.2014.

7.5 Zusammenfassung und Ausblick

In diesem Kapitel haben wir eine Übersicht über Aspekte der Multilingualität im Kontext von Linked Data gegeben. Wir haben dabei die Wichtigkeit der Anreicherung von Linked Data mit Information darüber, wie die verschiedenen Elemente der verwendeten Vokabulare in verschiedenen natürlichen Sprachen lexikalisiert werden, motiviert und Herausforderungen diskutiert, die sich aus der Nutzung von Linked Data in multilingualen Anwendungen ergeben.

Des Weiteren haben wir die wichtigen Aufgaben der Ontologie-Lokalisierung und des sprachübergreifenden Ontologie-Alignments eingeführt und einen kurzen Überblick über die Problemstellung und den Stand der Technik in diesen Bereichen gegeben.

Wir haben das *lemon*-Modell eingeführt und gezeigt, wie es dazu genutzt werden kann, um Linked-Data-Vokabulare mit linguistischen Informationen darüber anzureichern, wie die Elemente eines Vokabulars in verschiedenen Sprachen ausgedrückt werden können. Lexikalisierungen, wie sie vom *lemon*-Modell in Form von sogenannten Ontologielexika bereitgestellt werden, werden in der Zukunft als Basis für Anwendungen dienen, in denen zwischen Linked Data und einer sprachbasierten Repräsentation vermittelt werden muss. Wir haben drei solcher Anwendungen ausführlicher besprochen: das automatische Erzeugen von Grammatiken, die natürlichsprachliche Generierung von Texten aus strukturierten Daten und das multilinguale *Question Answering* über Linked Data. Außerdem haben wir einige andere Anwendungen skizziert.

Derzeitige Standardisierungsaktivitäten des W3C bauen auf dem *lemon*-Modell auf, um einen Standard für die Anreicherung von Linked-Data-Quellen mit lexikalischen Informationen zu erarbeiten. Die Vision, auf die die Mitglieder der W3C *Ontology Lexicon Community Group* hinarbeiten, ist die eines Linked-Data-Webs, in dem alle relevanten Vokabulare und Datensätze mit einem entsprechenden Lexikon verlinkt werden, in dem Daten und Ontologien über Sprachen hinweg miteinander verlinkt sind und so den sprachübergreifenden Zugriff auf das Datennetzwerk unterstützen. Dabei wird eine wichtige Frage sein, wie die Kosten für die Erzeugung von Lexika reduziert werden können, z. B. durch induktive Verfahren, die aus Daten lernen (siehe z. B. [28]), und durch Crowdsourcing-Verfahren oder kollaborative Arbeitsteilung.

Literatur

1. Androutsopoulos, Ion, Graeme D. Ritchie, und Peter Thanisch. 1995. Natural language interfaces to databases – an introduction. *Journal of Natural Language Engineering* 1(1): 29–81
2. Bontcheva, Kalina. 2005. Generating tailored textual summaries from ontologies. In *The Semantic Web: Research and Applications* Lecture Notes in Computer Science, Bd. 3532, Hrsg. Asunción Gómez-Pérez, Jérôme Euzenat, 531–545. Springer
3. Bouayad-Agha, Nadjet, Gerard Casamayor, und Leo Wanner. Natural language generation in the context of the semantic web. *Semantic Web – Interoperability, Usability, Application*, to appear

4. Cimiano, Philipp, Paul Buitelaar, John McCrae, und Michael Sintek. 2011. LexInfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1), 29–51
5. Cimiano, Philipp, Janna Lüker, David Nagel, und Christina Unger. 2013. Exploiting ontology lexica for generating natural language texts from rdf data. In *Proceedings of the 14th European Workshop on Natural Language Generation (ENLG 2013)*
6. Cimiano, Philipp, Elena Montiel-Ponsoda, Paul Buitelaar, Mauricio Espinoza Mejía, und Asunción Gómez-Pérez. 2010. A Note on Ontology Localization. *Journal of Applied Ontology* 5(2), 127–137
7. Cimiano, Philipp, Christina Unger, und John McCrae. 2014. *Ontology-based interpretation of natural language*. Morgan & Claypool
8. Monnet (FP-ICT-4-248458) Consortium. 2011. *D1.1.2 Final Use Case Definition and Scenario Development*. http://www.monnet-project.eu/Monnet/Monnet/English/Navigation/D2_2
9. d'Aquin, Mathieu, und Enrico Motta. 2011. Extracting relevant questions to an rdf dataset using formal concept analysis. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP)*, 121–128. ACM
10. Euzenat, Jérôme, und Pavel Shvaiko. 2013. *Ontology matching*, 2. Aufl.: Springer
11. Freitas, André, Edward Curry, Joao Gabriel. Oliveira, und Seán O'Riain. 2011. A distributional structured semantic space for querying RDF graph data. *International Journal of Semantic Computing* 5(4): 433–462
12. Galanis, Dimitrios, und Ion Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *Proc. 11th European Workshop on Natural Language Generation (ENLG '07)*, 143–146
13. Gerber, Daniel, und Axel-Cyrille Ngonga Ngomo. 2011. From RDF to Natural Language and Back. In *Towards the Multilingual Semantic Web: Principles, Methods and Applications*, Hrsg. Paul Buitelaar, Philipp Cimiano: Springer. to appear.
14. Gómez-Pérez, Asunción, Daniel Vila-Suero, Elena Montiel-Ponsoda, Gracia Jorge, und Guadalupe Aguado de Cea. 2013. Guidelines for multilingual linked data. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics (WIMS'13)*, 3:1–3:12. ACM
15. Kwiatkowski, Tom, Eunsol Choi, Yoav Artzi, und Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*
16. Lopez, Vanessa, Christina Unger, Philipp Cimiano, und Enrico Motta. 2013. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web* 21(0): 3–13
17. Lopez, Vanessa, Victoria Uren, Marta Sabou, und Enrico Motta. 2011. Is question answering fit for the semantic web? A survey. *Semantic Web Journal* 2: 125–155
18. McCrae, John, Dennis Spohr, und Philipp Cimiano. 2011. Linking lexical resources and ontologies on the semantic web with lemon. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications – Volume Part I* ESWC'11., 245–259. Springer
19. Espinoza Mejía, Mauricio, Elena Montiel-Ponsoda, und Asunción Gómez-Pérez. 2009. Ontology localization. In *Proceedings of the 5th International Conference on Knowledge Capture (KCAP09)*, 33–40

20. Mellish, Chris, und Xiantang Sun. 2006. The Semantic Web as a linguistic resource: Opportunities for natural language generation. *Knowl.-Based Syst.* 19(5): 298–303

21. Reiter, Ehud, und Robert Dale. 2000. *Building natural language generation systems*. Cambridge University Press

22. Spohr, Dennis, Laura Hollink, und Philipp Cimiano. 2011. A machine learning approach to multilingual and cross-lingual ontology matching. In *Proceedings of 10th International Semantic Web Conference*, 665–680

23. Sun, Xiantang, und Chris Mellish. 2007. An experiment on "free generation" from single RDF triples. In *Proc. 11th European Workshop on Natural Language Generation (ENLG '07)*, 105–108

24. Trojahn, Cáassia, Bo. Fu, Ondrej Zamazal, und Dominique Ritze. 2011. State-of-the-art in Multilingual and Cross-Lingual Ontology Matching. In *Towards the Multilingual Semantic Web: Principles, Methods and Applications*, Hrsg. Paul Buitelaar, Philipp Cimiano. Springer

25. Unger, Christina, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, und Philipp Cimiano. 2012. Template-based question answering over rdf data. In *Proceedings of the 21st World Wide Web Conference (WWW 2012)*, 639–648

26. Unger, Christina, und Philipp Cimiano. 2011. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems (NLDB 2011)*, 153–160. Springer

27. Van Grondelle, Jeroen, und Menno Gülpers. 2011. Specifying Flexible Business Processes using Pre and Post Conditions. In *Practice of Enterprise Modeling* LNBIP, Bd. 92, Hrsg. Paul Johannesson, John Krogstie, Andreas L. Opdahl, 1–14. Springer

28. Walter, Sebastian, Christina Unger, und Philipp Cimiano. 2013. A corpus-based approach for the induction of ontology lexica. In *Proc. of the 18th International Conference on Natural Language Processing and Information Systems (NLDB)* Lecture Notes in Computer Science, Bd. 7934, 102–113

Linked Data Warehousing

8

Sebastian Bayerl und Michael Granitzer

Zusammenfassung

Data-Warehousing bezeichnet die technologische Realisierung analytischer Datenbestände sowie entsprechender Schnittstellen zu deren Exploration und Analyse. Linked Data bietet vor allem mit der vor Kurzem begonnenen Entwicklung des RDF Data Cube Vokabulars neue Entwicklungsmöglichkeiten für Data-Warehousing Technologien und deren Einsatzspektrum. Der Beitrag stellt die Grundlagen zu Data-Warehouses vor und führt in das RDF Data Cube Vokabular als Linked Data Äquivalent ein. Beide Grundlagen dienen der Diskussion sowohl der Anwendung von RDF Data Cubes im Data-Warehousing als auch der Erweiterung traditioneller Data-Warehousing Ansätze, z. B. durch Integration offener Daten in Data-Warehousing Prozessen.

8.1 Einleitung

Daten- und Trendanalysen sind beliebte Werkzeuge zum Fällen fundierter strategischer Entscheidungen in Unternehmen. Grundlage hierfür sind unternehmensrelevante Kennzahlen wie z. B. Kundendaten oder die Produktverkaufszahlen der letzten Jahre. Bei einem Data-Warehouse System (DWS) handelt es sich um eine Sammlung von Technologien und technisch abgebildeten Prozessen, die derartige Analysen ermöglichen und damit wirtschaftliche Entscheidungsprozesse unterstützen [1, 2, 4]. Die Komplexität und der Funktionsumfang eines DWS gehen dabei weit über die Möglichkeiten einer Analyse mit gängigen Tabellenkalkulationen hinaus und bilden damit eine Kerntechnologie für datenbasierte Entscheidungsprozesse in einem Unternehmen. Doch inwieweit profitieren Data-Warehouse Systeme von semantischen Technologien im Allgemeinen und

S. Bayerl ✉ · M. Granitzer
MiCS – Media Computer Science, Universität Passau, Innstraße 41, 94032 Passau, Deutschland
e-mail: bayerl@dimis.fim.uni-passau.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_8

Linked Data im Speziellen? Um diesen Mehrwert aufzuzeigen, werden im Rahmen dieses Beitrags Szenarien vorgestellt, wie Linked Data im Bereich des Data-Warehousing nutzbringend eingesetzt werden kann. Dazu wird gezeigt, wie Linked Data Warehousing die Integration von RDF-basierten Daten aus eigenen Data-Marts, Data-Marketplaces oder der Linked Open Data Cloud ermöglicht. Die semantische Repräsentation des Datenbestandes liefert dazu (i) neue Automatisierungsmöglichkeiten zur Datenintegration verteilter analytischer Datenbestände, (ii) semi-automatische Integrationsmöglichkeiten existierender offener Datenbestände und ontologischer Wissensbasen im WWW sowie (iii) Möglichkeiten zur Generierung von Aggregationshierarchien zur Unterstützung klassischer Interaktionsmöglichkeiten mit analytischen Datenbeständen.

Mit dem Ziel diese Möglichkeiten aufzuzeigen, befasst sich dieses Kapitel mit den folgenden Themen:

- Vermittlung der Grundlagen des Data-Warehousing
- Veranschaulichung des multidimensionalen Datenmodells und dessen Umsetzung im Kontext von Linked Data
- Darstellung von Szenarien, wie Linked Data im Bereich des Data-Warehousing nutzbringend eingesetzt werden kann.

Zum Aufbau: Zunächst gibt Abschn. 8.2 einen Überblick über die Grundlagen des Data-Warehousing. In Folge stellt Abschn. 8.3 dessen multidimensionales Datenmodell vor. Aufbauend auf dem RDF Data Cube Vokabular, welches in Abschn. 8.4 beschrieben wird, werden in Abschn. 8.5 Szenarien für Linked Data-Warehousing aufgezeigt. Im Laufe dieses Kapitels wird ein fortlaufendes Bespiel verwendet, um die einzelnen Konzepte zu veranschaulichen. Es handelt sich dabei um den Verkauf von Produkten. Diese Verkäufe werden mit Produktinformationen, Verkaufsort und Verkaufszeitpunkt unter verschiedenen Gesichtspunkten in Zusammenhang gebracht.

8.2 Data-Warehousing im Überblick

Die Aufgabe klassischer Datenbanksysteme im produktiven Einsatz ist meist das Speichern und Laden einzelner Datensätze. Das Anlegen eines neuen Kunden oder die Verwaltung eines Produktkataloges sind Beispiele hierfür. Die Herausforderung besteht darin, eine große Anzahl kleiner Zugriffe effizient zu verarbeiten, um einen hohen Durchsatz bieten zu können.

Ähnlich einer klassischen Datenbank, handelt es sich bei einem DWS generell um eine Datenbank, die genutzt wird, um Daten zu speichern und Anfragen zu beantworten. Die Anforderungen an ein Data-Warehouse unterscheiden sich jedoch stark von denen an ein klassisches Datenbanksystem. Grund dafür ist die Art der Anfragen, die an sie gestellt werden. Priorität hat nicht mehr die schnelle Abarbeitung vieler Anfragen, sondern die Analyse tendenziell riesiger Datenmengen innerhalb weniger Anfragen. Analyse

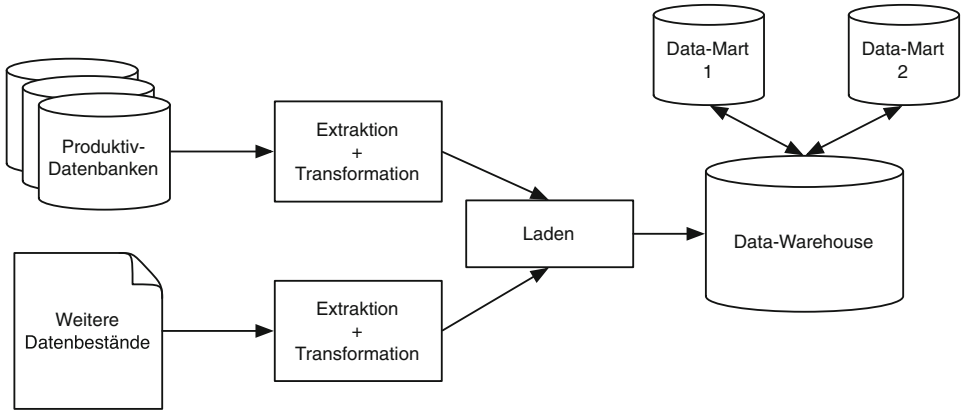


Abb. 8.1 Der ETL-Prozess

bedeutet hier vor allem die komplexe Aggregation numerischer Werte, welche zuvor sortiert und selektiert wurden. Dieser Vorgang wird auch als Online Analytical Processing (OLAP) [5] bezeichnet. Als Datengrundlage dienen verschiedenste Datenquellen aus den unterschiedlichen Unternehmensbereichen, welche miteinander kombiniert werden, um eine möglichst vollständige und integrierte Basis für Entscheidungsprozesse zu schaffen. Diese Integration trägt bereits entscheidend zur Komplexität und vor allem auch zu den Kosten eines Data-Warehouse Systems bei. Die Datenintegration lässt sich in drei Prozessschritte – (1) die Extraktion, (2) die Transformation und (3) das Laden (ETL-Prozess) [3] – untergliedern, wie in Abb. 8.1 veranschaulicht.

1. **Extraktion**: Relevante Daten liegen meist in unterschiedlichen Quellen vor und müssen aus diesen extrahiert werden. Dabei kann es sich um Excel- oder CSV-Dokumente, um klassische relationale Datenbanken oder um NoSQL- und Graph-Datenbanken handeln. Die Heterogenität beschränkt sich dabei nicht nur auf die unterschiedlichen Datenformate, die Syntax, sondern auch auf die Bedeutung und Strukturierung der Daten, die Semantik. Aufgabe dieses Prozessschrittes ist nun die vollständige und korrekte Extraktion der relevanten Daten, ohne dabei den Produktivbetrieb des Ursprungssystems zu stören. Diese Extraktion kann periodisch (z. B. wöchentlich), ereignisgesteuert (z. B. nach einer Anzahl neuer Datensätze) oder anfragegesteuert (z. B. durch Anfrage des Data-Warehouse-Systems) stattfinden.
2. **Transformation**: Die Transformation bezeichnet die Überführung der unterschiedlichen Quellformate in das Schema des Data-Warehouse. Die Aufgaben reichen hier von trivialen Typ-Konvertierungen bis hin zu komplexen regel-basierten Integritätschecks und der Bereinigung der Daten von Fehlern (dem Data-Cleaning). Zusammengehörige Datensätze aus den verschiedenen Quellen müssen identifiziert und basierend auf entsprechenden Transformationsregeln zusammengeführt werden. Zur Datenbereinigung

gehört auch die Erkennung und Beseitigung von Duplikaten; ein oft unterschätztes, nicht-triviales Problem in großen Datenbeständen. Allgemeine Datenqualitätsprobleme reichen von veralteten oder nur partiell aktuellen Daten bis zu schlechthin falschen Daten. So können Kundendaten unterschiedlich hinterlegt sein. Wird z. B. der Name eines Kunden zur Identifikation verwendet und in den einzelnen Quellen unterschiedlich gespeichert, so ist der rein syntaktische Vergleich erschwert und potentiell mit Mehrdeutigkeiten belegt. Hier zeichnet sich bereits der Einsatz semantischer Technologien zur Integration mehrdeutiger Daten ab.

3. **Laden**: Der letzte Schritt der Datenintegration besteht darin, die extrahierten und transformierten Daten in das Data-Warehouse zu laden, welches schon Daten enthalten könnte. Dies kann ein einfaches Hinzufügen oder ein Aktualisieren bestehender Datensätze nach sich ziehen, abhängig von den Eigenschaften der transformierten Daten.

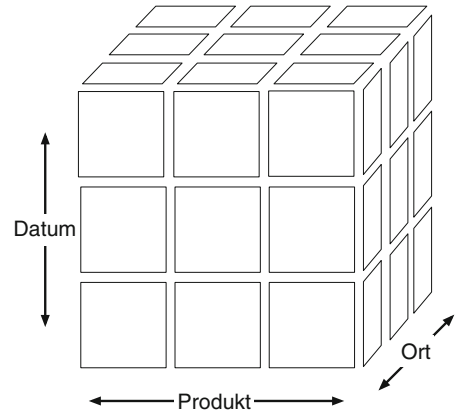
Da sehr große Datenmengen diesen Prozess durchlaufen sollen, ist der Einsatz effizienter Algorithmen in jedem dieser Schritte notwendig. Ein weiterer Ansatzpunkt der Integration sind verteilte Datenbanken für Data-Warehouse Systeme, welche Daten nur virtuell integrieren und somit die Anforderungen an die Datenhaltung reduzieren.

Des Weiteren lassen sich Data-Warehouses oftmals in Data-Marts untergliedern. In sehr großen Unternehmen kann es zu komplex oder einfach nicht zielführend sein ein Data-Warehouse bereitzustellen, in welches alle Daten integriert werden. Data-Marts bieten eine Lösung, indem sie nur Teile des kompletten Datenbestandes integrieren und sich auf einzelne Abteilungen oder Applikationen spezialisieren. Wiederum ergibt sich jedoch das Problem einer verteilten Datenhaltung und der Notwendigkeit der Datenintegration für Data-Mart übergreifende Abfragen. Hier zeichnen sich Einsatzmöglichkeiten für Linked Data zur Verbesserung der Datenintegration ab. Bevor jedoch darauf eingegangen werden kann und mit dem RDF Data Cube Vokabular ein entsprechendes Beschreibungsformat vorgestellt wird, muss das einem Data Warehouse zugrunde liegende multidimensionale Datenmodell eingeführt und erklärt werden.

8.3 Das multidimensionale Datenmodell

Klassische Datenbanken verwenden in der Regel normalisierte Schemata, um eine Redundanz der Daten zu vermeiden. Einzelne Entitäten wie z. B. Kunden oder Produkte werden jeweils in getrennten Tabellen gespeichert. Ein Data-Warehouse verwendet alternativ dazu ein multidimensionales Datenmodell, das aus Fakten- und Dimensionstabellen besteht. Erst durch dieses geänderte Schema ist es möglich, Anfragen effizient zu bearbeiten. Dies macht eine Transformation der Daten notwendig, bevor sie importiert werden können. Die multidimensionale Struktur wird auch OLAP-Würfel genannt und ist die grundlegende Datenstruktur im Data-Warehousing. Diese Würfelstruktur soll hier zusammen mit den darauf aufbauenden Interaktionsmöglichkeiten – den OLAP-Operatoren – kurz vorgestellt werden.

Abb. 8.2 OLAP-Würfel



8.3.1 OLAP Würfel

Die Datenstruktur eines Data-Warehouse ist darauf ausgelegt, numerische Werte, sogenannte Fakten, zu speichern. Bei diesen Fakten kann es sich z. B. um Verkaufszahlen oder andere unternehmerische Kennzahlen handeln. Zur eindeutigen Identifikation sind sie von einer Menge von Dimensionen abhängig. Um Verkaufszahlen zuzuordnen, könnten das Verkaufsdatum, das Produkt und der Verkaufsort als passende Dimensionen gewählt werden. Dieses Konzept lässt sich mit einem dreidimensionalen Koordinatensystem veranschaulichen, wobei, wie in Abb. 8.2 ersichtlich, an jeder Achse eine einzelne Dimension aufgetragen wird. Jedes Faktum kann nun entsprechend der Ausprägung seiner Dimensionen an einer eindeutigen Koordinate eingetragen werden. Der Begriff OLAP-Würfel lässt sich von dieser dreidimensionalen Visualisierung ableiten. Die tatsächliche Datenstruktur kann jedoch beliebig viele Dimensionen besitzen und wird daher als Hyperwürfel bezeichnet.

8.3.2 OLAP-Operationen

Mit dem OLAP-Würfel ist es möglich, Data-Warehouse-spezifische Anfragen und Operationen auf analytischen Datenbeständen zu definieren. Diese sogenannten OLAP-Operationen basieren auf der Aggregation der numerischen Werte in der Faktentabelle. Aggregation kann hierbei die Berechnung der Summe oder eines Mittels mehrerer Werte bedeuten. Diese Funktionalität ermöglicht z. B. die Berechnung der Verkaufszahlen für ein bestimmtes Produkt in einem festgelegten Land.

Verschiedene OLAP-Operationen stehen zur Verfügung, um eine derartige Auswahl in einem OLAP-Würfel treffen zu können. Die Operation *Roll-Up* beispielsweise basiert auf der Gruppierung der Werte einer oder mehrerer Dimensionen. Die Dimension Verkaufsdatum könnte nach Monaten gruppiert werden, was zu einer Aufsummierung aller

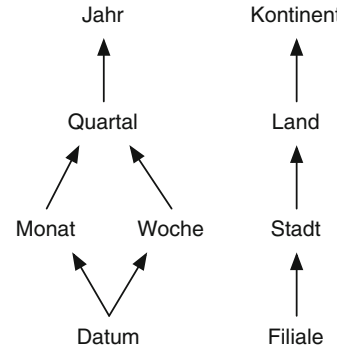


Abb. 8.3 Aggregationshierarchien

Verkaufszahlen für jeden Monat führen würde. Die Hierarchien der Dimensionen wie Woche, Monat und Jahr oder Stadt, Land und Kontinent dienen oftmals als Grundlage für die Gruppierungen und somit für die Aggregationen. Abbildung 8.3 zeigt beispielhaft derartige Hierarchien. Der *Roll-Up* entspricht also einem Schritt auf einer Hierarchieebene mit niedrigerer Granularität. Der *Drill-Down* bezeichnet die entgegengesetzte Operation zu einem *Roll-Up*. Die Gruppierung wird hierbei so verändert, dass eine Dimension z. B. nach Wochen und nicht mehr nach Monaten gruppiert wird. Die feinste Granularität wird hierbei natürlich von der Aggregationsstufe der importierten Fakten vorgegeben.

Will man nur die Anzahl der Verkäufe eines bestimmten Jahres berechnen, so muss diese Anfrage verfeinert werden. Mit der *Slicing*-Operation ist es möglich, einzelne Ausprägungen von Dimensionswerten zu selektieren. Damit lassen sich beispielhaft alle Verkäufe aus einem gewissen Jahr selektieren, wobei die Daten der anderen Jahre nicht mit in die Berechnung eingehen.

Neben *Roll-Up*, *Drill-Down* und *Slicing* gibt es noch weitere elementare Operationen wie *Pivoting* und *Dicing* oder komplexe Operationen wie die *Cube* Funktion [11], auf die hier jedoch nicht eingegangen werden soll.

8.3.3 Star- und Snowflake-Schema

Zur Abbildung des multidimensionalen Datenmodells eines OLAP-Würfels werden in einem auf relationalen Datenbanken basierenden Data-Warehouse System de-normalisierte Schemata, wie das Star- oder das Snowflake-Schema verwendet. Die Speicherung der numerischen Werte erfolgt in der Faktentabelle; die einzelnen Dimensionsausprägungen in den sogenannten Dimensionstabellen. Ein Star-Schema, wie in Abb. 8.4 zu sehen, besteht aus einer zentralen Faktentabelle Verkauf, die direkt mit mehreren Dimensionstabellen in Beziehung steht. Eine solche Dimensionstabelle kann das Produkt zusammen mit der Produktkategorie sein. Wegen dieser funktionalen Abhängigkeiten liegt das Star-Schema nur in der zweiten Normalform vor. Um diese Abhängigkeiten aufzulösen, kann das Schema in die dritte Normalform überführt werden. Dazu werden die Dimensionstabellen in

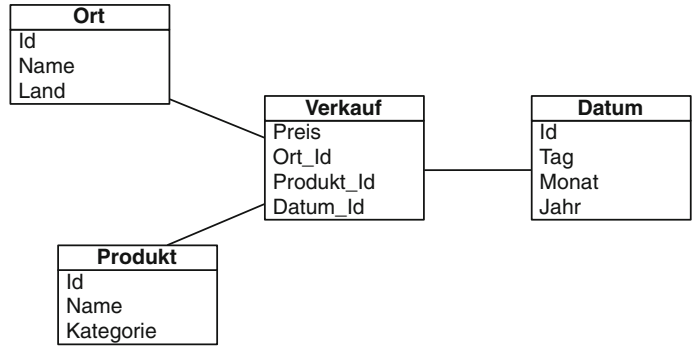


Abb. 8.4 Star-Schema

mehrere Tabellen aufgeteilt und sogenannte Satellitentabellen gebildet. Bei dem daraus resultierenden Schema handelt es sich um ein Snowflake-Schema. Abbildung 8.5 zeigt, wie die Abhängigkeiten durch zusätzliche Tabellen abgelöst werden können.

Der Unterschied zwischen dem Star- und dem Snowflake-Schema liegt vor allem in der Datenredundanz und Abfragegeschwindigkeit. So speichert das Star-Schema Daten aus den Dimensionstabellen redundant und hat somit einen höheren Speicherbedarf. Während im Snowflake-Schema lediglich der Schlüssel zum Eintrag in der Produktkategorietabelle abgelegt werden muss, muss beim Star-Schema zu jedem Produkt die Produktkategorie ebenfalls gespeichert werden. Da in Data-Warehouse Systemen jedoch die Hauptlast an Daten in der zentralen Faktentabelle vorliegt, ergibt sich nur ein geringer Unterschied im Datenvolumen. Der Vorteil des Star-Schemas hingegen liegt in der rascheren Anfra-

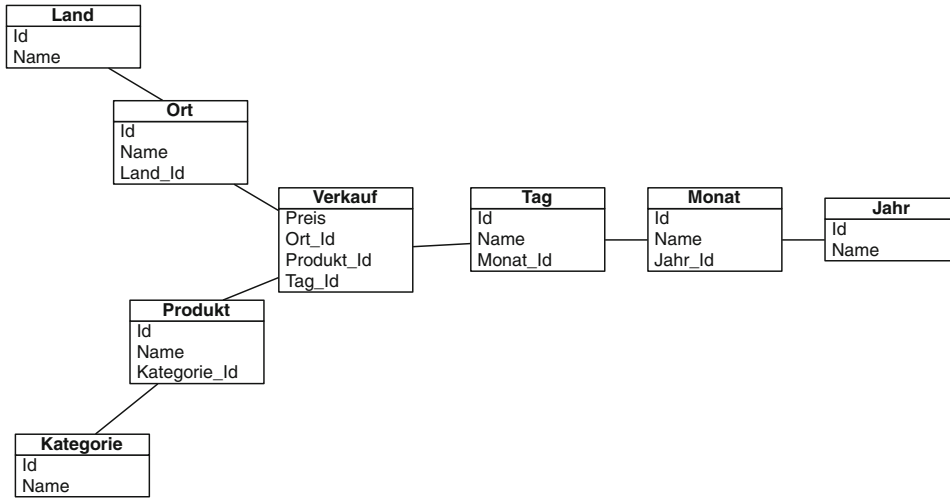


Abb. 8.5 Snowflake-Schema

gebeantwortung, da keine Verknüpfungen zwischen verschiedenen Dimensionstabellen notwendig sind.

Star- und Snowflake-Schema bilden die Grundlage für die relationale Repräsentation analytischer Datenbestände mit allen Vor- und Nachteilen. Zusammenhänge zwischen Fakten und Dimensionen werden rein mengenmäßig, also quantitativ betrachtet. Die Bedeutung von Fakten und die damit verbundenen erlaubten oder verbotenen Operationen können nicht direkt in der Datendefinition spezifiziert werden. Durch die fehlende Beschreibung der Bedeutung einzelner Elemente obliegt die Interpretation der Daten wiederum dem Data-Warehouse System. Bei der Verwendung mehrerer Data-Marts erhöht sich der Aufwand der Datenwartung und Integration. Das im Folgenden beschriebene RDF Data Cube Vokabular kombiniert nun das multidimensionale Datenmodell mit semantischen Datenbeschreibungssprachen und realisiert somit ein semantikerhaltendes, multidimensionales Datenmodell.

8.4 Das RDF Data Cube Vokabular

Bei dem *RDF Data Cube Vocabulary* handelt es sich um die Datenstruktur eines OLAP-Würfels im Resource Description Framework (RDF) Format. Eine vollständige Dokumentation findet sich in [8], wobei im Folgenden nur eine Auswahl an grundlegenden Elementen vorgestellt wird.

Das *RDF Data Cube Vocabulary* befindet sich im Prozess der Standardisierung durch das W3C und ist als *W3C Candidate Recommendation* verfügbar. Ziel ist es, die Publikation mehrdimensionaler Daten als Linked Data, wie sie z. B. beim Data-Warehousing generiert werden, zu vereinfachen. Unter Verwendung des *W3C Resource Description Framework* (RDF) [7] definiert das Vokabular ein Schema aus Klassen und Eigenschaften, welches die Struktur der zu publizierenden Daten vorgibt und die Verwendung semantischer Beschreibungssprachen ermöglicht.

Ähnlich dem Schema eines Data-Warehouse muss für einen RDF-Würfel die Datensatz-Struktur-Definition (*DatasetStructureDefiniton*) angegeben werden. Diese definiert, aus welchen Komponenten der Datensatz zusammengesetzt ist. Zur Verfügung stehen die Komponenten *Measure*, *Dimension* und *Attribute*.

- *Measure*: Diese Komponente definiert, wie die numerischen Werte semantisch zu interpretieren sind (z. B. Anzahl der verkauften Produkte).
- *Dimension*: Äquivalent zum klassischen OLAP-Würfel werden meist mehrere Dimensionen definiert, um die numerischen Werte eindeutig identifizieren zu können (z. B. Verkaufsdatum, Verkaufsort).
- *Attribute*: Ähnlich dem *Measure* kann diese Komponente verwendet werden, um den numerischen Wert näher zu beschreiben. Hierbei ist zu beachten, dass es um Metadaten auf syntaktischer Ebene handelt (z. B. die Maßeinheit).

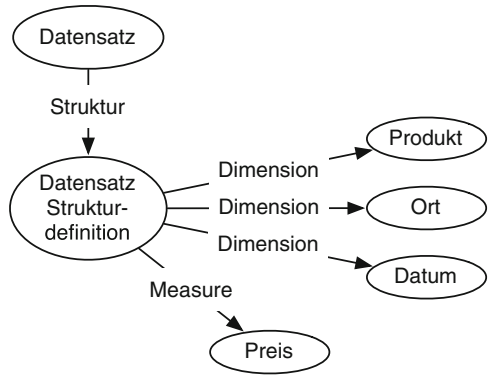


Abb. 8.6 DatasetStructureDefinition

Abbildung 8.6 zeigt die entsprechende Strukturdefinition für das Schema des Produktverkauf-Beispiels mit der Definition aller Komponenten.

Neben der Strukturdefinition werden auch die tatsächlichen Daten im RDF Format gespeichert. Ein Faktum wird dabei als eine sogenannte Beobachtung (*Observation*) gespeichert. Hierzu werden jeder *Observation* alle Komponenten zusammen mit deren Ausprägungen zugeordnet. Bei der Ausprägung der *Measure*-Komponente handelt es sich um den zu speichernden numerischen Wert. Abbildung 8.7 verdeutlicht dies für eine einzelne *Observation*.

Schließlich können für den gesamten Datensatz auch Metadaten hinterlegt werden. Informationen zu Autor, eine Beschreibung oder der Generierungszeitpunkt bieten sich hier an.

Das *RDF Data Cube Vocabulary* bietet noch weitere Möglichkeiten, die hier jedoch nicht näher beschrieben werden. Beispielsweise können *Observations* in *Slices* gruppiert werden oder die Ausprägungen einer Dimension durch eine *Code-List* eingeschränkt werden.

Data-Warehouse Systeme verwenden klassischerweise Anfragesprachen wie SQL [17] oder MDX [18], in denen OLAP-Operatoren verfügbar sind. Da RDF nicht mehr rela-

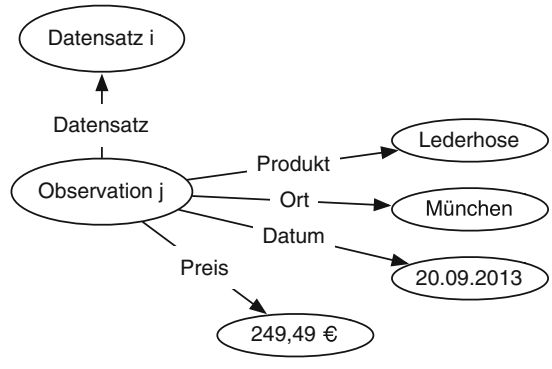


Abb. 8.7 Observation

tional, sondern als Graph vorliegt, wird auch eine eigene Anfragesprache benötigt. Zu diesem Zweck wurde SPARQL [10, 15] entwickelt. Mit Version 1.1 sind nun auch Operatoren verfügbar, die grundlegend für Linked Data Warehousing sind. Dazu gehören die *Group By* Klausel und Aggregationsfunktionen. Komplexere Klauseln haben noch keinen Einzug in die Abfragesprache gefunden, es gibt jedoch bereits den Ansatz MDX Anfragen auf SPARQL abzubilden [6]. Alternativ ist es auch vorstellbar, SPARQL in Spezialfällen nur als Datenlieferant zu verwenden, um die Algorithmen, wie z. B. eine lineare Regression, getrennt davon zu implementieren [14].

8.5 Szenarien für Linked Data Warehousing

Um den Einsatz von Linked Data im Bereich des Data-Warehousing zu motivieren, wird zunächst ein mögliches Szenario beschrieben. Von diesem können anschließend verschiedene Anwendungsmöglichkeiten abgeleitet werden, die einen Mehrwert gegenüber traditionellen Data-Warehouses darstellen.

Bei größeren Unternehmen kann man meist davon ausgehen, dass jede Abteilung ihre anfallenden Daten selbst verwaltet. Diese Daten sind potentiell heterogen und müssen, wie bereits in Abschn. 8.2 dargestellt, einen mehrstufigen Prozess durchlaufen, bis sie in ein Data-Warehouse oder in einen Data-Mart integriert werden können. Es gibt also die Notwendigkeit einer unternehmensinternen Daten-Integration – ein zentraler Punkt, der durch RDF im Allgemeinen und dem RDF Data Cube Vokabular im Speziellen erleichtert wird.

Durch die Modellierung und Bereitstellung multidimensionaler Datenbestände mittels des *RDF Data Cube Vocabulary* reduziert sich der Aufwand der Datenintegration drastisch. Zudem sind auch externe in RDF modellierte Datenquellen wie z. B. Daten aus der Linked Open Data Cloud als Teil des Data-Warehousing Prozesses einfach nutzbar. Diese offenen Daten könnten eigene Daten anreichern und erweitern und damit eine verbesserte Entscheidungsgrundlage liefern. Hierbei kann es sich z. B. um statistische Daten handeln, die im Rahmen einer Open Data Initiative von einem Bundesamt veröffentlich wurden. Abbildung 8.8 fasst dieses Szenario nochmals schematisch zusammen.

Somit bietet das RDF Vokabular unterschiedliche Mehrwerte. Die Verwendung eines standardisierten gemeinsamen Formats ermöglicht die Integration interner und externer Daten und vereinfacht die Erweiterbarkeit des eigenen Datenbestandes. Zusätzlich kann bestehendes Wissen leichter wiederverwendet werden. Dazu gehören z. B. vordefinierte Aggregationshierarchien oder funktionale Abhängigkeiten der Daten. Diese und andere Möglichkeiten sollen in den folgenden Abschnitten näher beleuchtet werden.

8.5.1 Erweiterung von Daten-Würfeln

Wie bereits angeschnitten lassen sich interne und externe Datensätze in den eigenen Datenbestand integrieren. Hierbei sind zwei grundlegende Fälle zu unterscheiden.

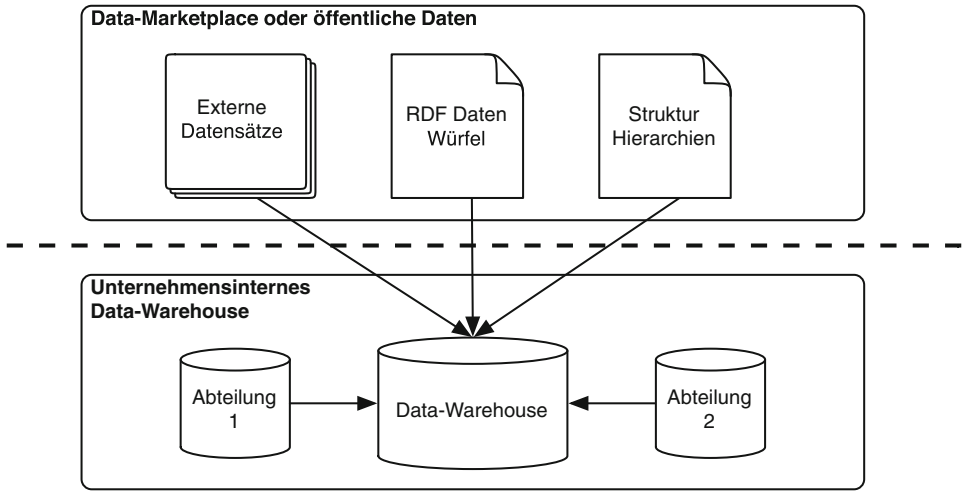


Abb. 8.8 Linked Data Warehouse Szenario

Der einfachere Fall liegt vor, wenn der bestehende Datensatz und die zu importierenden Daten bereits im selben Schema vorliegen. Die *Dimension*- und *Measure*-Definitionen müssen also identisch sein. Die Integration der neuen *Observations* ist somit dem Hinzufügen neuer Zeilen in einem klassischen Data-Warehouse gleichzusetzen. Zur Beschreibung zweier identischer *Dimensionen* und *Measures* stehen nun jedoch alle Sprachelemente semantischer Beschreibungssprachen wie z. B. *owl:sameAs* zur Verfügung und erleichtern somit die Feststellung der Gleichheit von Würfelelementen.

Die Integration wird anspruchsvoller, wenn die Schemata nicht mehr identisch oder sogar vollständig disjunkt sind. Um hier eine sinnvolle Integration zu ermöglichen, müssen Schemaanpassungen vorgenommen werden. Dazu gehören Umbenennungen oder das Hinzufügen neuer Dimensionen. Hierbei ist besonders auf die Erhaltung der Primärschlüsseleigenschaft der Menge aller resultierenden Dimensionen zu achten. Ausgehend von dem Produktverkaufsbeispiel lassen sich einfache Beispiele für solche Integrationen finden. Der Datensatz könnte z. B. eine Dimension für das Land enthalten, in dem der Verkauf getätigt wurde. Beim Import könnte diese Dimension nun um länderspezifische Daten wie die Anzahl der Bundesländer oder die Einwohnerzahl erweitert werden. Der Import bei nicht identischen Schemata kann sich jedoch noch komplexer gestalten. Auch die Erweiterung mit Daten, deren Schemata komplett disjunkt sind, hat seine Anwendungsfälle. Data Discovery und das iterative Hinzufügen einzelner *Dimensions* oder *Measures* fällt darunter. Beispielsweise könnte es nützlich sein, neben dem Verkaufsdatum auch den entsprechenden Wochentag zu speichern. Durch diese Erweiterung entscheidungsrelevanter Daten lassen sich neue Korrelationen identifizieren (siehe auch [20]).

8.5.2 Integration von RDF Daten-Würfeln

Im vorhergehenden Abschnitt wurde allgemein beschrieben, wie RDF Datensätze erweitert werden können. Zur Schemaintegration in klassische Datenbanken gibt es bereits umfangreiche Literatur [12, 13], deren Techniken zum Teil auch wiederverwendet werden können. Hier soll nun ein etwas technischerer Überblick gegeben werden, welche Schritte unter Verwendung des *RDF Data Cube Vocabulary* dazu notwendig sind.

Die Integration verläuft ähnlich zu dem bereits vorgestellten ETL-Prozess, wobei die Extraktionsphase dank dem gemeinsamen Datenformat nicht immer vollständig durchgeführt werden muss.

Der Transformationsschritt besteht aus der Generierung der finalen Struktur-Definition und der Anpassung der *Observations* auf dieses Schema. Dieser Prozess lässt sich nochmals wie folgt untergliedern. Zu Beginn sind eine Auswahl der relevanten Dimensionen und deren Identifizierung in beiden Datensätzen grundlegend, da sie den Primärschlüssel für die *Measure*-Komponenten darstellen. Dabei ist darauf zu achten, dass dieselben Werte für die Dimensionsausprägungen in beiden Datensätzen verwendet werden, damit spätere Gruppierungsoperationen ein korrektes Ergebnis liefern können. Identische Datumsformate und RDF Datentypen sind hier beispielhaft zu nennen. Zweitens kann dasselbe Verfahren auf die *Measure*-Komponenten angewendet werden, wobei diese die Primärschlüsseleigenschaft nicht erfüllen müssen. Schließlich müssen noch die einzelnen *Observations* generiert werden. Hierbei muss das eventuell veränderte Schema beachtet und auf eine eindeutige Identifizierung geachtet werden. Dieser letzte Schritt überdeckt sich bereits mit der Laden-Phase des ETL-Prozesses.

Diese Datenintegration kann unter gewissen Voraussetzungen (teil-)automatisiert durchgeführt werden. Bei identischem Schema und dem Wissen, dass es sich um gleichartige Daten handelt, kann die Integration auf einer rein syntaktischen Ebene vollständig automatisch gelöst werden. Erst die Zuordnung unterschiedlicher Komponenten erfordert eine menschliche Interaktion, wenngleich auch im geringen Umfang. Werden z. B. semantische Konzepte in unterschiedlichen Sprachen verwendet, so ist der einfache syntaktische Vergleich nicht mehr ausreichend. Für den Menschen erschließt sich sofort, dass sich die Dimensionen „Jahr“ und „Year“ wohl kombinieren lassen. Dies lässt sich eventuell auch auf syntaktischer Ebene ableiten, wenn die entsprechenden Gleichheitsinformationen ebenfalls vorliegen. Unter Einbeziehung von Linked Data kann der Grad der Automatisierung hier signifikant erhöht werden. Zum einen über die Verwendung von *owl:sameAs* Beziehungen zur Spezifikation der semantischen Gleichheit von syntaktisch unterschiedlichen Dimensionen und Measures. Zum anderen durch die Standardisierung von Maßeinheiten, die eine automatisierte Umrechnung von Kennzahlen ermöglichen. Auch die oben angeführte Mehrsprachigkeit lässt sich zum Teil automatisieren. Über Linked Data Prinzipien kann diese Information quasi online bezogen werden und bleibt somit aktuell.

Der über Linked Data erreichbare höhere Automatisierungsgrad reduziert somit die Kosten für die Verwaltung analytischer Datenbestände. Reduzierte Kosten führen wie-

derum zu einem verbesserten Kosten-Nutzen-Verhältnis und damit zu einem Gewinn für Unternehmen.

8.5.3 Aggregationshierarchien

Bisher wurde geschildert, wie Daten-Würfel mit zusätzlichen Daten erweitert werden können. Dabei konnte der Datenbestand, der zur Datenanalyse zur Verfügung steht, auf unterschiedliche Arten vergrößert werden. Aggregationshierarchien, wie sie bereits in vorherigen Abschnitten beschrieben wurden, werden nur indirekt dazu verwendet. Diese Hierarchien enthalten Wissen, wie Daten semantisch korrekt gruppiert werden können, damit die Aggregationsfunktion richtige Werte liefern kann. Diese Hierarchien sind zum Teil nicht trivial und können wiederum komplexe Graphen bilden. Der Datenbestand wird also um das Wissen erweitert, wie Daten sinnvoll aggregiert werden können. Vorgefertigte Aggregationshierarchien vermindern das Risiko falscher Berechnungen und ermöglichen Sichten auf die Daten, die zuvor nicht möglich waren.

Unterschiedliche Ansätze sind denkbar, wie diese Hierarchien nun mit Hilfe von Linked Data automatisiert erstellt werden können. Zunächst könnten sie aus tatsächlich verwendeten Anfragen abgeleitet werden, um dann bei anderen Daten-Würfeln mit gleichen Dimensionen wiederverwendet zu werden. Alternativ könnten auch bestehende Hierarchien auf Schema- oder Instanz-Ebene verwendet werden, wie sie in öffentlichen Datensätzen vorkommen. Wiederum spielt die semantische Beschreibung der Daten eine entscheidende Rolle. Durch die Explizierung hierarchischer Beziehungen ermöglichen semantische Beschreibungssprachen die Einbeziehung externer, offener Datenbestände. Hierarchische Beziehungen wie jene zwischen Regionen und Ländern können aus Linked Open Data Quellen wie z. B. Geonames.org automatisiert ergänzt werden. Damit erhöhen sich die Analysemöglichkeiten der Daten und potentiell die gewinnbaren Erkenntnisse. Ohne die Kombination aus semantischer Beschreibung und offenen Daten müsste eine solche Erweiterung wiederum von Datenbestand zu Datenbestand unterschiedlich gelöst werden. Dies erhöht die Kosten und damit verbunden ändert sich das Kosten-Nutzen-Verhältnis.

8.5.4 Provenance

Mit Provenance bezeichnet man Informationen über den Ursprung von Daten. Dazu gehören Fakten über die involvierten Personen und den Prozess, durch den die Daten entstanden oder manipuliert wurden. Durch diese Informationen kann nachvollzogen werden, aus welchen Quellen die Daten stammen und wer damit interagiert hat. Im Kontext der Datenintegration mehrerer interner und externer Datensätze handelt es sich um eine sinnvolle Erweiterung. Die Provenance-Information kann beispielsweise auf Daten-Würfel- oder sogar auf Fakten-Ebene protokolliert werden und zeigt für jede Manipulation eine verantwortliche Person auf. Dies kann von großer Wichtigkeit sein, wenn es darum geht,

Entscheidungsprozesse fundiert zu unterstützen. Berechnungen werden nachvollziehbarer und fremde Datenbestände bleiben weiterhin als solche identifizierbar.

Das World Wide Web Konsortium (W3C) hat mit der PROV Ontologie [9, 16] eine *W3C Recommendation* entworfen, mit der Provenance-Informationen als RDF Daten gespeichert werden können. Hierzu wurden Strukturen definiert, mit denen sich die Entitäten, Aktivitäten und Akteure beschreiben lassen. Die Integration eines externen Datensatzes in einen internen Daten-Würfel könnte wie folgt beschrieben werden. Bei den beiden Datensätzen handelt es sich um Entitäten, über die Provenance-Informationen gespeichert werden. Die Aktivität ist ein Integrationsprozess, der zu einem gewissen Zeitpunkt durchgeführt wurde und sich auf die beiden Entitäten bezieht. Der Akteur ist schließlich die Person oder auch die Software, die den Prozess durchgeführt hat. Zusätzlich definiert die Ontologie mehrere semantische Beziehungen zwischen den genannten Strukturen, um die Bedeutung der Zusammenhänge zu verfeinern.

Die natürliche Fähigkeit von RDF und Linked Data Datenbestände einfach um maschinen-lesbare Zusatzinformation zu ergänzen, ermöglicht die ansonsten nur mühsam erreichbare Beschreibung der Datenherkunft sowie aller durchgeführten Transformationsprozesse. Diese Provenance-Information erhöht somit das Vertrauen in die Analyseergebnisse und erlaubt die vollständige Nachvollziehbarkeit derselbigen. Das Risiko der Fehlinformation sinkt.

8.6 Zusammenfassung

Data-Warehouse Systeme sind im Unternehmensbereich bereits weit verbreitet. Es existieren umfangreiche Literatur und Forschung zu den Data-Warehousing Konzepten, wie die multidimensionale Datenstruktur, deren Schemata und die OLAP-Operatoren.

Das kürzlich entwickelte *RDF Data Cube Vokabular* ermöglicht nun die Kombination von Data-Warehouse Systemen mit semantischen Beschreibungssprachen und damit verbunden die Bereitstellung und Integration analytischer Datenbestände als Linked (Open) Data. Das RDF Data Cube Vokabular vereinfacht die Integration von Daten und reduziert somit die Kosten für den Betrieb eines Data-Warehouses. Zusätzliche Datenpunkte können aus offenen (Linked Data) Datenquellen in das System eingepflegt werden, wobei auch das Schema flexibel angepasst werden kann.

Mit RDF ist es zusätzlich möglich semantisches Wissen in disambiguierter Form zu hinterlegen. Dies ist besonders bei der halbautomatischen Schemaintegration von Vorteil. Wissen aus der Linked Open Data Cloud macht es möglich, Äquivalenzen zwischen Konzepten herzuleiten und daraus Vorschläge zu generieren. Außerdem können Klassen- und Aggregationshierarchien abgeleitet werden und somit bestehende Informationen wiederverwendet bzw. erweitert werden. Diese Funktion erhöht die Analysemöglichkeiten bis hin zu Korrelationen, welche zuvor nicht erkannt werden konnten.

Die Entwicklung dieser RDF-basierten Data-Warehouses liegt derzeit im Fokus der Forschung und es existieren bereits erfolgreiche Prototypen. Es bleibt jedoch abzuwarten,

inwieweit diese neuen Möglichkeiten auch in kommerzielle Data-Warehouse Produkte Einzug halten und die existierenden Produkte um Linked Data ergänzen.

Acknowledgments Die vorgelegte Arbeit wurde im Rahmen des CODE Projektes entwickelt. Dieses Projekt wird durch das 7. Rahmenprogramm der EU unter der Identifikationsnummer 296150 gefördert.

Literatur

1. Lehner, Wolfgang 2003. *Datenbanktechnologie für Data-Warehouse-Systeme. Konzepte und Methoden*. dpunkt Verlag
2. Inmon, William 1996. *Building the data warehouse*. John Wiley & Sons
3. Kimball, Ralph, und Joe Caserta. 2004. *The data warehouse ETL toolkit*. John Wiley & Sons
4. Chaudhuri, Surajit, und Umeshwar Dayal. 1997. An overview of data warehousing and OLAP technology. *SIGMOD* 26(1): 65–74
5. Codd, E.F., S.B. Codd, und C.T. Salley. 1993. *Providing OLAP (on-line Analytical Processing) to User-analysts: An IT Mandate*. E. F. Codd & Associates
6. Kämpgen, Benedikt, Sean O'Riain, und Andreas Harth. 2012. *Interacting with Statistical Linked Data via OLAP Operations* Proceedings of Interacting with Linked Data (ILD 2012), workshop co-located with the 9th Extended Semantic Web Conference, Mai., 36–49
7. Manola, Frank, und Eric Miller. 2004. *RDF Primer. W3C Recommendation*. http://www.w3.org/TR/rdf-primer/
8. Cyganiak, Richard, und Dave Reynolds. 2013. *The RDF Data Cube Vocabulary. W3C Candidate Recommendation*. http://www.w3.org/TR/2013/CR-vocab-data-cube-20130625/
9. Lebo, Timothy, Satya Sahoo, und Deborah McGuinness. 2013. *PROV-O: The PROV Ontology. W3C Recommendation*. http://www.w3.org/TR/prov-o/
10. Prud'hommeaux, Eric, und Andy Seaborne. 2013. *SPARQL 1.1 Overview. W3C Recommendation*. http://www.w3.org/TR/sparql11-overview/
11. Jim, Gray et al, 1997. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery* 1(1): 29–53
12. Batini, C., M. Lenzerini, und S. B. Navathe. Dezember 1986. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys* 18(4)
13. Rahm, Erhard, und Philip Bernstein. Dezember 2001. A survey of approaches to automatic schema matching. *The VLDB Journal* 10(4)
14. Zapilko, Benjamin, und Brigitte Mathiak. 2011. *Performing Statistical Methods on Linked Data* International Conference on Dublin Core and Metadata Applications
15. Pérez, Jorge, Marcelo Arenas, und Claudio Gutierrez. August 2009. Semantics and Complexity of SPARQL. *ACM Transactions on Database Systems* 34(3)
16. Zhao, Jun, und Olaf Hartig. 2012. *Towards Interoperable Provenance Publication on the Linked Data Web* Proceedings of the 5th Linked Data on the Web (LDOW) Workshop at the World Wide Web Conference (WWW), Lyon, France

17. o.V. 1999. *ISO Database Language SQL: Amendment 1: On-line Analytical Processing (SQL/OLAP). Final Proposed Draft Amendment*

18. Spofford, George, Sivakumar Harinath, Chris Webb, Dylan Hai Huang, und Francesco Civardi. 2006. *MDX-Solutions: With Microsoft SQL Server Analysis Services 2005 and Hyperion Essbase*. Wiley

19. Stegmaier, F., C. Seifert, R. Kern, H. Patrick, S. Bayerl, M. Granitzer, H. Kosch, S. Linstaedt, B. Mutlu, V. Sabol, K. Schlegel, und S. Zwicklbauer. 2013. *Unleashing Semantics of Research Data* Proceedings of the 2nd Workshop on Big Data Benchmarking

20. Paulheim, Heiko 2012. *Generating Possible Interpretations for Statistics from Linked Open Data* Proceedings; 9th Extended Semantic Web Conference, ESWC 2012. Lecture Notes in Computer Science The Semantic Web: Research and Applications. Berlin: Springer, 560–574

Linked Data Reasoning

9

Jens Lehmann und Lorenz Bühmann

Zusammenfassung

In diesem Kapitel beschreiben wir die Grundlagen des Reasonings in RDF/OWL-Wissensbasen. Wir gehen darauf ein, welche unterschiedlichen Arten des Reasonings es gibt, geben einen Überblick über verwendete Verfahren und beschreiben deren Einsatz im Linked Data Web.

9.1 Einleitung

Die Publikation von Informationen als Linked Data bietet Möglichkeiten maschinenlesbare Daten auszutauschen und miteinander zu integrieren. Ein Vorteil des Einsatzes von semantischen Technologien, insbesondere der W3C-Standards RDF, RDFS und OWL, ist in diesem Fall die klar definierte *Semantik*. Diese erlaubt es Anwendungen die Daten nicht nur einzulesen, sondern auch zu „verstehen".

In diesem Beitrag befassen wir uns mit diesem „Verstehen" der Daten, indem wir zeigen, wie Maschinen auf Basis strukturierter Daten Schlussfolgerungen ziehen. Dazu verwenden wir den Begriff *Reasoning*, der weitaus geläufiger als eingedeutschte Formen ist. Insbesondere möchten wir dem Leser zuerst einen Überblick über die verschiedenen Arten von Reasoning geben und so ein Grundverständnis der Thematik und der ihr zugrundeliegenden Technologien vermitteln. Darauf aufbauend möchten wir konkrete Reasoning-Verfahren skizzieren und anschließend Anwendungsszenarien mit einem Fokus auf das Linked Enterprise Data Management beschreiben.

Als laufendes Beispiel werden wir uns, soweit möglich, auf folgendes Szenario beziehen (Abb. 9.1):

J. Lehmann ✉· L. Bühmann
Institut für Informatik, Universität Leipzig, Leipzig, Deutschland
e-mail: lehmann@informatik.uni-leipzig.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_9

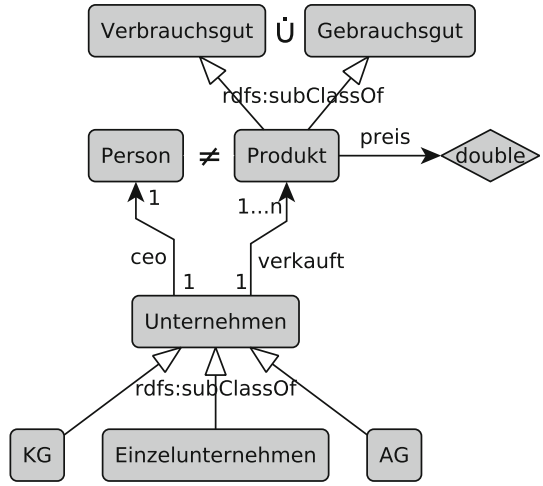


Abb. 9.1 Ausschnitt aus einer Unternehmensontologie für Zwecke des Reasonings

Beispiel 1 (Laufendes Beispiel)
Hierbei beschreiben wir einen kleinen Ausschnitt aus einer Unternehmensontologie. In dieser gibt es Unternehmen, die sich in Einzelunternehmen, KG und AG unterteilen. Diese Unternehmen verkaufen Produkte, welche Gebrauchs- oder Verbrauchsgüter sind. Zudem besitzt jedes Produkt einen Preis. Außerdem haben die Unternehmen einen CEO der Kategorie „Person“. Zusätzlich ist bekannt, dass Personen und Produkte nicht dasselbe sein können.

Eine Darstellung des Schemas in Form von Tripeln könnte folgendermaßen aussehen:

```
:ceo        rdfs:domain :Unternehmen; rdfs:range :Person.
:verkauft   rdfs:domain :Unternehmen; rdfs:range :Produkt.
:preis      rdfs:domain :Produkt; rdfs:range  xsd:double.
:Produkt    rdfs:subClassOf
  [owl:unionOf(:Gebrauchsgut, :Verbrauchsgut)].
:Person     rdfs:subClassOf
       [owl:intersectionOf(
        [owl:complementOf :Gebrauchsgut],
        [owl:complementOf :Verbrauchsgut])].
:Unternehmen rdfs:subClassOf
  [rdf:type owl:Restriction; owl:onProperty :ceo;
   owl:someValuesFrom :Person],
  [rdf:type owl:Restriction; owl:onProperty :verkauft;
   owl:someValuesFrom :Produkt].
:Gebrauchsgut rdfs:subClassOf
  [owl:complementOf :Verbrauchsgut].
```

Anhand dieses Beispiels werden wir in Abschn. 9.2 die verschiedenen Arten des Reasonings erläutern. In den Abschn. 9.3 und 9.4 erläutern wir sogenannte regelbasierte und

tableaubasierte Verfahren. Danach wird in 9.5 der Einsatz solcher Verfahren in Unternehmen erläutert. In Abschn. 9.6 schließen wir das Kapitel mit einem Ausblick auf zukünftige Entwicklungen ab.

9.2 Arten von Reasoning

Reasoning selbst ist ein relativ allgemeiner Prozess, welcher dazu genutzt werden kann basierend auf bereits vorhandenem Wissen neues Wissen abzuleiten, Erklärungen für Fakten zu finden, Vorhersagen zu treffen u. a. m. Im Rahmen dieses Kapitels werden wir die drei gängigsten Arten von Reasoning unterscheiden und genauer erläutern: Die erste Methode ist das sogenannte *Induktive Reasoning*, bei der versucht wird auf Basis einer Menge von Beobachtungen etwas Allgemeingültiges herzuleiten. Das heißt der Weg führt hier „vom Speziellen zu etwas Allgemeinem“. Bei der zweiten Form, dem *Deduktiven Reasoning*, wird dagegen von einer Menge von allgemein geltenden Regeln ausgegangen und unter Anwendung derselben versucht etwas Konkretes, möglicherweise Neues herzuleiten. Man kann sich diese Vorgehensweise also prinzipiell als den Prozess „vom Allgemeinen zu etwas Speziellem“ vorstellen. Neben diesen beiden Verfahren gibt es noch eine dritte Form von Reasoning, das *Abduktive Reasoning*. Hierbei wird vergleichbar dem Induktiven Reasoning (üblicherweise) von einer Menge von Beobachtungen ausgegangen, welche aber zusätzlich unvollständig sein kann. Für diese Menge wird dann versucht eine „passende“ Erklärung zu finden. Im Folgenden zeigen wir Reasoning anhand eines einfachen Beispiels.

9.2.1 Deduktives Reasoning

Das deduktive Reasoning ist die geläufigste der drei Arten von Reasoning. In vielen Fällen wird Reasoning mit deduktivem Reasoning gleichgesetzt aufgrund der erheblichen Unterschiede zu den beiden anderen Arten. Innerhalb dieses Buchkapitels wollen wir jedoch eine breitere Perspektive bieten. Um die Begriffe näher zu bringen, verwenden wir hier sehr einfache Beispiele und verweisen auf Abschn. 9.5 für reale Anwendungsmöglichkeiten.

Beispiel 2 (Deduktives Reasoning)

Gegeben:	Alles was einen CEO hat, ist ein Unternehmen. `ReasoningAG` hat einen CEO.
Deduktives Reasoning:	`ReasoningAG` ist ein Unternehmen.

In diesem Beispiel sind zwei Informationen gegeben, aus denen eine dritte Information geschlussfolgert werden kann. Das deduktive Reasoning zeichnet sich dadurch

aus, dass die Schlussfolgerung zwangsläufig aus den gegebenen Informationen folgt: Da `ReasoningAG` einen CEO hat, muss es ein Unternehmen sein. Wie wir in den Abschn. 9.3 und 9.4 zeigen, können diese Schlussfolgerungen mit Reasoning-Verfahren berechnet werden. Innerhalb einer Sprache wie OWL DL kommt jedes solche Verfahren, welches berechnet, ob eine Aussage aus anderen Aussagen folgt, immer zur gleichen Lösung.[1]

9.2.2 Induktives Reasoning

Obiges Beispiel kann etwas abgewandelt werden um induktives Reasoning zu erläutern:

Beispiel 3 (Induktives Reasoning)

Gegeben: `ReasoningAG` hat einen CEO. `ReasoningAG` ist ein Unternehmen.
Induktives Reasoning: Alle Unternehmen haben einen CEO.

In diesem Fall sind zwei Fakten gegeben. Aus diesen Fakten kann deduktiv nicht geschlussfolgert werden, dass alle Unternehmen einen CEO haben. Wir wissen lediglich, dass dies für ein konkretes Unternehmen zutrifft: `ReasoningAG`. Beim induktiven Reasoning geht es also darum konkrete Fakten zu verallgemeinern und in einer allgemeineren schematischen Aussage zusammenzufassen. Inwiefern diese Verallgemeinerung tatsächlich der Wahrheit entspricht, kann entweder durch die Prüfung durch einen Menschen (wie in diesem Fall möglich) oder z. B. ein Kreuzvalidierungsverfahren herausgefunden werden. Je mehr konkrete Fakten die Schlussfolgerung unterstützen, desto besser. Wenn also in diesem Fall neben `ReasoningAG` noch zahlreiche andere Unternehmen einen CEO haben, dann kann ein induktiver Reasoner mit höherer Sicherheit lernen, dass alle Unternehmen einen CEO haben. Wichtig ist beim induktiven Reasoning auch anzumerken, dass es nicht nur dazu dienen kann, absolute Wahrheiten herauszufinden, sondern auch schwächere Zusammenhänge. Zum Beispiel könnte ein induktiver Reasoner im E-Commerce Bereich lernen, dass Personen, die sich für den ersten Teil einer Filmtrilogie interessieren, auch Interesse am zweiten und dritten Teil haben. In diesem Fall ist es offensichtlich, dass es sich nicht um eine absolute Wahrheit handelt, da es zweifelsohne Personen gibt, die einer Filmtrilogie nach dem ersten Teil den Rücken kehren. Dennoch kann der obige Zusammenhang für den Betreiber eines Online-Shops sehr sinnvoll und wichtig sein, um Kunden gezielt Angebote zu unterbreiten.

[1] Wir setzen hier voraus, dass die Verfahren korrekt und vollständig sind, d. h. dass die Verfahren fehlerfrei jede Schlussfolgerung berechnen können. In der Praxis wird teilweise eine dieser Eigenschaften „geopfert“ um bessere Performance zu erzielen.

9.2.3 Abduktives Reasoning

Unser Beispiel kann wiederum adaptiert werden, um abduktives Reasoning zu erklären:

Beispiel 4 (Abduktives Reasoning)

Gegeben: Alle Unternehmen haben einen CEO. `ReasoningAG` hat einen CEO.
Abduktives Reasoning: `ReasoningAG` ist ein Unternehmen.

In diesem Fall ist eine schematische Aussage und ein konkreter Fakt gegeben. Analog zum induktiven Reasoning kann auch hier die präsentierte Aussage, dass `ReasoningAG` ein Unternehmen ist, nicht logisch geschlussfolgert werden. Statt eines Unternehmens könnte `ReasoningAG` beispielsweise ein Produkt sein. Beim abduktiven Reasoning geht es also darum konkrete Fakten zu finden, die als Erklärung für unser bekanntes Wissen verwendet werden können. Diese Fakten sollten möglichst plausibel sein, aber wie bereits beim induktiven Reasoning wird auch hier eine separate Prüfung z. B. durch einen Menschen benötigt. Abduktion kann zum Beispiel zum Vervollständigen von Wissen eingesetzt werden. In einigen Fällen wurden abduktive Reasoner sogar eingesetzt, damit Roboter komplett selbständig wissenschaftliche Vermutungen anstellen und diese anschließend prüfen können [19], um so letztendlich Erkenntnisse zu gewinnen.

9.3 Leichtgewichtiges regelbasiertes Reasoning

Die Semantik von RDF und RDFS [12], sowie das Reasoning in OWL RL [26] kann durch sogenannte Ableitungsregeln repräsentiert werden. Intuitiv handelt es sich bei einer Ableitungsregel um eine „Wenn-Dann“ Regel, die festlegt (Regelkopf, Konklusion), welche Information aus der vorhandenen Wissensbasis (Regelkörper, Prämisse) folgt. Der Regelkörper besteht dabei für gewöhnlich aus RDF Tripeln, in denen Variablen an jeder Position (Subjekt, Prädikat, Objekt) vorkommen können. Der Regelkopf umfasst die Konsequenzen, wobei jede davon ebenfalls ein RDF Tripel repräsentiert, mit der Einschränkung, dass keine Variablen auftreten dürfen, die nicht auch im Regelkörper verwendet werden. Die Liste von RDF/RDFS Regeln ist in [12] definiert, für OWL RL findet man die Regeln unter [26].

Beispiel 5 (Ableitungsregel)
Als ein Beispiel sei hier die RDFS Ableitungsregel *rdfs9* für die Subklassenrelation (rdfs:subClassOf) aufgeführt:

```
WENN (?x, rdf:type, ?c), T(?c, rdfs:subClassOf, ?d)
DANN T(?x, rdf:type, ?d)
```

Die Regel kann dabei folgendermaßen verstanden werden: Wenn es eine Instanz `x` der Klasse `c` gibt, und `c` als Subklasse der Klasse `d` definiert ist, dann ist `x` auch Instanz von `d`.

Üblicherweise unterscheidet man zwischen zwei Vorgehensweisen bei der Anwendung von Regeln um zu einer Konklusion zu gelangen:

1. Man beginnt mit der gegebenen Menge an Fakten und wendet rekursiv die Regeln „vorwärts“ an, um neue Fakten zu erzeugen. Dieses Vorgehensweise wird auch als *Forward-Chaining* bezeichnet.
2. Man beginnt mit einer Hypothese und versucht durch rekursive Anwendung der Regeln Fakten zu finden, die diese Hypothese belegen. Die Regeln werden dabei „rückwärts“ angewendet, weshalb man dieses Prinzip auch als *Backward-Chaining* bezeichnet.

Ein einfaches Beispiel für Reasoning in OWL RL könnte dabei folgendermaßen aussehen:

Beispiel 6 (Forward Chaining in OWL RL)
Nehmen wir einmal folgendes Wissen basierend auf unserem eingangs definierten Schema an: Es gibt ein Unternehmen `ReasoningAG`, das `ReasonerXYZ` verkauft. Anhand der Daten ist allerdings nicht explizit bekannt, um was es sich bei `ReasonerXYZ` handelt. Aus dem Schema wissen wir bereits, dass der Range der Relation `verkauft` ein Produkt ist. Mit Hilfe der OWL RL Regel *prp-rng*

```
WENN T(?p, rdfs:range, ?c), T(?x, ?p, ?y)
DANN T(?y, rdf:type, ?c)
```

können wir ableiten, dass `ReasonerXYZ` ein Produkt ist.

Erweitern wir nun unsere Wissensbasis um den Fakt, dass `ReasonerXYZ` der CEO von `ReasoningAG` ist. Aus unserer Ontologie wissen wir, dass jeder CEO eine Person ist, d. h. nach obiger Regel ist auch `ReasonerXYZ` eine Person. Zusätzlich ist aus dem Schema bekannt, dass ein Produkt keine Person ist. Wenden wir jetzt OWL RL Regel *cax-dw* an, welche definiert ist als

```
WENN T(?c1, owl:disjointWith, ?c2), T(?x, rdf:type, ?c1),
     T(?x, rdf:type, ?c2)
DANN Widerspruch
```

so erhalten wir offensichtlich einen Widerspruch, denn `ReasonerXYZ` ist sowohl ein Produkt als auch eine Person, jedoch ist dies explizit in unserem Schema untersagt.

9.4 Tableau-basiertes Reasoning

Während für RDF/RDFS und einige OWL 2 Profile meist regelbasierte Methoden für deduktives Reasoning verwendet werden können (siehe Abschn. 9.3), ist dies für OWL 2 nicht möglich, da man bedingt durch die höhere Ausdrucksmächtigkeit nicht in der Lage ist, eine vollständige Menge von Regeln zu definieren. OWL basiert auf sogenannten Beschreibungslogiken [3], so dass es sich anbietet auf deren Beweisverfahren zurückzugreifen. Die in der Praxis mit am häufigsten verwendete Methode ist das Tableauverfahren. Wir verzichten hier bewusst auf theoretische Grundlagen und werden im Folgenden nur eine sehr grobe Übersicht darüber geben. Für eine ausführliche Beschreibung verweisen wir daher auf [3].

Die Grundidee beim Tableauverfahren ist, dass man eine Aussage versucht zu beweisen, indem man annimmt, dass die gegenteilige Aussage gilt und die gegebene Wissensbasis dann auf Erfüllbarkeit testet. Man zählt das Tableauverfahren deshalb auch zu den sogenannten Widerlegungskalkülen. Die Erfüllbarkeit einer Aussage wird getestet, indem versucht wird ein minimales Modell zu konstruieren. Dazu wird schrittweise unter Anwendung von Tableau-Erweiterungsregeln (ein Auszug siehe Tab. 9.1) ein Tableau aufgebaut. Ein Tableau ist ein Baum, welcher solch ein Modell repräsentiert. Die Knoten in dem Baum stehen dabei für Individuen und die Kanten stellen Beziehungen zwischen den Individuen dar. Jeder Knoten ist mit den Konzepten markiert, zu denen das Individuum gehört und jede Kante ist mit der Relation ausgezeichnet, welche die Beziehung beschreibt. Je nach anzuwendender Regel wird u. a. das Tableau um neue Knoten erweitert, Markierungen bestehender Knoten werden mit weiteren Konzepten versehen, oder es entstehen Verzweigungen im Baum. Ein Pfad vom Wurzelknoten zu einem Blattknoten wird als abgeschlossen bezeichnet, wenn entlang des Pfades Knoten auftreten, die widersprüchliche Aussagen repräsentieren, d. h. Knoten, deren Markierung sowohl ein Konzept als auch dessen Negation enthalten. Ein Tableau ist abgeschlossen, wenn alle Pfade abgeschlossen sind. In diesem Fall wurde die Unerfüllbarkeit der Wissensbasis gezeigt, da kein widerspruchsfreies Modell konstruiert werden konnte.

Betrachten wir erneut die erweiterte Wissensbasis aus Beispiel 9.3 und testen deren Erfüllbarkeit, d. h. wir suchen nach einem Modell oder finden einen Widerspruch. Zunächst einmal haben wir nach Regel $C_{\mathcal{A}}$ zwei Knoten im Tableau: Knoten `:ReasoningAG`, welcher das Label `:Unternehmen` enthält sowie einen Knoten `:ReasonerXYZ`, vorerst ohne Label. Analysieren wir zuerst die `:verkauft` Relation aus unserer Ontologie. Wir wissen, dass `ReasoningAG` `ReasonerXYZ` verkauft. Nach Regel $R_{\mathcal{A}}$ fügen wir eine Kante von `:ReasoningAG` nach `:ReasonerXYZ` hinzu, welche die Bezeichnung `:verkauft` erhält. Als nächstes können wir nun Regel $\forall$ anwenden, wodurch Knoten `:ReasonerXYZ` das Label `:Produkt` erhält. Aus unserer Ontologie wissen wir, dass ein Produkt entweder ein Gebrauchsgut oder ein Verbrauchsgut sein kann, so dass wir hier Regel $\sqcup$ nutzen können. Dadurch erhalten wir eine Verzweigung im Tableau, denn es besteht die Möglichkeit, dass `:ReasonerXYZ` ein Gebrauchsgut oder aber ein Verbrauchsgut ist. Bis hierhin haben wir soweit alles hinsichtlich Relation `:verkauft` getestet und das im

Tab. 9.1 Auszug aus Tableau-Erweiterungsregeln basierend auf OWL2 DL

Name	Auswahl	Aktion
$C_{\mathcal{A}}$	`:x rdf:type :C`	Füge neuen Knoten `:x` mit dem Label `:C` hinzu.
$R_{\mathcal{A}}$	`:a :p :b`	Füge Kante mit Label `p` von Knoten `a` zu Knoten `b` hinzu.
$\sqcap$	`:x rdf:type [owl:intersectionOf (:A, :B)]`	Füge `:A` und `:B` zu Knoten `:x` hinzu.
$\sqcup$	`:x rdf:type [owl:unionOf (:A, :B)]`	Dupliziere den Zweig. Füge zum einen Zweig $C(a)$ und zum anderen Zweig $D(a)$ hinzu.
$\exists$	`:x rdf:type [rdf:type owl:Restriction; owl:onProperty :p; owl:someValuesFrom :A]`	Füge $R(a, b)$ und $C(b)$ für neues Individuum b hinzu.
$\forall$	`:x rdf:type [rdf:type owl:Restriction; owl:onProperty :p; owl:allValuesFrom :A]`	Falls es eine Kante `:p` von `:x` zu einem Knoten `:y` gibt, so füge `:A` zu `:y` hinzu.

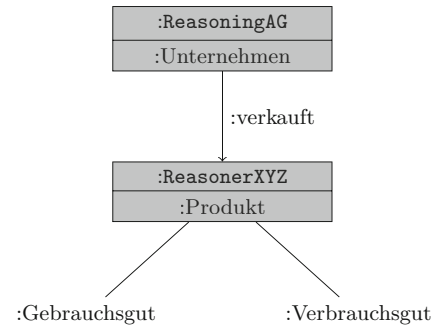


Abb. 9.2 Verzweigung innerhalb eines Tableaus entlang unterschiedlicher Produktkategorien

Folgenden graphisch veranschaulichte Tableau (Abb. 9.2) wäre nicht abgeschlossen, d. h. die Wissensbasis wäre erfüllbar.

Allerdings müssen wir für eine vollständige Ausführung des Tableauverfahrens auch noch die Relation `:ceo` untersuchen. Der Übersichtlichkeit halber betrachten wir dies zunächst separat. Wir haben nach Regel $C_{\mathcal{A}}$ wieder die beiden Knoten `:ReasoningAG` mit dem Label `:Unternehmen`, sowie `:ReasonerXYZ` ohne Label. In unserer Wissensbasis ist `:ReasonerXYZ` als CEO von `:ReasoningAG` angegeben, so dass aus Regel $R_{\mathcal{A}}$ eine Kante bezeichnet mit `:ceo` folgt. Außerdem wissen wir, dass eine Person weder Gebrauchs- noch Verbrauchsgut ist, so dass wir nach Regel $\sqcap$ die jeweilige Negation (`owl:complementOf`) als Label zum Knoten `:ReasonerXYZ` hinzufügen. Das resultierende Tableau, in Abb. 9.3 wieder als Graph dargestellt, wäre ebenso nicht

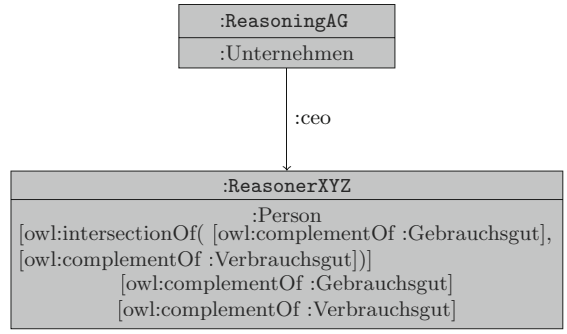


Abb. 9.3 Beispiel einer erfüllbaren Wissensbasis für die Relation :ceo

abgeschlossen, wie das für die :verkauft Relation. Das bedeutet, dass auch in diesem Fall unsere Wissensbasis erfüllbar wäre.

Wir haben beide relevanten Relationen bisher getrennt betrachtet, aber selbstverständlich ist dies im Tableauverfahren nicht der Fall. Wenn wir also beide Relationen gemeinsam analysieren und das entsprechende Tableau aufbauen, so entsteht der Graph in Abb. 9.4.

Wir sehen also, dass wir zwei Kanten vom Knoten :ReasoningAG zu :ReasonerXYZ haben. Und wir haben eine Menge von Labels für Knoten :ReasonerXYZ sowie eine Verzweigung. Viel entscheidender ist aber, dass wir jetzt einen Widerspruch in allen Pfaden haben – hervorgehoben durch die rote Kanten –, denn es kann nicht sein, dass ein Individuum (hier :ReasonerXYZ) sowohl zu einer Klasse als auch zu deren Negation gehört. Das bedeutet, dass alle Pfade im Tableau abgeschlossen sind, und es ist somit nicht möglich ein Modell für unsere Wissensbasis zu erzeugen. Insbesondere bedeutet

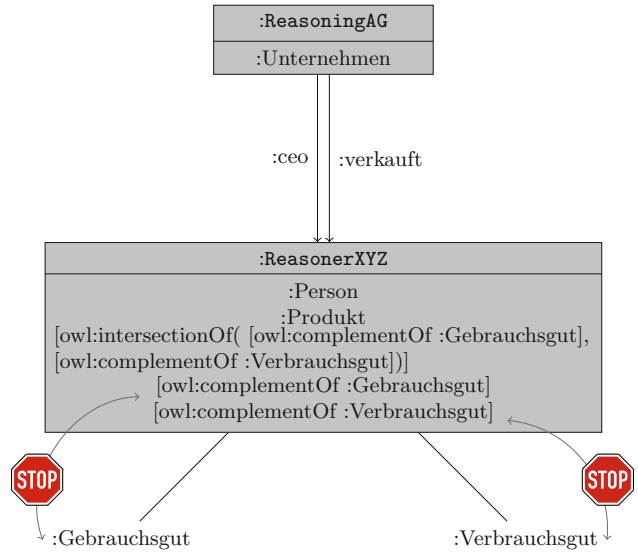


Abb. 9.4 Beispiel für einen Widerspruch in einer nicht erfüllbaren Wissensbasis

dies, dass die Wissensbasis unerfüllbar ist. Es gibt also Aussagen, die widersprüchlich sind.

9.5 Reasoning im Linked Data Web

9.5.1 Besondere Herausforderungen des Einsatzes von Reasoning im Linked Data Web

Der Einsatz von den oben eingeführten Reasoning-Technologien ist keinesfalls auf das Linked Data Web beschränkt. Es muss nicht zwangsläufig im Webkontext eingesetzt werden und kann neben den Semantic Web Standards für andere logische Sprachen eingesetzt werden. Gegenüber der eher traditionellen Nutzung in geschlossenen, lokalen Wissensbasen ergeben sich im Linked Data Web spezielle Herausforderungen, die wir hier erläutern möchten.

Der Mehrwert einer gut maschinenlesbaren Syntax zur Wissensrepräsentation wird oft unterschätzt, obwohl sie eine fundamentale Basis für den Austausch von Wissen darstellt. Das World Wide Web Consortium (W3C) hat eine Reihe von Empfehlungen verabschiedet, zum Beispiel RDF [21], OWL [14], und RIF [4], aber auch spezialisierte Vokabulare wie SKOS Simple Knowledge Organization System [25], SSN Semantic Sensor Networks [8] und Provenance [11]. Obwohl es allgemein üblich ist die Skalierbarkeit von solchen Sprachstandards zu untersuchen, wird dies durch den Einsatz im Web nochmal auf ein anderes Komplexitätsniveau gehoben. Paralleles und verteiltes Reasoning spielt hierbei eine große Rolle [18, 27].

Außerdem ist es derzeit häufig der Fall, dass Reasoning-Anwendungen auf saubere, anwendungsspezifische, manuell kuratierte Wissensbasen zugreifen. Im Linked Data Web ist solch ein Szenario oft unrealistisch. Obwohl es über Crowdsourcing-Ansätze teilweise möglich ist die Datenqualität zu erhöhen [1], müssen Reasoning-Verfahren im Linked Data Web in der Lage sein massive, verteilte Datenvolumina zu verarbeiten, die von mehreren Autoren für verschiedene Anwendungszwecke erschaffen wurden und in vielen Fällen Fehler enthalten oder nicht vollständig sind [15, 17]. Hierbei handelt es sich um typische Big Data Anforderungen, insbesondere die drei V's betreffend: *volume* (Datenmenge), *velocity* (Änderungsgeschwindigkeit der Daten) und *variety* (Diversität der Daten). Es gibt bereits zahlreiche Ansätze für diese Herausforderungen [28, 24, 2, 10], die jedoch derzeit eher noch in der Forschung als in der praktischen Anwendung zu sehen sind.

Eine weitere Herausforderung für den Einsatz von Reasoning im Linked Data Web ist die Verbreitung stabiler und intuitiver Software und Schnittstellen. Solche Werkzeuge könnten den Endanwender von der Hürde befreien Experten für Wissensrepräsentation sein zu müssen um Anwendungen zu entwickeln. Es sind zwar bereits zahlreiche Werkzeuge verfügbar, wie z. B. [22, 7, 9, 16, 29], die jedoch auf praktische Anwendungen zugeschnitten werden müssen. Zudem werden derzeit hauptsächlich Triple Stores eingesetzt um größere Datenmengen im RDF-Format zu speichern und abfragbar zu machen.

Da diese (ursprünglich) lediglich dazu dienen existierende Informationen zu ermitteln ohne Schlussfolgerungen zu ziehen, ist es notwendig parallel einen Reasoner zu betreiben um dessen Funktionalität zu erhalten. Seit Kurzem ändert sich dies durch die Ausstattung von Triple Stores mit Reasoning-Funktionen und deren Standardisierung[2]. Außerdem wird daran gearbeitet Reasoning-Funktionalitäten als Module auf Triple Stores aufzusetzen [6, 5, 13, 23].

9.5.2 Inferenz

Durch Einsatz von Reasoningverfahren kann implizites Wissen aus explizitem Wissen gefolgert werden. Ein positiver Effekt, der sich dadurch ergibt, ist, dass nur Basiswissen gespeichert werden muss und sich weitere Konsequenzen aus der Ontologie ergeben. Dadurch wird die Datenhaltung vereinfacht, insbesondere muss bei der Datenaktualisierung nur das Basiswissen geändert werden statt einer Änderung aller sich daraus ergebender Fakten. Außerdem ergibt sich gerade bei etwas komplexeren Strukturen das Problem, dass sich aus einer Menge expliziter Fakten eine Unmenge an impliziten Schlussfolgerungen ergeben, so dass es in vielen Fällen gar nicht möglich wäre diese vollständig zu speichern.

Eine Schlüsselrolle beim Reasoning kommt der Wahl der Zielsprache zu. Einfach gesagt gilt: Je ausdrucksstärker und mächtiger die verwendete Sprache, desto höher die Komplexität der Inferenzverfahren. Sehr mächtige Sprachen, wie zum Beispiel die Prädikatenlogik sind sogar unentscheidbar, dass heißt es gibt kein Verfahren, welches in endlicher Zeit bestimmen kann, ob eine Schlussfolgerung gilt. Aus diesem Grund hängt es stark vom Einsatzzweck ab, welche Sprachen und Verfahren sich eignen. Zum Beispiel muss bei sehr großen Wissensbasen häufig auf einfache regelbasierte Verfahren zurückgegriffen werden.

9.5.3 Qualitätssicherung

Einer der Haupteinsatzzwecke von Reasoningverfahren ist die Qualitätssicherung von strukturierten Wissensbasen. Wie oben erklärt, können zum Beispiel Tableauverfahren eingesetzt werden um Widersprüche zu finden. In der Praxis ist dies jedoch nur der erste Schritt zur Qualitätssicherung. Bei Wissensbasen realistischer Größe reicht es natürlich nicht aus zu wissen, dass eine Wissensbasis einen Widerspruch enthält. Um diese Widersprüche tatsächlich aufzulösen, muss die genaue Ursache für die Widersprüche gefunden werden („pinpointing“). Häufig werden dazu minimale Teile der Wissensbasis gesucht, die einen Widerspruch enthalten (sogenannte „justifications“). Diese sind oftmals klein genug, um sie manuell zu analysieren und Widersprüche aufzulösen.

[2] Siehe http://www.w3.org/TR/sparql11-entailment/, aufgerufen am 10.04.2014.

Qualitätssicherung ist auch über mehrere Ontologien hinweg möglich, das heißt es können zum Beispiel mehrere Wissensbasen innerhalb eines Unternehmens miteinander kombiniert werden und somit deren Konsistenz im Zusammenspiel geprüft werden. Unter anderem eignet sich dies zur Überprüfung sogenannter `owl:sameAs` Links.

9.6 Zukünftige Perspektiven für Reasoning im Web of Data

Für den Bereich des Reasonings im Web of Data macht es Sinn zwischen theoretischen und praktischen Entwicklungen zu unterscheiden. Auf der theoretischen Seite zeichnet sich zunehmend ein Zusammenführen von regelbasierten und Beschreibungslogik-basierten Ansätzen ab. Zum Beispiel ist OWL 2 deutlich ausdrucksstärker als OWL 1, so dass man viele Regeln inzwischen mit dieser Sprache darstellen kann [20]. Auf praktischer Ebene gibt es industrielle Bemühungen um Weblogiken einzubinden, zum Beispiel deren Unterstützung in Datenbanken wie Oracle. Insgesamt gilt es hier zu beachten, dass relationale Datenbanken den Markt dominieren. Aus diesem Grund wird intensiv daran geforscht Ontologien auf solchen Datenbanken aufzusetzen. Dieser Bereich wird als *ontology-based data access* (OBDA) [7] bezeichnet. Einige dieser Systeme, z. B. OnTop, bieten sehr umfangreichen Reasoning-Support an. Dadurch wird es ermöglicht unverändert relationale Datenbanken zu betreiben und gleichzeitig die Vorteile von Ontologien und Reasoning zu nutzen, z. B. die Integration von mehreren Wissensbasen und das automatische Finden von zusätzlichem Wissen und Widersprüchen.

Danksagung Wir bedanken uns bei Prof. Pascal Hitzler für die Diskussion zu den besonderen Herausforderungen und den zukünftigen Perspektiven von Reasoning im Linked Data Web. Die Autoren dieses Artikels werden gefördert von dem EU FP7 Projekt GeoKnow (GA no. 318159) und dem DFG-Forschungsprojekt GOLD.

Literatur

1. Acosta, Maribel, Amrapali Zaveri, Elena Simperl, Dimitris Kontokostas, Sören Auer, und Jens Lehmann. 2013. Crowdsourcing linked data quality assessment. In *12th International Semantic Web Conference* Sydney, Australia, 21–25 October 2013
2. Baader, F., und B. Hollunder. 1995. Embedding Defaults into Terminological Representation Systems. *J. Automated Reasoning* 14: 149–180
3. Baader, Franz 2003. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press
4. Boley, Harold, Gary Hallmark, Michael Kifer, Adrian Paschke, Axel Polleres, und Dave Reynolds. 2013. *RIF Core Dialect (Second Edition). W3C Recommendation*. http://www.w3.org/TR/rif-core/
5. Bühmann, Lorenz, und Jens Lehmann. 2012. Universal OWL axiom enrichment for large knowledge bases. In *Proceedings of EKAW 2012*, 57–71. Springer

6. Bühmann, Lorenz, und Jens Lehmann. 2013. *Pattern based knowledge base enrichment* 12th International Semantic Web Conference, Sydney, Australia, 21–25 October 2013
7. Calvanese, Diego, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, und Domenico Fabio Savo. 2011. The MASTRO system for ontology-based data access. *Semantic Web* 2(1): 43–53
8. Compton, Michael, Payam M. Barnaghi, Luis Bermudez, Raul Garcia-Castro, Óscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory A. Henson, Arthur Herzog, Vincent A. Huang, Krzysztof Janowicz, W.David. Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin R. Page, Alexandre Passant, Amit P. Sheth, und Kerry Taylor. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics* 17: 25–32
9. David, Jérôme, Jérôme Euzenat, François Scharffe, und Cássia Trojahn dos Santos. 2011. The Alignment API 4.0. *Semantic Web* 2(1): 3–10
10. Donini, F.M., D. Nardi, und R. Rosati. 2002. Description Logics of Minimal Knowledge and Negation as Failure. *ACM Transactions on Computational Logic* 3(2): 177–225
11. Groth, Paul, und Luc Moreau (Hrsg.). 2013. *PROV-Overview, An Overview of the PROV Family of Documents*. W3C Working Group Note 30 April 2013, 2010. Available from http://www.w3.org/TR/prov-overview
12. Hayes, Patrick. 2004. RDF Semantics. W3C Recommendation
13. Hellmann, Sebastian, Jens Lehmann, und Sören Auer. 2009. Learning of OWL class descriptions on very large knowledge bases. *International Journal on Semantic Web and Information Systems* 5(2): 25–48
14. Hitzler, Pascal, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, und Sebastian Rudolph. 2012. *OWL 2 Web Ontology Language: Primer (Second Edition). W3C Recommendation 11 December 2012*. http://www.w3.org/TR/owl2-primer/
15. Hitzler, Pascal, und Frank van Harmelen. 2010. A reasonable semantic web. *Semantic Web* 1(1–2): 39–44
16. Horridge, Matthew, und Sean Bechhofer. 2011. The OWL API: A Java API for OWL ontologies. *Semantic Web* 2(1): 11–21
17. Janowicz, Krzysztof, und Pascal Hitzler. 2012. The Digital Earth as knowledge engine. *Semantic Web* 3(3): 213–221
18. Kazakov, Yevgeny, Markus Krötzsch, und František Simančík. 2011. Concurrent classification of $\mathcal{EL}$ ontologies. In *Proceedings of the 10th International Semantic Web Conference (ISWC'11)* LNCS, Bd. 7032, Hrsg. Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, Eva Blomqvist: Springer
19. King, Ross D., Jem Rowland, Stephen G. Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, und Larisa N. Soldatova. et al, 2009. The automation of science. *Science* 324(5923): 85–89
20. Krötzsch, Markus 2010. *Description Logic Rules* Studies on the Semantic Web, Bd. 008.: IOS Press/AKA
21. Lassila, O., und R.R. Swick. 2004. *Resource Description Framework (RDF) Model and Syntax Specification. v*. http://www.w3.org/TR/REC-rdf-syntax/
22. Lehmann, Jens 2009. DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research (JMLR)* 10: 2639–2642

23. Lehmann, Jens, und Lorenz Bühmann. 2010. ORE – a tool for repairing and enriching knowledge bases. In *Proceedings of the 9th International Semantic Web Conference (ISWC2010)* Lecture Notes in Computer Science., 177–193. Springer

24. Maier, Frederick, Yue Ma, und Pascal Hitzler. 2013. Paraconsistent OWL and related logics. *Semantic Web* 4(4): 395–427

25. Miles, A., und S. Bechhofer. 2009. *SKOS Simple Knowledge Organization System Reference*. http://www.w3.org/TR/skos-reference. Zugegriffen: August 2009

26. Motik, Boris, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, znd Carsten Lutz. Dezember 2008. Owl 2 web ontology language: Profiles. World Wide Web Consortium, Working Draft WD-owl2-profiles-20081202

27. Mutharaju, Raghava, Pascal Hitzler, und Prabhaker Mateti. 2013. DistEL: A distributed EL+ ontology classifier. In *SSWS 2013, Scalable Semantic Web Knowledge Base Systems 2013. Proceedings of the 9th International Workshop on Scalable Semantic Web Knowledge Base Systems, co-located with the International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013* CEUR Workshop Proceedings, Bd. 1046, Hrsg. Thorsten Liebig, Achille Fokoue, 17–23. Sydney, Australia

28. Straccia, Umberto 2001. Reasoning within fuzzy description logics. *J. Artif. Intell. Res. (JAIR)* 14: 137–166

29. Tudorache, Tania, Csongor Nyulas, Natalya Fridman Noy, und Mark A. Musen. 2013. Webprotégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic Web* 4(1): 89–99

Teil III
Fallbeispiele

Linked Open Data Cloud

10

Anja Jentzsch

Zusammenfassung

Das Web of Data besteht aus mittlerweile 82 Milliarden RDF-Tripeln verteilt auf fast 1000 Datensätze, die verschiedene thematische Domänen abdecken. Dieser Beitrag behandelt die Vielfalt im Web of Data und zeigt diese anhand einer Analyse der Datensätze, welche im gemeinschaftlich gepflegten *LOD Cloud Data Catalog* eingetragen sind, sowie der *Linking Open Data Cloud*, welche die Datensätze und deren Beziehungen zueinander visualisiert.

10.1 Linked Open Data Cloud

Der Einsatz von Linked Data im World Wide Web wurde vom W3C *Linking Open Data-Projekt*[1] initialisiert, einem Januar 2007 gegründetes Community-Projekt. Das Gründungsziel des Projektes war es, vorhandene Datensätze, die bereits unter offenen Lizenzen verfügbar sind, entsprechend den Linked Data-Prinzipien nach RDF zu konvertieren und sie im Web zu veröffentlichen.

Abbildung 10.1 zeigt den Stand der *Linking Open Data Cloud*, welche aus dem W3C Linking Open Data-Projekt hervorgegangen ist. Sie klassifiziert die Datensätze nach thematischer Domäne und illustriert damit die Vielfalt der Datensätze im Web of Data. Jeder Knoten im Diagramm stellt einen als Linked Data veröffentlichten Datensatz dar. Die Kanten identifizieren existierende RDF-Verknüpfungen (Links) zwischen Elementen in

[1] Siehe http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData, aufgerufen am 20.04.2014.

A. Jentzsch ✉
Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Straße 2–3, 14482 Potsdam, Deutschland
e-mail: mail@anjajentzsch.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_10

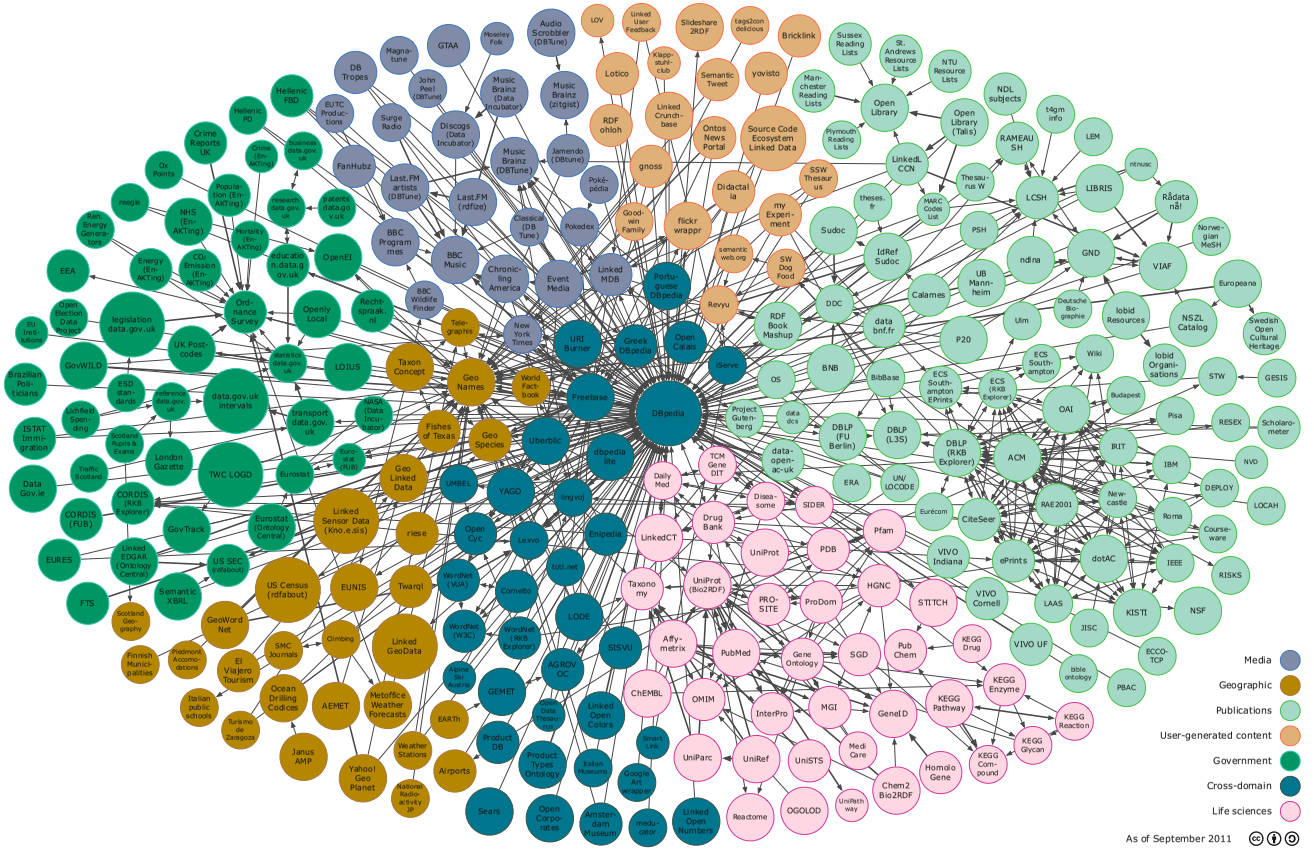


Abb. 10.1 Linking Open Data Cloud im September 2011. Die Farben klassifizieren Datensätze nach thematischer Domäne

Tab. 10.1 Anzahl der Datensätze, Anzahl der Tripel und Anzahl der RDF-Links nach thematischer Domäne, Stand September 2011

Domäne	Datensätze	Tripel	RDF-Links
Domänenübergreifend	41	4.184.635.715 (13,23 %)	63.183.065 (12,54 %)
Geographie	31	6.145.532.484 (19,43 %)	35.812.328 (7,11 %)
Öffentlicher Sektor	49	13.315.009.400 (42,09 %)	19.343.519 (3,84 %)
Medien	25	1.841.852.061 (5,82 %)	50.440.705 (10,01 %)
Bibliotheken und Bildung	87	2.950.720.693 (9,33 %)	139.925.218 (27,76 %)
Biowissenschaften	41	3.036.336.004 (9,60 %)	191.844.090 (38,06 %)
Benutzergeneriert	20	134.127.413 (0,42 %)	3.449.143 (0,68 %)
	295	**31.634.213.770**	**503.998.829**

den verbundenen Datensätzen. Die Kantenstärke korrespondiert zur Anzahl von Verknüpfungen zwischen zwei Datensätzen. Bidirektionale Kanten zeigen die Existenz von gegenseitigen Verknüpfungen zweier Datensätze.

Die LOD Community unterhält einen Katalog aller bekannten Linked Data-Datensätze, den *LOD Cloud Data Catalog*[2]. Tabelle 10.1 gibt einen Überblick über die Datensätze, die bis September 2011 katalogisiert wurden. Sie enthält die Anzahl der Datensätze in den thematischen Domänen, die Anzahl der RDF-Tripel dieser Datensätze sowie die Anzahl von RDF-Verknüpfungen zwischen den Datensätzen. Die Statistiken stammen vom *State of the LOD Cloud*-Dokument[3], welches regelmäßig zusammenfassende Statistiken über die Datensätze, welche im LOD Cloud Data Catalog katalogisiert sind, erstellt.

Insgesamt enthielten im September 2011 die 295 katalogisierten Datensätze über 31 Milliarden RDF-Tripel. Insgesamt waren 504 Millionen dieser Tripel Links, welche Entitäten in verschiedenen Datensätzen verbinden.

Im April 2014 umfasst das Web of Data mehr als 82 Milliarden RDF-Tripel verteilt auf mindestens 964 Linked Data-Datensätze, die durch mehr als 808 Millionen RDF-Links miteinander verbunden sind. Diese Statistiken werden in Tab. 10.2 veranschaulicht. Die Geschwindigkeit, in der das Web of Data wächst, erschwert leider auch die konstante Validierung und Vollständigkeit der Daten im LOD Cloud Data Catalog. Daher sind diese Zahlen als untere Grenze zu sehen.

Weiterhin existieren mehr als 17 Milliarden Tripel in Form von RDFa, Microdata und Microformats auf über 585 Millionen HTML-Seiten[4].

In den folgenden Abschnitten geben wir einen kurzen Überblick über die wichtigsten Datensätze aus jeder thematischen Domäne und verdeutlichen die Vielfalt der zur Verfügung stehenden Daten.

[2] Siehe http://www.datahub.io/group/lodcloud, aufgerufen am 20.04.2014.
[3] Siehe http://lod-cloud.net/state/, aufgerufen am 20.04.2014.
[4] Siehe http://webdatacommons.org, aufgerufen am 20.04.2014.

Tab. 10.2 Anzahl der Datensätze, Anzahl der Tripel und Anzahl der RDF-Links nach thematischer Domäne, Stand April 2014

Domäne	Datensätze	Tripel	RDF-Links
Domänenübergreifend	84	9.161.202.967 (10,87 %)	84.753.768 (10,49 %)
Geographie	46	4.830.360.509 (5,73 %)	37.314.881 (4,62 %)
Öffentlicher Sektor	139	38.246.327.678 (45,50 %)	100.354.246 (12,42 %)
Medien	41	2.568.964.515 (3,05 %)	90.677.098 (11,22 %)
Bibliotheken und Bildung	128	4.670.049.835 (5,54 %)	329.413.865 (40,75 %)
Biowissenschaften	92	7.370.024.776 (8,75 %)	7.250.244 (0,90 %)
Benutzergeneriert	49	2.283.111.223 (2,71 %)	4.935.842 (0,61 %)
Unbekannte Domäne	385	15.113.617.629 (17,94 %)	153.584.080 (19,00 %)
	964	**84.243.659.132**	**808.284.024**

10.1.1 Domänenübergreifende Daten

Einige der ersten Datensätze, die im Web of Data erschienen sind, sind nicht spezifisch für ein Thema, sondern sind domänenübergreifend. Diese multithematische Domänen umfassenden Datensätze sind entscheidend, um domänenspezifische Datensätze in einem einzigen vernetzten Datenraum zu verbinden und dadurch die Fragmentierung des Web of Data in isolierte, thematische *Dateninseln* zu vermeiden. Domänenübergreifende Datensätze sind oft zentrale Knoten im Web of Data und sind somit gut verlinkte Datensätze.

Das prominenteste Beispiel für einen domänenübergreifenden Linked Data-Datensatz ist *DBpedia*[5] [1], eine Linked Data-Version der Wikipedia. Entitäten, die Bestandteil eines Wikipedia-Artikels sind, werden automatisch einem DBpedia-URI zugeordnet, basierend auf dem entsprechenden Wikipedia-Artikel-URI. Zum Beispiel hat der Wikipedia-Artikel über die Stadt Berlin folgenden URI: http://en.wikipedia.org/wiki/Berlin. Somit hat Berlin den entsprechenden DBpedia-URI http://dbpedia.org/resource/Berlin, welcher kein URI einer Webseite über Berlin ist, sondern ein URI, der die Stadt selbst identifiziert. RDF-Aussagen, die sich auf diesen URI beziehen, werden durch Extrahieren von Informationen aus verschiedenen Teilen der Wikipedia-Artikel, insbesondere der *Infoboxen* erzeugt, welche häufig auf der rechten Seite der Wikipedia-Artikel zu sehen sind. Aufgrund der breiten thematischen Abdeckung hat DBpedia schon seit Beginn des Linking Open Data-Projektes als Hub im Web of Data gedient. Die Vielzahl der ein- und ausgehenden Links, welche DBpedia-Entitäten mit Entitäten anderer Datensätze verbinden, zeigt sich in Abb. 10.1.

Ein weiterer wichtiger domänenübergreifender Linked Data-Datensatz ist *Freebase*[6], eine frei editierbare Datenbank unter offener Lizenz, welche aus Benutzerbeiträgen sowie Datenimporten aus Quellen wie Wikipedia und Geonames entstanden ist. Freebase bietet

[5] Siehe http://dbpedia.org/, aufgerufen am 20.04.2014.
[6] Siehe http://www.freebase.com, aufgerufen am 20.04.2014.

die Objekte der Datenbank als Linked Data an und ist mit vielen anderen Datensätzen, unter anderem der DBpedia über ein- und ausgehende Links verbunden.

Weitere domänenübergreifende Datensätze sind unter anderem UMBEL[7], YAGO [2] und OpenCyc[8]. Diese sind ebenfalls mit DBpedia verlinkt, was Datenintegration in einem breiten Spektrum von miteinander verbundenen Datensätzen erleichtert.

Neben der Vielfalt an Entitätsdaten enthalten domänenübergreifende Datensätze umfassendes Domänenwissen in Form von taxonomischen Strukturen, wodurch sie eine gute Wissensbasis für Suchanwendungen darstellen.

10.1.2 Geographische Daten

Geographische Datensätze dienen oft als Verbindung unterschiedlicher anderer thematischer Datensätze. Dies zeigt sich im Web of Data zum Beispiel an *Geonames*[9], der als Hub für Datensätze dient, die geographische Daten enthalten. Geonames ist eine offen lizenzierte Geodatenbank, die Linked Data zu etwa 8 Millionen Standorten veröffentlicht.

Ein zweiter bedeutender geographischer Datensatz ist *LinkedGeoData* [3], eine Linked Data-Version des *OpenStreetMap*-Projektes, das Informationen zu mehr als 350 Millionen Standorten zur Verfügung gestellt. Wann immer möglich, werden Standorte in Geonames und LinkedGeoData mit entsprechenden Entitäten in DBpedia verknüpft, wodurch eine große Grundmenge miteinander verknüpfter Daten über geographische Standorte existiert.

Linked Data-Versionen von Eurostat[10], World Factbook[11] und US Census[12] bilden eine Brücke zwischen Statistik, Politik und Sozialgeographie. Weiterhin hat das *Ordnance Survey* (die nationale Vermessungsbehörde von Großbritannien) damit begonnen, Linked Data zur Beschreibung der Verwaltungsbereiche innerhalb Großbritanniens zu veröffentlichen[13]. Dies steht in Zusammenhang rund um die Bemühungen der Initiative *data.gov.uk*, welche im Abschn. 10.1.4 näher beschrieben wird.

10.1.3 Mediendaten

Eines der ersten großen Unternehmen, welches das Potenzial von Linked Data erkannt hat und die Grundsätze und Technologien in seine Publishing- und Content Management Workflows integriert hat, ist die British Broadcasting Corporation (BBC). Nach früheren Experimenten mit der Veröffentlichung ihres Programmkataloges als RDF, hat die BBC

[7] Siehe http://umbel.org, aufgerufen am 20.04.2014.
[8] Siehe http://sw.opencyc.org, aufgerufen am 20.04.2014.
[9] Siehe http://www.geonames.org, aufgerufen am 20.04.2014.
[10] Siehe http://datahub.io/dataset?q=eurostat&organization=lodcloud, aufgerufen am 20.04.2014.
[11] Siehe http://www4.wiwiss.fu-berlin.de/factbook/, aufgerufen am 20.04.2014.
[12] Siehe http://www.rdfabout.com/demo/census/, aufgerufen am 20.04.2014.
[13] Siehe http://data.ordnancesurvey.co.uk, aufgerufen am 20.04.2014.

im Jahr 2008 begonnen Linked Data in konventionellen Webseiten zu integrieren und dies bisher in Form von zwei großen Pilotprojekten umgesetzt. Das erste Projekt bezieht sich auf Programmdaten der BBC. Unter */programmes*[14] bietet die BBC einen URI und eine RDF-Beschreibung zu jeder TV-Episode und jedes Radioprogramm an, das über die verschiedenen Kanäle ausgestrahlt wird [4]. Das zweite Projekt bezieht sich auf Musikdaten. Unter */music*[15] veröffentlicht die BBC Linked Data zu jedem Künstler, dessen Musik von einer BBC-Radiostation gespielt worden ist. Diese Musikdaten sind mit DBpedia verknüpft und viele andere Musik-Datensätze im Web of Data verlinken wiederum darauf. Diese datensatzübergreifenden Links ermöglichen es Anwendungen, Daten aus all diesen verbundenen Quellen zu erhalten und zu integrieren, um zum Beispiel umfangreiche Künstlerprofile zu erzeugen oder aus Abspiellisten Ähnlichkeiten zwischen Künstlern zu erkennen und darauf basierend Empfehlungen zu generieren.

Ein weiterer großer Publisher von Linked Data in der Medienbranche ist die New York Times. Das Zeitungshaus hat einen bedeutenden Anteil seiner internen Schlagwörter unter einer *Creative Commons Attribution*-Lizenz als Linked Data veröffentlicht[16] und mit DBpedia, Freebase und Geonames verlinkt. Diese offen lizenzierten Daten vereinfachen den Zugang zum umfangreichen Archiv der New York Times.

10.1.4 Daten des öffentlichen Sektors

Regierungsstellen und staatliche Organisationen produzieren eine Fülle von Daten, wie Wirtschaftsstatistiken, Unternehmensregisterdaten, Grundbesitzregisterdaten, Schulberichte, Kriminalstatistiken oder Wahlergebnisse.

Das Potenzial von Linked Data, den Zugang zu Daten des öffentlichen Sektors zu vereinfachen, wird zunehmend verstanden und umgesetzt. Die beiden staatlichen Initiativen *data.gov.uk*[17] (Großbritannien) und *data.gov*[18] (Vereinigte Staaten von Amerika) veröffentlichen erhebliche Mengen von Linked Data im Web. Die Herangehensweise in den beiden Ländern unterscheidet sich jedoch leicht: Während die Vereinigten Staaten sehr große Datenmengen zu Linked Data konvertieren, fokussierte Großbritannien vorerst auf die Schaffung einer Kernstruktur für die Veröffentlichung von Linked Data, wie stabile URIs, um später zunehmende Mengen von Linked Data damit verbinden zu können [5].

Auch das European Union Open Data Portal[19] macht sowohl seine Katalog-Metadaten als Linked Data verfügbar als auch viele weitere Datensätze.

[14] Siehe http://www.bbc.co.uk/programmes, aufgerufen am 20.04.2014.
[15] Siehe http://www.bbc.co.uk/music, aufgerufen am 20.04.2014.
[16] Siehe http://data.nytimes.com, aufgerufen am 20.04.2014.
[17] Siehe http://data.gov.uk/linked-data, aufgerufen am 20.04.2014.
[18] Siehe http://www.data.gov/semantic, aufgerufen am 20.04.2014.
[19] Siehe http://open-data.europa.eu/en/linked-data, aufgerufen am 20.04.2014.

Eine weitere interessante Initiative wird von der *UK Civil Service*[20] vorangetrieben, welche begonnen hat, Stellenangebote mit RDFa zu versehen. Durch die Bereitstellung von Informationen über offene Stellen in einer strukturierten Form wird es einfacher für externe Job-Portale Stellenangebote des öffentlichen Dienstes zu integrieren [6]. Wenn mehr Unternehmen diesem Beispiel folgten, könnte die Transparenz auf dem Arbeitsmarkt deutlich erhöht werden [7].

Um die Arbeit zur Verwendung von Linked Data und anderen Web-Standards für Regierungsdaten zu koordinieren und Transparenz zu gewährleisten, hat das W3C die eGovernment Interest Group[21] gegründet.

10.1.5 Bibliotheks- und Bildungsdaten

Bibliotheken sind durch ihre Erfahrung und Expertise in der Erstellung von qualitativ hochwertigen strukturierten Daten sowie der Notwendigkeit neuer Methoden zur Verfügbarmachung ihrer Inhalte für die Umsetzung der Linked Data-Prinzipien prädestiniert. Dieser Bereich hat bereits frühzeitig die Integration von Bibliothekskatalogen auf globaler Ebene, die Verlinkung der Bibliothekskataloge (zum Beispiel nach Thema, Standort oder Epoche), die Verlinkung der Bibliothekskataloge mit externen Quellen (wie Bild- und Videoarchiven oder Wissensdatenbanken wie DBpedia), sowie die bessere Zugänglichkeit von Bibliotheksdaten mit Hilfe von Web-Standards angestrebt.

Beispiele sind die American Library of Congress[22] und die Deutsche Zentralbibliothek für Wirtschaftswissenschaften (ZBW) [8], die ihre Schlagwortkataloge als Linked Data veröffentlichen, während in Schweden sogar der komplette Inhalt von *LIBRIS* und des Swedish National Union Catalogue[23] als Linked Data zur Verfügung stehen. Auch der Katalog der *OpenLibrary*[24], ein Gemeinschaftsprojekt für die Erstellung einer Webseite für jedes Buch, das jemals erschienen ist, veröffentlicht seine Daten in RDF.

Wissenschaftliche Aufsätze aus Zeitschriften und Konferenzen sind ebenfalls im Web of Data vertreten. Hierzu gehören Projekte wie DBLP[25, 26, 27], RKBexplorer[28] und der Semantic Web Dogfood Server[29] [9].

[20] Siehe http://www.civilservice.gov.uk, aufgerufen am 20.04.2014.
[21] Siehe http://www.w3.org/egov/wiki/Main_Page, aufgerufen am 20.04.2014.
[22] Siehe http://id.loc.gov/authorities/about.html, aufgerufen am 20.04.2014.
[23] Siehe http://blog.libris.kb.se/semweb/?p=7, aufgerufen am 20.04.2014.
[24] Siehe http://openlibrary.org, aufgerufen am 20.04.2014.
[25] Siehe http://dblp.l3s.de/, aufgerufen am 20.04.2014.
[26] Siehe http://www4.wiwiss.fu-berlin.de/dblp/, aufgerufen am 20.04.2014.
[27] Siehe http://dblp.rkbexplorer.com/, aufgerufen am 20.04.2014.
[28] Siehe http://www.rkbexplorer.com/data/, aufgerufen am 20.04.2014.
[29] Siehe http://data.semanticweb.org/, aufgerufen am 20.04.2014.

Zweifelsohne werden die hohen Aktivitäten in der Bibliothekswelt zu weiteren signifikanten Linked Data-Entwicklungen in diesem Bereich führen. Besonders hervorzuheben ist der *Object Reuse and Exchange (OAI-ORE)*-Standard der *Open Archives Initiative* [10], welcher auf den Linked Data-Prinzipien basiert. Vokabulare wie OAI-ORE, Dublin Core, SKOS und FOAF bilden die Grundlage des neuen Europeana Datenmodells[30]. Die Umsetzung dieses Modells von Bibliotheken, Museen und Kultureinrichtungen, die an der Europeana teilnehmen, wird die Verfügbarkeit von Linked Data von Publikationen und kulturellem Erbe beschleunigen.

Um die weltweiten Bemühungen zur Interoperabilität der Bibliotheksdaten zu koordinieren, hat das W3C die Library Linked Data Incubator Group[31] gegründet.

10.1.6 Biowissenschaftsdaten

Linked Data wurde im Bereich der Biowissenschaften schnell akzeptiert, um die verschiedenen Datensätze, die von Forschern und Forscherinnen in diesem Bereich verwendet werden, zu verbinden. Insbesondere das Projekt Bio2RDF [11] hat mehr als 30 weit verbreitete Datensätze, einschließlich UniProt (Universal Ressource Protein), KEGG (Kyoto Encyclopedia of Genes and Genomes), CAS (Chemical Abstracts Service), PubMed und die Gene Ontology als Linked Data veröffentlicht und miteinander verlinkt.

Die W3C *Linking Open Drug Data*-Initiative (LODD)[32] hat die Pharmaunternehmen Eli Lilly, AstraZeneca, und Johnson & Johnson zusammengebracht, um in einem gemeinschaftlichen Projekt offen lizenzierte Daten über Medikamente und klinische Studien zu verknüpfen, um die Arzneimittelentwicklung zu unterstützen [12].

10.1.7 Benutzergenerierte Daten und Soziale Medien

Einige der ältesten Datensätze im Web of Data basieren auf Linked Data-Konvertierungen oder Wrappern von *Web 2.0*-Seiten mit großen Mengen benutzergenerierter Daten. Zu diesen gehören Datensätze und -dienste wie *FlickrWrappr*[33], ein Linked Data-Wrapper um den Photoservice Flickr. Diese wurden um Webseiten mit benutzergenerierten Inhalten mit nativer Linked Data-Unterstützung ergänzt, wie *Revyu.com* [13] für Testberichte und Bewertungen, und Faviki[34] zur Annotierung von Web-Inhalten mit Linked Data-URIs.

Zunächst wurden die Linked Data-Prinzipien vor allem von Forschungsprojekten und Web-Enthusiasten angenommen und umgesetzt. Diese konvertierten bestehende Datensätze nach RDF und veröffentlichten diese im Web. Alternativ implementierten sie Daten-

[30] Siehe http://pro.europeana.eu/edm-documentation, aufgerufen am 20.04.2014.
[31] Siehe http://www.w3.org/2005/Incubator/lld/, aufgerufen am 20.04.2014.
[32] Siehe http://esw.w3.org/HCLSIG/LODD, aufgerufen am 20.04.2014.
[33] Siehe http://www4.wiwiss.fu-berlin.de/flickrwrappr/, aufgerufen am 20.04.2014.
[34] Siehe http://www.faviki.com/, aufgerufen am 20.04.2014.

Wrapper um vorhandene Web-APIs. Heute werden Linked Data-Technologien verstärkt durch die Datenproduzenten angenommen und von ihnen verwendet, um einen verbesserten Zugang zu ihren Datensätzen zu gewährleisten. Im September 2011 wurden von den 295 Datensätzen in der Linked Open Data Cloud 113 (38,57 %) von den ursprünglichen Datenproduzenten veröffentlicht, während 180 (61,43 %) von Dritten veröffentlicht wurden.

10.2 Linked Data-Vokabulare in der LOD Cloud

Es ist gängige Praxis in der Linked Data Community, wenn möglich Begriffe aus bereits etablierten Vokabularen zu verwenden, um Datensätze semantisch zu beschreiben. Dies erhöht die Homogenität der Beschreibungen und damit die Verständlichkeit von Beschreibungen für die maschinelle Verarbeitung. Da das Web of Data eine Vielzahl von Themen umfasst, existieren nicht immer weit verbreitete Vokabulare, die alle Aspekte dieser Themen abdecken. Daher werden häufig auch proprietäre Begriffe definiert und mit Begriffen aus weit verbreiteten Vokabularen vermischt, um spezifische Aspekte abzudecken und den kompletten Inhalt eines Datensatzes im Web zu veröffentlichen.

Fast alle Datensätze in der Linked Open Data Cloud verwenden Begriffe der W3C-Basisvokabulare RDF, RDF Schema und OWL. Zusätzlich verwenden 191 (64,75 %) der 295 Datensätze Begriffe anderer weit verbreiteter Vokabulare. Tabelle 10.3 zeigt die Verteilung der am häufigsten verwendeten Vokabulare im Web of Data.

Diese weit verbreiteten Vokabulare decken generische Arten von Entitäten ab. So wird zum Beispiel Friend of a Friend (FOAF)[35] verwendet, um Menschen und soziale Beziehungen zu beschreiben oder das Basic Geo Vocabulary (geo)[36], um Standorte zu beschreiben. Domänenspezifische Vokabulare wie die Music Ontology (mo)[37] oder die Bibliographic Ontology (bibo)[38] gewinnen jedoch an Bedeutung.

Insgesamt verwenden 190 (64,41 %) der 295 Datensätze in der Linked Open Data Cloud proprietäre Begriffe zusätzlich zu Begriffen aus öffentlichen Vokabularen. Um es Anwendungen zu ermöglichen, die Definition dieser proprietären Begriffe automatisch aus dem Web abzurufen, sollten deren URIs dereferenzierbar sein. Richtlinien hierzu werden in der W3C Note Best Practice Recipes for Publishing RDF Vocabularies[39] gegeben. Diese bewährte Methode wird derzeit von 159 (83,68 %) der 190 Datensätze, die proprietäre Begriffe verwenden umgesetzt, während 31 (16,32 %) proprietären Begriffe nicht dereferenzierbar sind.

[35] Siehe http://xmlns.com/foaf/0.1/, aufgerufen am 20.04.2014.
[36] Siehe http://www.w3.org/2003/01/geo/wgs84_pos, aufgerufen am 20.04.2014.
[37] Siehe http://purl.org/ontology/mo/, aufgerufen am 20.04.2014.
[38] Siehe http://bibliontology.com, aufgerufen am 20.04.2014.
[39] Siehe http://www.w3.org/TR/2008/NOTE-swbp-vocab-pub-20080828/, aufgerufen am 20.04.2014.

Tab. 10.3 Verteilung der am häufigsten verwendeten Vokabulare

Vokabular	Anzahl der Datensätze
Dublin Core (dc)	92 (31,19 %)
Friend-of-a-Friend (foaf)	81 (27,46 %)
Simple Knowledge Organization System (skos)	58 (19,66 %)
Basic Geo Vocabulary (geo)	25 (8,47 %)
AKTive Portal (akt)	17 (5,76 %)
Bibliographic Ontology (bibo)	14 (4,75 %)
Music Ontology (mo)	13 (4,41 %)
Electronic Business Cards (vcard)	10 (3,39 %)
Semantically-Interlinked Online Communities (sioc)	10 (3,39 %)
Creative Commons (cc)	8 (2,71 %)

Eine zentrale Idee von Linked Data ist es, RDF-Links zwischen Entitäten im Web zu veröffentlichen. Dies gilt nicht nur für Datensätze, sondern kann auch verwendet werden, um Übereinstimmungen zwischen Begriffen verschiedener Vokabulare im Web zu veröffentlichen und somit Linked Data-Anwendungen die Datenintegration zu vereinfachen, indem Daten zwischen verschiedenen Vokabularen übersetzt werden können. Die W3C-Empfehlungen definieren folgende Eigenschaften für die Darstellung solcher Korrespondenzen (Mappings): `owl:equivalentClass`, `owl:equivalentProperty`, oder wenn eine lockeres Mapping definiert werden soll: `rdfs:subClassOf`, `rdfs:subPropertyOf` sowie `skos:broadMatch`, `skos:narrowMatch`. Nur 15 (7,89 %) der 190 Datensätze, die proprietäre Begriffe verwenden, verlinken diese zu anderen Vokabularen.

Die Webseite Linked Open Vocabularies[40] bildet einen Katalog der 421 verschiedenen Vokabulare im Web of Data (Stand: April 2014) und veranschaulicht deren Beziehungen untereinander.

Literatur

1. Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, und C. Bizer. 2014. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 5: 1–29
2. Suchanek, F.M., G. Kasneci, und G. Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web* WWW 2007, Banff, Alberta, Canada, May 8–12, 2007., Hrsg. C.L. Williamson, M.E. Zurko, P.F. Patel-Schneider, P.J. Shenoy, 697–706. ACM
3. Auer, Sören, Jens Lehmann, und S. H. 2009. LinkedGeoData – Adding a Spatial Dimension to the Web of Data. In *Proceedings of the International Semantic Web Conference*

[40] Siehe http://labs.mondeca.com/dataset/lov/, aufgerufen am 20.04.2014.

4. Kobilarov, G., T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer, und R. Lee. 2009. Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. In *The Semantic Web: Research and Applications* 6th European Semantic Web Conference, 723–737
5. Sheridan, J., und J. Tennison. 2010. Linking UK Government Data. In *Proceedings of the WWW2010 Workshop on Linked Data on the Web*
6. Birbeck, M. 2009. RDFa and Linked Data in UK government web-sites. *Nodalities Magazine* 7: 15–16
7. Bizer, C., R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, und R. Eckstein. 2005. The Impact of Semantic Web Technologies on Job Recruitment Processes. In *Proceedings of the 7. Internationale Tagung Wirtschaftsinformatik*
8. Neubert, J. 2009. Bringing the Thesaurus for Economics on to the Web of Linked Data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web*
9. Möller, K., T. Heath, S. Handschuh, und J. Domingue. 2007. Recipes for Semantic Web Dog Food – The ESWC and ISWC Metadata Projects. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference* Busan, Korea
10. Van de Sompel, H., C. Lagoze, M. Nelson, S. Warner, R. Sanderson, und P. Johnston. 2009. Adding eScience Assets to the Data Web. In *Proceedings of the 2nd Workshop on Linked Data on the Web (LDOW2009)*
11. Belleau, F., M. Nolin, N. Tourigny, P. Rigault, und J. Morissette. 2008. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics* 41(5): 706–716
12. Jentzsch, A., O. Hassanzadeh, C. Bizer, B. Andersson, und S. Stephens. 2009. Enabling tailored therapeutics with linked data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web*
13. Heath, T., und E. Motta. 2008. Revyu: Linking reviews and ratings into the web of data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 6(4): 266–273

Linked Data und Digitale Bibliotheken

11

Natalja Friesen und Christoph Lange

Zusammenfassung

Wichtige Ziele bei der Entwicklung Digitaler Bibliotheken sind Informationen leicht auffindbar zu machen, sie miteinander zu verknüpfen sowie die Inhalte der Bibliothek für Mensch und Maschine nutzbar zu machen. Wir erklären, wie Linked-Data-Technologie dazu beiträgt, diese Ziele zu erreichen. Dazu stellen wir zunächst wichtige Standards, Vokabulare und Ontologien für bibliographische (Meta-)Daten vor und diskutieren dann Herausforderungen beim Publizieren Digitaler Bibliotheken als Linked Data. Zu den Herausforderungen gehören Datenmodellierung, Mapping, sowie Verknüpfung der Daten miteinander und mit anderen Datenbeständen. Als konkrete Anwendungsfälle geben wir einen Überblick über die Europeana und die Deutsche Digitale Bibliothek (DDB), stellen aber auch weitere Digitale Bibliotheken vor, die Linked Data einsetzen. Wir schließen mit einem Ausblick auf zukünftige Entwicklungen.

11.1 Einleitung

Seit Jahrhunderten sind Bibliotheken, Museen und Archive wichtige Bewahrer und Vermittler des kulturellen Erbes. Die zunehmende Verbreitung neuer Informationstechnologien hat die Anforderungen und Ansprüche der Nutzer an diese Institutionen geändert und damit eine rasante Entwicklung befördert. Die zahlreichen Digitalisierungsprojekte, die auf nationaler und europäischer Ebene in den letzten Jahrzehnten gestartet und zum größten Teil erfolgreich abgeschlossen worden sind, schlagen eine Brücke von der Ära des Buches zur digitalen Ära. Eines der ersten digitalen Bibliotheks-Projekte ist das Internet

N. Friesen ✉ · C. Lange
Fraunhofer IAIS und Universität Bonn, 53117 Bonn, Deutschland
e-mail: natalja.friesen@iais-extern.fraunhofer.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_11

Archive [17], dessen Ursprung ins Jahr 1996 zurückgeht. Eines der größten internationalen Projekte, das Google-Books-Projekt[1], wurde im Jahr 2004 gestartet und hat das Ziel Bücher in jeder Sprache zu digitalisieren.

Die neuesten Entwicklungen im Bibliotheksbereich gehen jedoch weit über die Sicherung von Wissen durch Digitalisierung hinaus. Die Frage, wie Erreichbarkeit und Nutzbarkeit digitalisierter Inhalte verbessert werden kann, ist von großer Bedeutung. Mit solchen Fragestellungen befasst sich das Informatik-Forschungsgebiet Digitale Bibliotheken. In der Fachliteratur sind unterschiedliche Definitionen für den Begriff „Digitale Bibliothek" zu finden. Allgemein wird unter diesem Begriff eine Vielfalt von Systemen zusammengefasst, die sowohl in ihrer Funktionalität als auch in ihrem Anwendungsbereich sehr heterogen sind. Solche Systeme erstrecken sich von einer Sammlung digitaler Datenbestände einer einzelnen Bibliothek, die auf deren eigenem Server zugänglich sind[2], bis zu komplexen Informationssystemen mit innovativen Diensten, so wie die in Abschn. 11.4.2 eingehender behandelte Deutsche Digitale Bibliothek.[3] Im engeren Sinne bezeichnet man als Digitale Bibliothek eine organisierte, dauerhafte Sammlung von digitalen Inhalten, die ihren Nutzern, nach festgelegten Regeln und in definierter Qualität, spezielle Dienste für diese Inhalte bereitstellt [21].

Digitale Bibliotheken bieten sowohl für den Nutzer als auch für die bibliothekarischen Einrichtungen selbst Vorteile [2]. Die Nutzer profitieren von der ortsunabhängigen Recherchemöglichkeit, die häufig einen direkten Zugriff auf digitale Informationsressourcen wie gescannte Bücher, Manuskripte, Bilder und Filme einschließt. Es ist nicht mehr nötig, Bücher aus einer Bibliothek zu holen. Nicht zu unterschätzen ist auch die Zeitersparnis. Alle erforderlichen Informationen können schnell und bequem von einem PC oder mobilen Gerät abgerufen werden und sind immer verfügbar. Eine Bibliothek ist jetzt überall erreichbar, wo es Internet-Zugang gibt. Außerdem gewähren Digitale Bibliotheken den Zugriff auf Bestände, die aufgrund ihres historischen Wertes oder ihres Formats gar nicht oder nur unter Aufsicht eingesehen werden können und die auch in Antiquariaten nicht zu erhalten sind.

Für bibliothekarische Einrichtungen bietet die Veröffentlichung ihrer Inhalte im Internet ebenfalls signifikante Vorteile. Es ist viel einfacher Informationen aktuell zu halten, wenn Dokumente in digitaler Form und auf einem zentralen Rechner gespeichert sind. Gedruckte Materialien sind umständlich zu aktualisieren. Nicht nur das veraltete Dokument muss neu gedruckt werden, sondern theoretisch müssten auch alle Kopien der alten Version aufgespürt und ersetzt werden.

Der Hauptzweck einer Digitalen Bibliothek besteht darin, die in digitaler Form vorliegenden Informationsressourcen auffindbar zu machen. Dazu tragen eine Suchfunktion für Metadaten und Volltext, die Auffindbarkeit für Web-Suchmaschinen, sowie eine Verlin-

[1] GoogleBooks. http://books.google.com/, aufgerufen am 21.02.2014.
[2] Siehe z. B.: Universitätsbibliothek. Freie Universität Berlin. 14. Okt. 2013. http://www.ub.fu-berlin.de/digibib_neu/, aufgerufen am 21.02.2014.
[3] Deutsche Digitale Bibliothek. https://www.deutsche-digitale-bibliothek.de/, aufgerufen am 21.02.2014.

kung von anderen Webseiten bei. Außerdem soll es leicht sein, Inhalte zu einer Digitalen Bibliothek hinzuzufügen; dazu gehört, Inhalte zu verschlagworten und Metadatensätze anzulegen. Semantic-Web-Technologien und das Linked-Data-Prinzip helfen dabei, all diese Funktionen intelligenter umzusetzen, stellen allerdings spezielle Anforderungen an die Repräsentation der Datenbestände. Dieses Kapitel verfolgt nicht das Ziel einen umfassenden Überblick über Linked Data in digitalen Bibliotheken zu geben, sondern an ausgewählten Beispielen aufzuzeigen, welche neuen Perspektiven Linked Data für die Bibliothekswelt eröffnet. Das Buch „(Open) Linked Data in Bibliotheken“ [9], das sich gezielt an Personen aus Bibliothekspraxis und -management richtet, befasst sich tiefergehend mit dem Thema.

11.2 Motivation

Die folgenden Abschnitte stellen diejenigen Vorteile vor, die Linked Data für Digitale Bibliotheken besonders attraktiv machen. Die Auffindbarkeit von Informationen sowohl für Menschen als auch für Maschinen wird gewährleistet durch „coole“ URIs (Abschn. 11.2.1); das flexible Datenmodell von RDF ermöglicht die Integration heterogener Informationen (Abschn. 11.2.2), und schließlich sind viele im Zusammenhang mit Digitalen Bibliotheken relevante Datenbestände schon in Form von Linked Data verfügbar (Abschn. 11.2.3).

11.2.1 Auffindbarkeit im Web dank „cooler“ URIs

Die Linked-Data-Grundsätze empfehlen, allen Dingen von Interesse (hier vor allem: Informationsressourcen in Digitalen Bibliotheken) URIs zu geben.[4] „Coole“ URIs zeichnen sich dadurch aus, dass sie über inhaltliche Umstrukturierungen und technische Änderungen hinweg stabil bleiben und von den technischen Gegebenheiten wie Datenformaten oder der Konfiguration des Webservers abstrahieren.[5] So sollte eine Digitale Bibliothek z. B. die Person Leonardo da Vinci nicht mit dem URI http://www.kunst-bibliothek.org/users/~mueller/data.php?source=person_db.xls&version=27&format=rdf#vinci bezeichnen, sondern eher mit http://www.kunst-bibliothek.org/personen/Leonardo_da_Vinci. Coole URIs erleichtern es, von außerhalb auf Inhalte einer Digitalen Bibliothek zu verlinken; Anwender können sie leicht verschicken und als Lesezeichen speichern.

Unter einem coolen URI kann dank HTTP Content Negotiation Information sowohl für Menschen als auch für Maschinen in unterschiedlichen Formaten bereitgestellt werden [23]. Eine übliche Lösung in einer Digitalen Bibliothek ist, HTML für Menschen

[4] Berners-Lee, T. Design Issues: Linked Data. 27. Juli 2006. http://www.w3.org/DesignIssues/LinkedData.html, aufgerufen am 20.01.2010.

[5] Berners-Lee, T. Cool URIs don’t change. 1998. http://www.w3.org/Provider/Style/URI, aufgerufen am 02.10.2014.

anzubieten, RDF/XML für Maschinen, die Linked Data verstehen, und möglicherweise andere Metadatenformate (siehe Abschn. 11.3.1) für Maschinen, die zwar HTTP-Downloads durchführen können, aber kein RDF verstehen.

11.2.2 Flexibles Datenmodell und selbsterklärende Schemata

Das RDF-Datenmodell zeichnet sich aus durch Flexibilität und Erweiterbarkeit, sowohl auf der Ebene der Daten als auch auf der Ebene der Schemata.

Von der Beschreibung eines Dings[6] (z. B. eines Werks in einer Digitalen Bibliothek) kann man jederzeit auf Beschreibungen anderer, damit in Beziehung stehender Dinge (z. B. seines Autors) verweisen, weil in RDF alle Dinge – RDF nennt sie *Ressourcen* – durch URIs identifiziert werden. Wenn es für die gewünschte Art der Beziehung noch keinen geeigneten Begriff gibt, kann man ihn definieren (siehe die Ausführungen zu Metadatenvokabularen weiter unten in diesem Abschnitt), oder zunächst den allgemeinsten Link-Typ *rdfs:seeAlso* verwenden.

Gibt es zu einem Ding mehrere RDF-Datensätze, d. h. mehrere Mengen von RDF-Tripeln, die das Ding als gemeinsames Subjekt haben, so können diese Datensätze leicht zu einem einzigen vereinigt werden. Eine völlige, physische Vereinigung, d. h. Bereitstellung aller Tripel an derselben Adresse im Netz, ist dann möglich, wenn das Ding jeweils durch denselben URI identifiziert wird. In dem anderen Fall, dass ein und dasselbe Ding mit unterschiedlichen URIs bezeichnet wird und es erwünscht (oder nicht anders machbar) ist, Beschreibungen seiner Eigenschaften an unterschiedlichen Adressen im Netz bereitzustellen (etwa wenn sie von unterschiedlichen Organisationen gepflegt werden), kann man mit einem Tripel der Form `<URI1> owl:sameAs <URI2>` die Äquivalenz jeweils zweier Ressourcen ausdrücken, und die tatsächliche Vereinigung der RDF-Tripel zu URI 1 und URI 2 dem Nutzer der Daten überlassen. Syntaktisch sind diese beiden Arten der Vereinigung immer möglich. Semantisch ist sie dann möglich, wenn die Einträge in den unterschiedlichen Datensätzen sich nicht widersprechen. Unvollständige Datensätze und heterogene Datenbestände können unproblematisch als RDF-Graph modelliert werden, und statt sich auf *ein* Metadatenformat festzulegen, kann man beliebig viele RDF-Vokabulare zugleich verwenden, um alle Eigenschaften einer Ressource und alle ihre Beziehungen zu anderen Ressourcen zu modellieren.

Werden Metadatenformate selbst in Form von RDF als Linked Data realisiert – im Falle eines Metadatenformats hätte man es dann mit einem RDF-*Vokabular* zu tun, welches aber auch wieder nur ein Spezialfall eines RDF-Graphen ist –, so übertragen sich die bisher genannten Vorteile auf die Metadatenvokabulare, also auf die Schemata, nach

[6] Der Begriff „Ding“ soll hier für etwas Beliebiges stehen, über das wir Informationen haben. Wenn im Folgenden auch Personen als „Dinge“ bezeichnet werden, soll dies keine Abwertung darstellen. Wir verwenden bewusst nicht den Begriff „Objekt“, um eine Verwechslung mit dem Objekt eines aus Subjekt, Prädikat und Objekt bestehenden RDF-Tripels auszuschließen.

denen die Metadaten strukturiert sind. Das ist z. B. dann hilfreich, wenn ein Nutzer (sei es ein Mensch oder ein automatisierter Dienst) in einem Linked Dataset eine Eigenschaft einer Ressource vorfindet, die ihm noch nicht bekannt ist. Linked-Data-konforme Veröffentlichung des entsprechenden RDF-Vokabulars vorausgesetzt, kann der Nutzer einfach durch Dereferenzieren des URIs dieser Eigenschaft (RDF property) eine Dokumentation der Eigenschaft und ihrer Verwendung in Form von HTML, RDF/XML oder anderen Datenformaten nachschlagen. Somit wird das Schema selbsterklärend.

11.2.3 Gewinnung zusätzlichen Wissens

Bibliotheks-, Museums- und Archivdaten enthalten wertvolles Wissen über kulturelle Werke, das oft nirgendwo sonst zu finden ist. Eine Informationsressource sollte jedoch nicht nur aus dieser eingeschränkten Perspektive betrachtet werden: Zwar steht das kulturelle Objekt selbst im Mittelpunkt, aber auch die mit ihm zusammenhängenden Personen und Orte sind von Interesse. Informationen über Letztere finden sich oft an anderer Stelle im Web, z. B. in der Wikipedia. Solche Quellen betrachten Personen und Orte oft aus verschiedenen Blickwinkeln, stellen Inhalte in mehreren Sprachen bereit und werden ständig von der Community angereichert. Strukturierte Informationen aus Wikipedia sind unter dem Namen DBpedia als Linked Open Dataset veröffentlicht worden (siehe Abschn. 11.4.3). Ebenso liegt zahlreiches weiteres Wissen von allgemeinem Interesse, oder auch von speziellem Interesse für Digitale Bibliotheken, in Form von Linked Open Datasets vor, und diese sind oft stark miteinander verlinkt. Linked Data erleichtert die Verknüpfung einer Digitalen Bibliothek mit solchen Daten. Dadurch können den Nutzern der Digitalen Bibliothek neue Zusammenhänge offen gelegt werden; teilweise können sogar neue Zusammenhänge entdeckt werden, die zuvor nicht erkennbar waren. Durch diese Kontexterweiterung wächst schließlich die Relevanz der bibliographischen Daten in einer Digitalen Bibliothek.

11.3 Bibliographische Metadaten

Bibliothekarische Daten bestehen zum größten Teil aus Metadaten, welche eine Informationsressource strukturiert beschreiben und ihre Identifikation und Lokalisierung in einem Bibliothekskatalog ermöglichen. Der Inhalt und Umfang der Metadaten wird durch Regelwerke normiert und ergibt sich aus der in den jeweiligen Einrichtungen üblichen Katalogisierungspraxis und den anwendungsspezifischen Anforderungen. Darüber hinaus können die für die Erschließung erforderlichen Informationen je nach Typ der Ressource stark variieren. So spielen für die Beschreibung eines Museumsexponates neben den typischen Metadaten wie Titel und Künstler auch die Größe des Objektes, Material und Herstellungstechnik eine wichtige Rolle, während diese Angaben für eine Tonaufnahme irrelevant sind. Dort sind aber Merkmale wie Stilistik, Haupt- und Nebeninstrumente,

Genre, Tempo, Tonart, Dynamik von großer Bedeutung. Entsprechend den Anforderungen einer kulturellen Einrichtung wird ein mehr oder weniger umfangreiches und komplexes Metadatenschema entwickelt, in dem alle Erschließungselemente genannt und spezifiziert werden [22]. Jede Ressource wird bezüglich dieses Schemas erfasst und in einer Datenbank gespeichert.

Das Modellieren der bibliothekarischen Daten hat das Ziel die Arbeitsabläufe (Ausleihen, Katalogisieren) zu unterstützen und einen Datenaustausch zwischen den Bibliotheken/Verbünden zu ermöglichen. Zur Verwaltung der Daten wurden in der Bibliothekswelt seit Jahrzehnten erfolgreich Datenbanksysteme eingesetzt. Sie bieten den Vorteil, die erforderlichen Informationen schnell auffindbar zu machen und mehreren Benutzern gleichzeitig den Zugriff auf gleiche Daten zu gewährleisten. Ein Beispiel für ein solches System sind Online Public Access Catalogues (OPACs) [15].

Abschnitt 11.3.1 beschreibt die meistverbreiteten Metadatenstandards, Abschn. 11.3.2 stellt die für bibliographische Daten relevanten Linked-Data-Vokabulare und -Ontologien vor.

11.3.1 Metadatenstandards

Ein Metadatenstandard (auch Metadatenformat genannt) spezifiziert die Elemente, die in einem bestimmten Kontext verwendet werden sollen, sowie die Struktur und Bedeutung dieser Elemente. Außerdem sorgen Metadatenstandards für einen unproblematischen Datenaustausch zwischen Datenanbietern und gewährleisten eine anwendungsübergreifende Nutzbarkeit der Daten. Bezüglich seines Einsatzgebiets kann jeder Metadatenstandard einer der drei folgenden Gruppen zugeordnet werden:

(i) Ein **Erfassungsformat** wird zur Eingabe der Metadaten benutzt.
(ii) Ein **Internes Format** dient zur Speicherung und Verwaltung von Daten.
(iii) Ein **Austauschformat** dient zum Datenaustausch zwischen verschiedenen Datenanbietern.

Oft stimmt das Austauschformat mit dem intern verwendeten Format nicht überein. Das kann in einigen Fällen zu Informationsverlust führen, wenn die beiden Formate eine unterschiedliche Auszeichnungstiefe haben. Heterogenität von Informationsressourcen und einrichtungsspezifische Anforderungen an Metadaten haben zur Entstehung zahlreicher Standards geführt; jeder wurde für einen bestimmten Objekttyp konzipiert. So gibt es spezielle Standards für Archivdokumente, Videotheken oder Museen.

Zu den populärsten Metadatenstandards zählen MARC, DC, MAB, LIDO, EAD, MODS und METS. Die folgenden Unterabschnitte stellen jedes davon kurz vor; für einen tiefer gehenden Überblick siehe [14]. Diese XML-basierten Metadatenstandards repräsentieren die bibliographischen Daten in einer strukturierten maschinenlesbaren Form –

ein wichtiger Schritt in Richtung Linked Data. XML-Schemata können zwar komplexe Sachverhalte darstellen, aber es bedarf zusätzlicher Spezifikations- und Implementationsarbeit, um Verknüpfungen auf externe Ressourcen in XML darzustellen. Ebenso ist es nicht ohne Übersetzungsaufwand und Informationsverlust möglich, mehrere Metadatensätze über dieselbe Ressource, die in unterschiedlichen XML-Schemata (hier konkret: unterschiedlichen XML-basierten Metadatenstandards) vorliegen, zu einem Metadatensatz zusammenzuführen. Verknüpfung mit externen Ressourcen ist in RDF leicht; ebenso können mehrere Metadatensätze, die dieselbe Ressource mittels unterschiedlicher Vokabulare beschreiben, leicht zu *einem* Datensatz vereinigt werden (siehe Abschn. 11.2.2). Die Frage, wie ein RDF-basierter Metadatenstandard konkret aussehen soll, wird ausführlich im Abschn. 11.4.1 diskutiert.

MARC (Machine-Readable Cataloguing) MARC ist ein Standard für die Repräsentation und den Austausch bibliographischer Daten in maschinenlesbarer Form[7] [6]. Er wurde in den 1960er Jahren von der US-amerikanischen Library of Congress[8] entwickelt und wird seit 1968 als universelles Austauschformat für bibliothekarische Daten weltweit eingesetzt. Es gibt mehrere Versionen dieses Formats; aktuell sind UNIMARC und MARC 21 von Bedeutung.

MAB (Maschinelles Austauschformat für Bibliotheken) Ein vergleichbares, aber ausschließlich im deutschsprachigen Raum verwendetes Austauschformat ist MAB (Maschinelles Austauschformat für Bibliotheken[9]). Seine Entwicklung wurde 1973 von der Deutschen Nationalbibliothek zusammen mit der Arbeitsstelle für Bibliothekstechnik initiiert und hatte zum Ziel, ein nationales Austauschformat zu erstellen. Er besteht aus fünf Teilformaten, die verschiedene Arten von Daten beschreiben: bibliographische Daten, Personennamen, Schlagwörter, Lokaldaten und Körperschaftsnamen. Alle Weiterentwicklungen des MAB wurden 2006 eingestellt, da dieses Format durch MARC 21 abgelöst wird. Im Gegensatz zu MARC erlaubt MAB eine feinere Granularität bei der Auszeichnung bibliographischer Elemente.

DC (Dublin Core) Das internationale, fachübergreifende Format DC (Dublin Core[10]) ist auf dem ersten Dublin Core Metadata Workshop im März 1995 in Dublin, Ohio (USA) entstanden und wird durch die Dublin Core Metadata Initiative (DCMI) gepflegt und weiterentwickelt. Der größte Vorteil dieses Formats liegt auf der Beschränkung auf 15 universell einsetzbaren und weitgehend allgemein verständlichen Kernelementen (core

[7] MARC Standards. Library of Congress, 20. Dez. 2013. http://www.loc.gov/marc/, aufgerufen am 21.02.2014.
[8] Library of Congress. http://www.loc.gov/, aufgerufen am 21.02.2014.
[9] MAB. Deutsche Nationalbibliothek, 14. Okt. 2013. http://www.dnb.de/EN/Standardisierung/Formate/MAB/mab_node.html, aufgerufen am 21.02.2014.
[10] Dublin Core Metadata Initiative. http://www.dublincore.org, aufgerufen am 21.02.2014.

elements) [13]; mit den „DCMI Metadata Terms“ [11] gibt es jedoch auch ein erweitertes Vokabular. Dublin Core wird nicht nur in bibliothekarischen Einrichtungen, wie Museen und Archiven, sondern auch in Behörden und in der Wirtschaft eingesetzt.

LIDO (Lightweight Information Describing Objects) LIDO [7] wird meistens zur Beschreibung der Objekte aus dem musealen Bereich eingesetzt. Er wurde entwickelt um Informationen über Sammlungsobjekte in Internet-Portalen zur Verfügung zu stellen. Im Gegensatz zu den anderen Metadatenformaten ermöglicht LIDO den Datenlieferanten selbst zu entscheiden, welche Metadaten an die Portale weitergegeben werden und welche ihnen vorenthalten werden sollen. Alle LIDO Elemente können in zwei Hauptgruppen aufgeteilt werden: deskriptive Metadaten und administrative Metadaten. Der entscheidende Vorteil von LIDO gegenüber Dublin Core besteht im Detaillierungsgrad der Beschreibung. Dieser erweist sich jedoch als Nachteil, wenn man LIDO-Metadaten auf Dublin Core abbildet, weil dies zu Informationsverlust führen kann.

METS (Metadata Encoding & Transmission Standard) METS[11] ist ein XML-Format, das durch die Library of Congress verwaltet wird. Im Gegensatz zu den anderen Formaten ermöglicht METS eine Beschreibung von digitalen Sammlungen oder strukturierten Werken, wie z. B. Buchreihen oder Büchern mit mehreren Kapiteln. Es ist ein Containerformat, das Metadaten unterschiedlicher Formate wie MODS (Metadata Object Description Schema), MAB (Maschinelles Austauschformat für Bibliotheken), MARC (Machine-Readable Catalogue) oder Dublin Core aufnehmen kann. METS enthält Elemente, um komplexe Objekte zu gruppieren und sie mit deskriptiven und administrativen Metadaten zu verbinden.

EAD (Encoded Archival Description) Das Format EAD ist ein XML-Kodierungsstandard für Archivalien. Das Format wurde in den 1990er Jahren vom Technical Subcommittee for Encoded Archival Description der Society of American Archivists in Kooperation mit der Library of Congress entwickelt. Die Richtlinien für die Implementierung des EAD-Formats wurden durch die Research Libraries Group (RLG) in einem Satz von ‘Best Practices’ veröffentlicht.[12] Das Dokument definiert obligatorische, empfohlene und optionale Elemente und Attribute. Wenige Jahre später wurde EAD auch in Deutschland eingesetzt. Mit der Übernahme des EAD-Formats ergab sich für deutsche Archivare die Möglichkeit, Informationen über Findmittel detailliert zu erfassen.

[11] Metadata Encoding and Transmission Standard (METS). Library of Congress, 4. Feb. 2014. http://www.loc.gov/standards/mets/, aufgerufen am 21.02.2014.

[12] RLG Best Practice Guidelines for Encoded Archival Description. RLG EAD Advisory Group. 2002. http://www.oclc.org/content/dam/research/activities/ead/bpg.pdf?urlm=161431, aufgerufen am 21.02.2014.

MODS (Metadata Object Description Schema) MODS[13] ist ein XML-basiertes Format zur Beschreibung bibliographischer Informationsressourcen, das von der Library of Congress entwickelt wurde. Im Juni 2002 wurde MODS für einen Probelauf freigegeben; im Juli 2013 ist die Version 3.5 erschienen. MODS wurde als Kompromiss zwischen der Komplexität des MARC-Formats und der extremen Einfachheit von Dublin Core konzipiert. MODS übernimmt die wichtigen Elemente des MARC-Formats, verlangt aber nicht die Definition aller MARC-Felder und verwendet nicht das Feld- und Unterfeld-Tagging des MARC-Standards. Einige Datenelemente sind somit nicht mehr mit MARC kompatibel; deshalb ist die Übersetzung von MARC nach MODS und umgekehrt nicht verlustfrei.

11.3.2 Vokabulare und Ontologien

Vokabulare definieren Konzepte und Beziehungen zur Beschreibung von Dingen in einem speziellen Fachgebiet, oder auch ganz allgemeiner Dinge. Ist die Semantik der Konzepte und Beziehungen mittels formaler Logik beschrieben, so spricht man auch von einer Ontologie. Mit der Entwicklung des Semantic Web ist eine Vielzahl maschinenlesbarer Vokabulare und Ontologien entstanden. Die Möglichkeit, Vokabulare bzw. Ontologien in RDF zu repräsentieren, wurde in Abschn. 11.2.2 erklärt. Wir stellen hier einige speziell für bibliothekarische Zwecke entwickelte Vokabulare und Ontologien vor; weitere sind in [18] beschrieben.

Seit 2002 steht der oben genannte **Dublin-Core**-Metadatenstandard (DC) als RDF-Vokabular zur Verfügung [13, 11]. Das DC-Vokabular kommt nicht nur in Datensätzen aus dem kulturellen Bereich, sondern fast überall in Linked Datasets zum Einsatz.

Das **CIDOC Conceptual Reference Model (CRM)** legt eine formale Struktur für die Beschreibung impliziter und expliziter Konzepte und Beziehungen für die Dokumente aus dem im Bereich des kulturellen Erbes fest.[14] Das CIDOC CRM ist das Ergebnis der Zusammenarbeit des Internationalen Ausschusses für Dokumentation (CIDOC) und der Internationalen Museumsrates ICOM. Im Jahr 2006 wurde das CIDOC CRM als Norm für den kontrollierten Austausch von Informationen im Bereich des kulturellen Erbes festgelegt. Die CRM-Version 5.1.2 umfasst 86 Klassen und 137 Eigenschaften. CIDOC CRM ist offiziell auf Papier spezifiziert, jedoch auch als OWL-Ontologie implementiert worden.[15]

[13] Metadata Object Description Schema. The Library of Congress. 2013. http://www.loc.gov/standards/mods/mods-3-5-announcement.html, aufgerufen am 21.02.2014.
[14] CIDOC CRM. http://www.cidoc-crm.org, aufgerufen am 21.02.2014.
[15] Erlangen CRM / OWL – An OWL DL 1.0 implementation of the CIDOC Conceptual Reference Model (CIDOC CRM). University of Erlangen-Nuremberg. http://erlangen-crm.org/, aufgerufen am 21.02.2014.

Die **Bibliographic Ontology (BIBO [10])** ist eine OWL-Ontologie zum Modellieren bibliographischer Daten als RDF. Diese Ontologie kann sowohl als Ontologie für Klassifizierung von Dokumenten als auch als Vokabular zur Beschreibung von beliebigen Dokumenttypen benutzt werden. ihr Aufbau berücksichtigt bestehende bibliographische Metadatenstandards.

Das **Simple Knowledge Organization System (SKOS**[16]**)** ist eine OWL-Ontologie[17], mit der man kontrollierte Vokabulare wie Thesauri und bibliothekarische Klassifikationen beschreiben kann.

11.4 Herausforderungen beim Publizieren von Daten Digitaler Bibliotheken als Linked Data

Bibliographische Daten zeichnen sich aus durch die Heterogenität der zu beschreibenden Ressourcen (Bücher, Bilder, Fotos, Tonaufnahmen, Videos, Museumsexponate, Archivalien) und durch die Vielzahl der vorhandenen Metadatenstandards, die sich in der Beschreibungstiefe stark unterscheiden. Oft werden Metadatenstandards noch zusätzlich an einrichtungsspezifische Anforderungen angepasst. Aus diesen Gründen ist die Veröffentlichung bibliographischer Daten als Linked Data eine Herausforderung. Die folgenden Abschnitte beschreiben anhand konkreter Projekte die wesentlichen Probleme bei der Datenmodellierung, beim Datenmapping und der Datenverknüpfung und zeigen Lösungsansätze auf.

11.4.1 Datenmodellierung. Anwendungsfall Europeana

Die traditionellen, in Abschn. 11.3.1 vorgestellten Metadatenstandards sind ursprünglich für Katalogisierungszwecke entwickelt worden und ähneln in ihrer Struktur Karteikarten. Bei diesem Ansatz werden die Metadaten einer Informationsressource durch einen festen Satz von Feldern mit genau definierten Werten dargestellt. Jedoch setzt die Linked-Data-Repräsentation das graphbasierte RDF-Datenmodell voraus (siehe Abschn. 11.2.2). Der Unterschied zwischen den beiden Modellierungsansätzen macht die Konvertierung bibliographischer Daten nach RDF zu einer komplexen Aufgabe. Das Problem besteht darin, die Vielfalt der existierenden Metadatenstandards ohne Informationsverlust in ein einheitliches Datenmodell zu übersetzen.

[16] SKOS Simple Knowledge Organization System. W3C. 13. Dez. 2012. http://www.w3.org/2004/02/skos/, aufgerufen am 21.02.2014.

[17] Die Semantik von SKOS geht teilweise über die Ausdrucksstärke von OWL hinaus.

Die *W3C Library Linked Data Incubator Group*[18] hat zwei Lösungsansätze vorgeschlagen, die sowohl höhere Interoperabilität als auch die bessere Verknüpfung von Bibliotheksdaten mit externen Datenquellen ermöglichen sollen [4]:

1. Verwendung bereits vorhandener Linked-Data-Vokabulare
2. Explizite Definition der Relation zwischen Vokabularen der Bibliothekswelt und Vokabularen in anderen Bereichen

Europeana[19], eines der größten und bedeutendsten Projekte im Bereich Linked Open Data für Kultureinrichtungen, verfolgt den ersten Lösungsansatz für die Datenmodellierung. Im Jahr 2005 hat die EU im Rahmen der Initiative „i2010“[20] den Aufbau einer spartenübergreifenden Europäischen Digitalen Bibliothek beschlossen. Ende 2008 ist ein entsprechendes Portal unter dem Namen Europeana in Betrieb gegangen, welches die Aufgabe einer virtuellen europäischen Bibliothek erfüllt. Die Europeana dient als Zugangsportal; sie bietet keine eigenen Inhalte an. Sie vernetzt die nationalen Portale (in Deutschland die in Abschn. 11.4.2 genauer beschriebene Deutsche Digitale Bibliothek) und schafft einen anwendungsfreundlichen übergreifenden Zugang zu kulturellen und wissenschaftlichen Inhalten in der ganzen EU. Die Entscheidung über Art und Umfang der über die Europeana zugänglichen digitalen Inhalte liegt damit im Wesentlichen auf nationaler Ebene. Im November 2013 erreichte die Anzahl der digitalen Objekte in der Europeana 30 Millionen, darunter Bilder, Texte, Tonaufnahmen und Videos aus mehr als 2.300 Institutionen aus ganz Europa. Unter den Beteiligten sind sowohl international bekannte Namen wie das Rijksmuseum in Amsterdam, die British Library oder der Louvre, als auch lokale Galerien und regionale Ton- und Bildarchive.

Das zentrale Prinzip des Europeana-Projektes ist, dem Nutzer ein Netzwerk semantischer Ressourcen als primäre Ebene für seine Interaktionen zur Verfügung zu stellen. Die Europeana ist mit Semantic-Web-Technologien verwirklicht worden; die Daten sind als Linked Data zugänglich gemacht worden [12]. Ursprünglich war der Metadatenstandard Europeana Semantic Elements (ESE) als Datenmodell für Europeana entwickelt worden. Jedoch hat sich dieses Format für Linked Data als ungeeignet herausgestellt, da es nicht die Möglichkeit bietet, die Original-Metadaten ohne Informationsverlust zu übertragen. Diese Nachteile wurden durch ein neu entwickeltes Europeana Data Model (EDM) be-

[18] W3C Library Linked Data Incubator Group. W3C, 16. Feb. 2012. http://www.w3.org/2005/Incubator/lld/, aufgerufen am 21.02.2014.
[19] Europeana. http://www.europeana.eu/, aufgerufen am 21.02.2014.
[20] i2010: Information Society and the media working towards growth and jobs. http://europa.eu/legislation_summaries/information_society/strategies/c11328_en.htm, aufgerufen am 21.02.2014.

hoben. EDM vereint sowohl eigene Klassen und Eigenschaften als auch Elemente der bewährten, in Abschn. 11.3.2 vorgestellten Vokabulare Dublin Core und SKOS.

Die wichtigsten Entwurfsprinzipien des Europeana-Datenmodells sind:[21]

- Unterscheidung zwischen einer realen Ressource (Bild, Buch) und seiner digitalen Repräsentation
- Unterscheidung zwischen einer Ressource und dem Metadatensatz, der diese Ressource beschreibt
- Mehrere Sichten auf eine Ressource, die einander widersprechende Informationen enthalten können
- Unterstützung von Ressourcen, die aus mehreren anderen Ressourcen zusammengesetzt sind
- Standard-Metadatenformat mit Spezialisierungsoption
- Maximale Wiederverwendung vorhandener Standards.

11.4.2 Metadaten-Mapping. Anwendungsfall Deutsche Digitale Bibliothek (DDB)

Bei der Integration von Ressourcen aus heterogenen Quellen werden Metadaten aus dem Originalformat in ein Austauschformat überführt. Dieser Transformationsprozess wird als Mapping bezeichnet. Das Ziel beim Mapping ist semantische Strukturen ohne Fehlinterpretation und Informationsverlust zu übertragen. Jedoch führt die Heterogenität von Informationsressourcen dazu, dass deren Metadaten in beinahe unzähligen Formaten kodiert sind. Die Frage, wie diese Vielfalt von Datenformaten automatisch in ein einheitliches Format transformiert werden kann, hat sich als eine der größten Herausforderungen bei der Deutschen Digitalen Bibliothek herausgestellt.

Das Projekt Deutsche Digitale Bibliothek (DDB[22]) wurde als deutscher Beitrag zum Europeana-Netzwerk im Jahr 2009 ins Leben gerufen. Das Ziel des Projektes wurde wie folgt definiert [5]:

> Ziel der Deutschen Digitalen Bibliothek (DDB) ist es, jedermann über das Internet freien Zugang zum kulturellen und wissenschaftlichen Erbe Deutschlands zu eröffnen, also zu Millionen von Büchern, Archivalien, Bildern, Skulpturen, Musikstücken und anderen Tondokumenten, Filmen und Noten. Als zentrales nationales Portal soll die DDB perspektivisch die digitalen Angebote aller deutschen Kultur- und Wissenschaftseinrichtungen miteinander vernetzen. Mit der DDB soll Deutschland seine Anschluss- und Wettbewerbsfähigkeit in Wissenschaft, Forschung und Bildung sichern, aber auch sein einzigartiges kulturelles

[21] Europeana Professional – Tech Details. http://pro.europeana.eu/web/guest/tech-details, aufgerufen am 21.02.2014.

[22] Deutsche Digitale Bibliothek. https://www.deutsche-digitale-bibliothek.de/, aufgerufen am 21.02.2014.

> Erbe und Wissen für jedermann komfortabel über einen zentralen Anlaufpunkt zugänglich machen. Durch die zentrale Zugänglichkeit, also indem an jedem PC-Arbeitsplatz mit Internetanschluss unabhängig von Ort und Zeit Zugang zur gesamten erforderlichen Information geschaffen wird, werden die Recherchemöglichkeiten in Forschung, Lehre und Wirtschaft grundlegend verbessert.

Das DDB-Projekt ist von hoher inhaltlicher, technischer, rechtlicher und organisatorischer Komplexität. Um die einfache Integration der DDB-Ressourcen im Europeana-Netzwerk zu gewährleisten, wurde zur Datenrepräsentation das in Abschn. 11.4.1 eingeführte Europeana-Datenmodell gewählt. Einerseits soll die Wiederverwendung von Metadatenstandards für Interoperabilität der Daten sorgen und unproblematische Datentransformationen ermöglichen. Andererseits wurden viele Standardformate an die spezifischen Anforderungen des Datenanbieters angepasst. So wurden neue Konzepte hinzugefügt oder deren Bedeutung wurde geändert. Besonders oft tritt dieses Problem bei Museumsdaten auf. Das liegt unter anderem daran, dass museale Ressourcen sehr heterogen sind. So haben Dublin-Core-Metadaten oft unterschiedliche Bedeutungen, wenn sie von unterschiedlichen Datenanbietern kommen. Bei einem Bild beschreibt z. B. das Element *dc:subject* den „Gegenstand des Bildes“; Bei einem Textobjekt dagegen bezeichnet es das „Thema des Textes“ [1]. Das Beispiel veranschaulicht, dass ein Mapping rein auf der Grundlage der schwachen Semantik von Dublin Core zu fehlerhaften Ergebnissen führen kann.

Zur Lösung dieses Problems wurde in der DDB eine Mapping-Bibliothek entwickelt, die aus mehreren XSLT-Transformationen (d. h. Übersetzungen zwischen verschiedenen XML-Formaten) besteht. Für jedes einzelne Format werden Vorschriften für eine Abbildung der Metadaten auf das DDB-Modell definiert, das so genannte konzeptuelle Mapping. In Einzelfällen sind für ein Format auch anbieterspezifische Mappings vorgesehen. Einen wichtigen Teil der Library bilden Templates, welche die Abbildung der EDM-Ressourcen (siehe Abschn. 11.4.1) in RDF festlegen. Diese Templates sorgen für die einheitliche Repräsentation der Ressourcen, unabhängig von Provider und Format.

Der DDB-Lösungsansatz lässt sich auf vergleichbare Aufgabenstellungen der Datenaufbereitung in anderen Domänen übertragen.

11.4.3 Verknüpfung

Das RDF-Datenmodell stellt eine Ressource durch die Menge ihrer Eigenschaften und ihrer Verknüpfungen mit anderen Ressourcen dar. Abbildung 11.1 zeigt beispielsweise, wie eine Reproduktion des Bildes ‚La Joconde‘ von Leonardo da Vinci und eine Zeichnung eines Hubschraubers jeweils durch einen Satz von Metadaten repräsentiert sind. So beschreibt der Metadatensatz des Bild-Objektes die Eigenschaften *‚Creator‘*, *‚Creation Date‘*, *‚Type‘* und *‚Data Provider‘*. In beiden Metadatensätzen kommt eine Person vor, der Schöpfer des Werks (*‚Creator‘*). Ist das Objekt dieses Metadatenfelds als Literal repräsentiert, d. h. als einfacher Datenwert wie z. B. ein Text-String (*„Leonardo da Vinci“*)

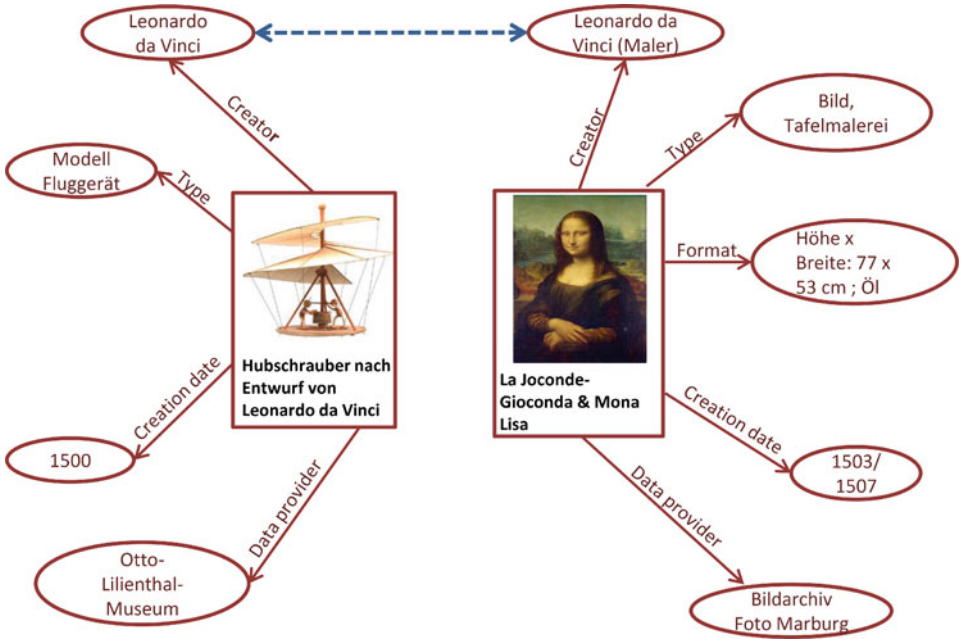


Abb. 11.1 Repräsentation von Metadatensätzen in RDF

oder eine Zahl, gibt es keine Möglichkeit, mit Sicherheit festzustellen, ob der Schöpfer der beiden Werke im Beispiel ein und dieselbe Person ist – selbst dann nicht, wenn das Literal in beiden Fällen den gleichen Wert hätte. Ist die Person als Ressource modelliert, d. h. mit einem URI bezeichnet, muss man im Allgemeinen davon ausgehen, dass unterschiedliche Datenbestände unterschiedliche URIs verwenden, also z. B. http://www.lilienthal-museum.de/data/persons/Leonardo_da_Vinci, bzw. http://www.fotomarburg.de/archive/creators/Leonardo_da_Vinci_(Maler).[23] Dann ist nicht ohne Weiteres klar, dass der Schöpfer der beiden Werke im Beispiel ein und dieselbe Person ist. Im RDF-Datenmodell ist es jedoch möglich, eine Verknüpfung herzustellen, die besagt, dass die beiden Ressourcen identisch sind, d. h. dass jede Eigenschaft der einen Ressource auch auf die andere Ressource zutrifft und umgekehrt. Dies geschieht durch Verwendung der Eigenschaft *owl:sameAs*, hier als blau gestrichelter Pfeil dargestellt. Abschnitt 11.5 erklärt, wie solche Verknüpfungen automatisch erkannt werden können. In jedem Fall helfen Verknüpfungen dabei, neues Wissen zu entdecken: Der Entwurf des Hubschraubers stammt vom Maler Leonardo da Vinci, dem Autor von ‚La Joconde'; der berühmte Maler war auch Ingenieur.

Vor dem Hintergrund dieser vielseitigen Sichten auf – in diesem Fall – eine Person wird klar, warum die Verknüpfung bibliothekarischer Datenbestände sinnvoll ist. Es sind unterschiedliche Szenarien für die Verknüpfung bibliographischer Daten denkbar:

[23] Diese URIs sind für dieses Beispiel frei erfunden.

1. **Verknüpfung mit Normdateien und Klassifikationen** kann als Vorbereitung für weitere Verknüpfungsszenarien angesehen werden. Unter anderem erleichtert sie die Verknüpfung der Daten innerhalb *eines* Datenbestands, wenn dasselbe Ding innerhalb einer Einrichtung mit unterschiedlichen URIs bezeichnet wird. Diese Aufgabe kann am einfachsten durch die primäre Verknüpfung mit einer Normdatei realisiert werden.
2. Bei der **Datenintegration** kommt es oft vor, dass z. B. zwei Datenanbieter Metadaten über dieselbe Ressource liefern, diese aber unterschiedlich bezeichnen. Dies führt zu Redundanzen in den Daten. Es ist jedoch nicht einfach, identische Ressourcen zu erkennen und somit solche Redundanzen aufzulösen.
3. **Verknüpfung mit externen Datenquellen** – Während die ersten beiden Verknüpfungsarten dafür sorgen, die Qualität der Daten durch Anreicherung und Vermeidung von Redundanzen zu verbessern, ermöglicht dieses Szenario die Daten aus unterschiedlichen Perspektiven zu beleuchten.

Die folgenden Unterabschnitte erklären die einzelnen Verknüpfungsszenarien ausführlicher. Abschnitt 11.5 behandelt technische Aspekte der Verknüpfung an einem konkreten Beispiel: Verknüpfung der Personen in der DDB mit der Gemeinsamen Normdatei GND.

Verknüpfung mit Normdateien Kulturelle Einrichtungen besitzen große Kompetenz und langjährige Erfahrung in der Erstellung und Verwaltung von Normdateien. Eine Normdatei ist eine Art kontrolliertes Vokabular, das für jedes einzelne Objekt einen normierten Namen zusammen mit abweichenden Namensformen beinhaltet. Wenn ein Datenbestand mit Normdateien verknüpft ist, bedeutet das, dass alle Namensvarianten (auch in mehreren Sprachen) für die Suche und Identifikation verfügbar sind. Somit eröffnen sich für die Nutzer neue Recherchemöglichkeiten. Neben unterschiedlichen Namensformen enthalten Normdateien weitere beschreibende Attribute. Oft sind z. B. für eine Person Informationen über die Lebensdaten, geographische Bezüge (Geburtsort, Sterbeort, Wirkungsort), ausgeübte Berufe und/oder die Verwandtschaftsbeziehungen zu anderen Personen (z. B. „Sohn von“) vorhanden. Je mehr unterschiedliche Informationen zu einem Ding vorhanden sind, desto besser kann man es verknüpfen. Da Normdateien oft für ein Gebiet zentral gepflegt werden und viele Informationen von hoher Qualität enthalten, bietet es sich an, Datenbestände zuerst mit einer Normdatei zu verknüpfen.

Die im deutschen Sprachraum bekannteste und meistverwendete **Gemeinsame Normdatei** (GND[24]) wurde von der Deutschen Nationalbibliothek entwickelt und umfasst Personen, Körperschaften, Kongresse, Geografika, Sachschlagwörter und Werktitel. Sie ist aus vier getrennten Normdateien entstanden: der Gemeinsamen Körperschaftsdatei (GKD), der Personennamendatei (PND), der Schlagwortnormdatei (SWD) sowie der Einheitssachtitel-Datei des Deutschen Musikarchivs (DMA-EST-Datei). Seit 2010 steht die

[24] Gemeinsame Normdatei (GND). Deutsche Nationalbibliothek, http://www.dnb.de/gnd, aufgerufen am 21.02.2104.

GND als Linked Dataset zu Verfügung und bietet somit eine hervorragende Grundlage für die Datenverknüpfung.

Das **Virtual International Authority File** (VIAF[25]) ist im Gegensatz zur GND eine internationale Normdatei und hat das Ziel, die nationalen Normdateien zu verknüpfen. In der Datei sind die Datenbestände von 25 Normdateien zusammengeführt; unter den Beteiligten sind z. B. die Königliche Bibliothek zu Stockholm, die Nationalbibliotheken von Israel und Australien, die Königliche Bibliothek der Niederlande und viele andere.

Verknüpfung mit Klassifikationen Im Abschn. 11.2.2 wurde als Vorteil des RDF-Datenmodells erwähnt, dass Linked-Data-konform veröffentlichte Vokabulare sich selbst dokumentieren können. In Digitalen Bibliotheken wird dieser Vorteil nicht nur für Vokabulare genutzt, sondern auch für Klassifikationen.

Eine Klasse fasst inhaltlich verwandte Bibliotheksressourcen zusammen, z. B. „alle Werke zur Geschichte der Bundesrepublik Deutschland von 1949 bis 1990“. In der Allgemeinen Systematik für Öffentliche Bibliotheken (ASB[26]), einer vor allem in öffentlichen Bibliotheken zur inhaltlichen Erschließung verwendeten Klassifikation, hat diese Klasse den alphanumerischen Bezeichner *Emp 820*. Es gibt eine Hierarchie übergeordneter Klassen; so steht z. B. die Oberklasse *E* für „Geschichte, Zeitgeschichte“. Die meisten Klassifikationen sind monohierarchisch aufgebaut und verwenden alphanumerische Klassenbezeichner. In internationalen, insbesondere englischsprachigen wissenschaftlichen Bibliotheken, ist die ursprünglich 1876 von Melvil Dewey entwickelte Dewey-Dezimalklassifikation (DDC) am weitesten verbreitet. Ihre deutsche Übersetzung ist 2002–2005 von der Deutschen Nationalbibliothek in Zusammenarbeit mit der Fachhochschule Köln entwickelt worden. Zu dem oben genannten Themengebiet Geschichte passt die DDC-Klasse 90 „Geschichte“; darunter steht z. B. die Klasse 907.2 für „Historische Forschung“.

In einem Linked Dataset könnte ein Buch mithilfe von Dublin Core so klassifiziert werden:[27]

```
<http://d-nb.info/1043855998>
  dcterms:title "Kleine Theorie des Archivs";
  dcterms:creator SSchenk, Dietmar";
  dcterms:issued 2014 ;
  dcterms:subject "907.2".
```

Damit wäre jedoch weder klar, dass `907.2` eine DDC-Klasse ist, noch welches Themengebiet diese Klasse umfasst. Die DDC selbst ist jedoch als Linked Dataset ver-

[25] VIAF. OCLC. http://viaf.org/, aufgerufen am 21.02.2014.
[26] ASB-KAB. http://asb-kab-online.ekz.de/, aufgerufen am 21.02.2014.
[27] Die hier gezeigten Daten sind teilweise dem DNB-Datensatz unter http://d-nb.info/1043855998 entnommen (zuletzt abgerufen am 21. Februar 2014), aber vereinfacht dargestellt.

öffentlicht worden, so dass man in dem obigen Beispiel auch `dcterms:subject <http://dewey.info/class/907.2/>` hätte schreiben können. Zu dieser Klasse ist wiederum unter diesem URI, wenn man ihn als URL auffasst und dereferenziert, eine Dokumentation für Menschen (in HTML) und Maschinen (RDF) verfügbar. Zur Modellierung der DDC in RDF wurde das in Abschn. 11.3.2 genannte SKOS verwendet [20]. Weitere Klassifikationen, die in SKOS als Linked Open Datasets vorliegen, sind die Library of Congress Subject Headings (LCSH [24]) und die Mathematical Subject Classification (MSC [19]), deren SKOS-Implementation der zweite Autor dieses Kapitels technisch koordiniert hat.

Verknüpfung mit externen Daten Als weitere Ziele von Verknüpfungen bieten sich die folgenden Linked Open Datasets an:

- **DBpedia**[28] ist ein Linked Open Dataset mit strukturierten Informationen aus Wikipedia; DBpedia in der Version 3.9 von September 2013 enthält unter anderem Informationen über 832.000 Personen und 639.000 Orte. DBpedia enthält Daten in vielen Sprachen, denn es wurden u. a. die englische, deutsche, französische, spanische, italienische, portugiesische, polnische, schwedische, niederländische, japanische, chinesische, russische, finnische und norwegische Wikipedia verwendet. Die Daten sind als RDFDump verfügbar; alternativ können die Daten über einen SPARQL-Endpoint abgefragt werden.
- **GeoNames**[29] ist eine Datenbank für geographischen Daten, die aus verschiedenen Datenquellen integriert wurden. GeoNames enthält Informationen über Länder, Regionen und Orte; neben geographischen Koordinaten sind Fakten zur Bevölkerung, Verwaltungsgliederung, Fläche usw. vorhanden. Die Datenbank umfasst über 10 Millionen Ortsnamen samt Alternativnamen in verschiedenen Sprachen; viele davon sind mit den entsprechenden DBpedia-Ressourcen verlinkt.

11.5 Automatische Verknüpfung der DDB-Personen-Ressourcen mit der Gemeinsamen Normdatei

Aktuell, im April 2014, umfasst die DDB ca. 7,8 Mio. Objekte von insgesamt über 2000 Kultureinrichtungen und Institutionen. Die wichtigsten Voraussetzungen für die Verknüpfung der Personen-Ressourcen der DDB mit einer Normdatei sind erfüllt – die Ressourcen in DDB sind zuverlässig dauerhaft identifiziert und liegen im RDF-basierten EDM-Datenmodell der DDB vor.

[28] DBpedia. http://dbpedia.org, aufgerufen am 21.02.2014.
[29] GeoNames. http://www.geonames.org, aufgerufen am 21.02.2014.

Der Autor eines Buches, der Fotograf, der ein Bild von einer bekannten Person gemacht hat, und sogar die Person auf diesem Bild sind nicht mehr nur als Literal-Objekte in einem Metadatenfeld repräsentiert, sondern selbst als Ressourcen. Durch die Verknüpfung mit der Gemeinsamen Normdatei (GND) sollen solche Personen-Ressourcen *eindeutig* identifiziert werden. In dem in Abb. 11.1 gezeigten Beispiel stünde dann in beiden Objekten statt des Literals *„Leonardo da Vinci“* der URI der Personenseite in GND (http://d-nb.info/gnd/118640445), oder alternativ gäbe es einen *owl:sameAs*-Link zwischen http://d-nb.info/gnd/118640445 und http://www.lilienthal-museum.de/data/persons/Leonardo_da_Vinci sowie http://www.fotomarburg.de/archive/creators/Leonardo_da_Vinci_(Maler).

Zur automatischen Verknüpfung der DDB mit der GND ist im Rahmen des DDB-Projekts eine Machbarkeitsstudie durchgeführt worden. Ziel dieser Verknüpfung ist, automatisch festzustellen, ob eine Personen-Ressource in der DDB und eine Personen-Ressource in der GND dieselbe reale Person beschreiben. Das RDF-Datenmodell repräsentiert Ressourcen durch die Menge ihrer Eigenschaften; so ist eine Person durch die von ihr geschaffenen Werke (und deren Eigenschaften) sowie durch Namen, Lebensdaten, Orte usw. beschrieben. Man kann deshalb davon ausgehen: Je größer die Ähnlichkeit der Eigenschaften zweier Ressourcen ist, desto höher ist die Wahrscheinlichkeit, dass die Ressourcen äquivalent sind. Somit lässt sich das Verknüpfungsproblem auf die Ähnlichkeitsberechnung reduzieren. Die Ähnlichkeit zweier Ressourcen kann man mit Hilfe von Ähnlichkeitsmaßen oder Distanzmaßen ausdrücken. Ein Ähnlichkeitsmaß ist eine Funktion, die den Grad der Übereinstimmung zwischen den zu vergleichenden Objekten durch einen reellen Wert widerspiegelt. Je näher sich der Ähnlichkeitswert an 1 befindet, desto ähnlicher sind zwei Objekte. Die wichtigsten Schritte bei der Bestimmung der Ähnlichkeit sind:

1. Identifikation der für Ähnlichkeit relevanten Objekt-Eigenschaften (d. h. Properties aus RDF-Vokabularen)
2. Definition geeigneter Distanzmaße
3. Ermittlung des globalen Ähnlichkeitswertes.

Beim ersten Schritt ist es von entscheidender Bedeutung, möglichst viele zuverlässige Eigenschaften zu identifizieren. Die beiden zu verknüpfenden Datenquellen (DDB und GND) beschreiben Personen nach unterschiedlichen Eigenschaften und mit unterschiedlichem Informationsgehalt. In der DDB ist eine Personen hauptsächlich aus der Sicht der von ihr geschaffenen Werke dargestellt (siehe Abb. 11.2).

Zu den beschreibenden Eigenschaften gehören *„Personenname“*, *„Titel eines Werkes“*, *„Publikationsdatum/Erstellungsdatum“*, *„Publikationsort“* und *„Thema eines Werkes“*. Die GND dagegen bietet zusätzlich zur Werk-Sicht die Sicht auf die Person (siehe Abb. 11.3).

Zusammen decken diese beiden GND-Sichten einen Satz von Eigenschaften ab (unterschiedliche Namensvarianten; Lebensdaten; geographischer Bezug: Geburtsort, Sterbeort; Berufe(e), Beziehungen zu anderen Personen; Werke), die eine gute Grundlage für eine

Thematisches Verzeichniß über die Compositionen von I Antonio Vivaldi II Arcangelo Corelli III Giuseppe Tartini IV [leer]

Contributor:	Fuchs, Aloys
Publication type:	Monographie
Published:	1839
Extent:	39 Bl.
PURL:	http://resolver.staatsbibliothek-berlin.de /SBB00003A5100000000
Location:	Staatsbibliothek zu Berlin – Preußischer Kulturbesitz
Legal status:	Attribution - NonCommercial - ShareAlike

Abb. 11.2 Repräsentation eines Werkes in der DDB

zuverlässige Identifikation von Personen bilden. Die Ähnlichkeit zweier Personen wird bezüglich derjenigen Objekteigenschaften berechnet, die in beiden Datenquellen vorhanden sind. Tabelle 11.1 stellt eine Auswahl der Properties dar, die für die DDB-GND-Verknüpfung ausgewählt wurden. Um den Verknüpfungsprozess zu beschleunigen, sollten für den Vergleich irrelevante Eigenschaften, z. B. ‚Data Provider', nicht berücksichtigt werden.

Im zweiten Schritt wird für jede ausgewählte Eigenschaft ein geeignetes Distanzmaß definiert. So werden z. B. für den Vergleich der Namen speziell dafür entwickelte Distanzmaße verwendet (z. B. die Monge-Elkan- oder Jaro-Winkler-Distanzen [8]). Titel werden z. B. mittels der für Texte geeigneten Kosinus-Distanz verglichen. Literale sollten vor dem

GND	
Link zu diesem Datensatz	http://d-nb.info/gnd/118693948
Person	Fuchs, Aloys
Geschlecht	männlich
Andere Namen	Fuchs, Alois Fuchs, Aloys Anton
Quelle	Riemann; LoC-NA; NDB OEML: http://www.musiklexikon.ac.at/ml/musik_F/Fuchs_Aloys.xml
Zeit	Lebensdaten: 1799-1853
Land	Österreich (XA-AT); Deutschland (XA-DE)
Geografischer Bezug	Geburtsort: Raase, Schlesien Sterbeort: Wien
Beruf(e)	Musikwissenschaftler Musikschriftsteller Musiker
Weitere Angaben	Österr. Beamter; Bassist; Sänger; Musikautographensammler
Beziehungen zu Organisationen	Kaiserlich-Königliche Hofmusikkapelle <Wien> (Bassist)
Systematik	14.4p Personen zu Musik
Typ	Person (piz)
Autor von	1 Publikation 1. *Thematisches Verzeichnis der sämtlichen Kompositionen von Joseph Haydn Fuchs, Aloys. - Wilhelmshaven : Heinrichshofen, 1968, Faks.-Nachdr. Hrsg.*

Abb. 11.3 Repräsentation einer Person in der GND

Tab. 11.1 Vergleich der RDF-Properties in der DDB und der GND

Aspekt	DDB-Property	GND-Property
Personen-Klasse	edm:Agent	gndo:DifferentiatedPerson
Personenname	skos:prefLabel	gndo:variantNameForThePerson gndo:preferredNameForThePerson gndo:variantNameEntityForThePerson gndo:forename gndo:prefix gndo:surname
Werktitel	dc:title dcterms:alternative	gnd:publication dc:title dcterms:alternative
Zeitangaben	dcterms:issued	gnd:dateOfBirth gnd:dateOfDeath gnd:periodOfActivity
Thema	dc:subject	gnd:professionOrOccupation gnd:preferredNameForTheSubjectHeading gnd:variantNameForTheSubjectHeading

Vergleich normiert werden: Sonderzeichen werden entfernt, Umlaute ersetzt und Strings in Kleinschreibung umgewandelt. Entsprechend der in der Tab. 11.1 ausgewählten Eigenschaften werden 4 Gruppen von Distanzmaßen definiert:

1. auf Namen (Monge-Elkan- und Jaro-Winkler-Distanzen).
2. auf Titeln (Jaccard-Distanz sowie die Kosinus-Distanz auf Trigrammen, die als q-Gramm-Distanz bezeichnet wird),
3. auf Zeitangaben (es wird überprüft, ob das Datum der Veröffentlichung von Werken in der bekannten Schaffensperiode einer Person liegt, oder alternativ im Intervall $[Geburtsjahr + 16; Sterbejahr]$).
4. Der Beruf einer Person wird mit dem Themengebiet eines Werks verglichen.

Danach werden die berechneten Einzeldistanzmaße zu einem globalen Wert kombiniert, mittels einer Funktion, die aus Basis-Aggregatfunktionen (z. B. Maximumbildung) und einem Gewichtsmodell besteht. Eine Basis-Aggregatfunktion legt fest, wie die einzelnen Distanzen innerhalb von einer Distanzgruppe miteinander kombiniert werden. So wird z. B. Durchschnitt über die Monge-Elkan- und Jaro-Winkler-Distanzen auf Namen-Properties gebildet. Das Gewichtsmodell definiert höhere Gewichte für die Distanzmaße auf Namen und Titel, während die beiden anderen Distanzmaße (auf Zeitangaben und Beruf) nur wenig zum globalen Ähnlichkeitswert beitragen. Solche komplexen Aggregatfunktionen, im Kontext der Verknüpfung von Datenbeständen auch Verknüpfungsregeln

genannt, lassen sich mithilfe von Verknüpfungs-Entdeckungs-Werkzeugen (link discovery tools) realisieren wie z. B. Silk[30] oder LIMES[31].

Die endgültige Entscheidung über die Ähnlichkeit der zu vergleichenden Personen-Ressourcen wird anhand des globalen Ähnlichkeitswertes getroffen. Liegt dieser oberhalb eines von einem Fachexperten definierten Schwellwerts, werden die beiden Personen für äquivalent befunden.

11.6 Ausblick auf zukünftige Entwicklungen

Dieses Kapitel hat sich hauptsächlich auf Verknüpfung des Datenbestands einer Digitalen Bibliothek mit anderen Datenbeständen konzentriert. Solche Verknüpfungen dienen nicht nur der Entdeckung neuen Wissens, sondern auch der inneren Organisation *einer* Digitalen Bibliothek. Mittels Linked Data können Verknüpfungen als Teil eines Datenbestands modelliert und einheitlich zugänglich gemacht werden. Die wichtigste Herausforderung angesichts der Größe Digitaler Bibliotheken besteht in der Automatisierung des Verknüpfungsprozesses. Wir haben speziell Normdateien als Verknüpfungsziel betrachtet, doch es bieten sich zahlreiche weitere Datenbestände dafür an, mit neuen Herausforderungen. Zum Beispiel geographische Informationen: Aktuelle Namen und Koordinaten sind inzwischen als Linked Open Data zugänglich, etwa in GeoNames (siehe Abschn. 11.4.3) oder den aus OpenStreetMap gewonnenen LinkedGeoData [3]. Die Erweiterung dieser Datenbestände um eine zeitliche Dimension zur Beschreibung historischer Daten – um etwa den in Abb. 11.3 genannten geographischen Bezug ‘Schlesien’ richtig zuzuordnen, steckt in den Anfängen[32]. Verknüpfungsregeln, die den räumlichen und zeitlichen Kontext berücksichtigen, werden entsprechend komplexer ausfallen.

Ein weiteres wichtiges Verknüpfungsziel sind Forschungsdaten. So sind in einer Digitalen Bibliothek wissenschaftlicher Veröffentlichungen auch die diesen Veröffentlichungen zu Grunde liegenden Forschungsdaten von Interesse. Forschungsförderer wie z. B. die EU im Horizon-2020-Programm verlangen zunehmend die freie Offenlegung von Forschungsdaten; auch hier bietet sich Linked Data unter dem Stichwort „Linked Research Data“ als Technologie an.

Als weitere zukünftige Herausforderungen beim Einsatz von Linked Data in Digitalen Bibliotheken benennen Geipel et al. die Wahl und Durchsetzung der richtigen Lizenzen für Linked Data, die Modellierung der Herkunft (Provenienz, englisch *Provenance*)

[30] Silk Link Discovery Framework. http://silk.wbsg.de/, aufgerufen am 17.04.2014.
[31] LIMES: LInk discovery framework for MEtric Spaces. http://aksw.org/Projects/LIMES.html, aufgerufen am 17.04.2014.
[32] Siehe z. B.: Historical OSM. https://wiki.openstreetmap.org/wiki/Historical_OSM, aufgerufen am 21.02.2014.

von Daten, etwa um ihre Vertrauenswürdigkeit zu bewerten, sowie die Entwicklung von Benutzerschnittstellen, die die Vorteile von Linked Data unmittelbar den Endanwendern zugänglich machen [16].

Abzusehen ist in jedem Fall, dass die nächste Generation von Bibliothekaren nicht nur Kenntnisse von Metadatenstandards haben sollte, sondern auch von den Grundzügen der Linked-Data-Technologie.

Literatur

1. Alieva, M. 2011. *Metadatenmapping im Projekt Deutsche Digitale Bibliothek anhand von Dublin Core, Lido und EAD.* (Magisterarbeit ed.). Universität zu Köln
2. Arms, W.Y. 2001. *Digital Libraries*. MIT Press
3. Auer, S., J. Lehmann, und S. Hellmann. 2009. LinkedGeoData – adding a spatial dimension to the web of data. In *Proc. of 8th International Semantic Web Conference (ISWC)*
4. Baker, T., E. Bermès, K. Coyle, G. Dunsire, A. Isaac, P. Murray, M. Panzer, J. Schneider, R. Singer, E. Summers, W. Waites, J. Young, und M. Zeng. Library linked data incubator group final report. Technical report, W3C
5. Bund-Länder-Fachgruppe DDB. Rahmenbedingungen zur anforderungsanalyse aus politischer, rechtlicher und funktionaler/technischer sicht. Technical report, Fraunhofer IAIS
6. Byrne, D.J. 1991. *MARC manual: understanding and using MARC records*. Libraries Unlimited
7. Coburn, E., R. Light, G. McKenna, R. Stein, und A. Vitzthum. Lido – lightweight information describing objects. Technical report, ICOM-CIDOC Working Group Data Harvesting and Interchange
8. Cohen, W.W., P. Ravikumar, und S.E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, 73–78
9. Danowski, P., und A. Pohl (Hrsg.). 2013. *(Open) Linked Data in Bibliotheken*. Bibliotheks- und Informationspraxis, Bd. 50. Berlin: De Gruyter Saur
10. D'Arcus, B., und F. Giasson. Bibliontology. Technical report
11. DCMI Usage Board. 2012. DCMI metadata terms. DCMI recommendation, Dublin Core Metadata Initiative
12. Dekkers, M., S. Gradmann, und C. Meghini. 2009. Europeana outline functional specification for development of an operational european digital library. Technical report. Europeana Thematic Network Deliverables 2.5. Contributors and peer reviewers: Europeana.net WP2 Working Group members, Europeana office
13. Dublin Core Metadata Element Set. 2008. Dublin Core metadata element set. DCMI recommendation, Dublin Core Metadata Initiative
14. Eversberg, B. 1994. *Was sind und was sollen Bibliothekarische Datenformate?* 2. Aufl.: Technische Universität Braunschweig Universitätsbibliothek
15. Gantert, K. 2008. *Bibliothekarisches Grundwissen* (8., vollst. neu bearb. und erw. Aufl. ed.). Saur

16. Geipel, M.M., C. Böhme, J. Hauser, und A. Haffner. 2013. Herausforderung Wissensvernetzung. In *(Open) Linked Data in Bibliotheken*, Hrsg. P. v. Danowski, A. Pohl, 168–185. De Gruyter Saur

17. Jaffe, E., und S. Kirkpatrick. 2009. Architecture of the internet archive. In *SYSTOR* ACM International Conference Proceeding Series., 11. ACM

18. Klee, C. 2013. Vokabulare für bibliographische Daten. In *(Open) Linked Data in Bibliotheken*, Hrsg. P. v. Danowski, A. Pohl, 45–63. De Gruyter Saur

19. Lange, C., P. Ion, A. Dimou, C. Bratsas, W. Sperber, M. Kohlhase, und I. Antoniou. 2012. Bringing mathematics to the web of data: the case of the mathematics subject classification. In *The Semantic Web* LNCS, Bd. 7295, Hrsg. E. Simperl, P. Cimiano, A. Polleres, O. Corcho, V. Presutti, 763–777. Springer

20. Panzer, M., und M.L. Zeng. 2009. Modeling classification systems in skos: Some challenges and best-practice recommendations. In *Proceedings of the International Conference on Dublin Core and Metadata Applications*. Dublin Core Metadata Initiative

21. Rowley, J., und R. Hartley. 2007. *Organizing knowledge: an introduction to managing access to information*. Ashgate

22. Rühle, S. (o.J.). *Kleines Handbuch Metadaten*. Kompetenzzentrum Interoperable Metadaten (KIM), siehe auch: http://kim-forum.org/Subsites/kim/SharedDocs/Downloads/DE/Handbuch/metadaten.pdf?_blob=publicationFile, aufgerufen am 02.10.2014

23. Sauermann, L., und R. Cyganiak. 2008, December. Cool URIs for the semantic web. W3C Interest Group Note, World Wide Web Consortium (W3C)

24. Summers, E., A. Isaac, C. Redding, und D. Krech. 2008. Lcsh, skos and linked data. In *Dublin Core*

Linked Data in der Automobilindustrie: Anwendungsfälle und Mehrwerte

12

Michael Gorriz und Kai Holzweißig

Man kann für eine große Klasse von Fällen der Benützung des Wortes „Bedeutung" – wenn auch nicht für alle Fälle seiner Benützung – dieses Wort so erklären: Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache. Und die Bedeutung eines Namens erklärt man manchmal dadurch, daß man auf seinen Träger zeigt (Ludwig Wittgenstein 2003, S. 40, Philosophische Untersuchungen).

Zusammenfassung

Unternehmen in der Automobilindustrie erleben gegenwärtig einen tiefgreifenden Wandel. Informationstechnologie bestimmt immer mehr die Art und Weise, wie Unternehmen arbeiten, und insbesondere die Entstehungsprozesse ihrer Produkte und Dienstleistungen. Kurzum: Die Idee des digitalen Unternehmens ist auch heute schon in traditionell geprägten Industriezweigen wie der Automobilindustrie zur Wirklichkeit geworden. Aufgrund der verschiedenen technologischen, organisationalen und kulturellen Herausforderungen, die dieser Paradigmenwechsel bedingt, bedarf es neuer Konzepte und Technologien, um diesen Wandel nachhaltig zu unterstützen. Im Rahmen des vorliegenden Artikels wird aufgezeigt, dass die Idee von „Linked Data" ein solches Konzept darstellen kann. Neben einer kurzen Diskussion der Grundlagen wird detailliert aufgezeigt, welche Anwendungsfälle und Mehrwerte für eine Praxisanwendung von Linked Data existieren.

12.1 Die digitale Wende in der Automobilindustrie

Unternehmen in der Automobilindustrie erleben gegenwärtig einen tiefgreifenden Wandel. Informationstechnologie bestimmt immer mehr die Art und Weise, wie Unternehmen

M. Gorriz ✉ · K. Holzweißig
Daimler AG, Stuttgart, Deutschland
e-mail: michael.gorriz@daimler.com

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_12

arbeiten, und insbesondere die Gestaltung ihrer Produkte, Dienstleistungen und Entstehungsprozesse.[1] Seien es Computer Aided Design (CAD) in der Konstruktion von Automobilen, Produktionssysteme in deren Herstellung oder die zunehmende Digitalisierung von Fahrzeugen, Informationstechnologie (IT) ist überall anzutreffen. In diesem Sinne beschreiben einige Autoren die Rolle der IT als eine Art „Engine of Industry Transformation" [14]. Als Motor dieses Transformationsprozesses ist die IT ein Treiber für den geschäftlichen Wandel im Unternehmen. Dabei wirkt die IT „as *deep infrastructure* capable of producing the vital *information capability* necessary for transformation" [8]. IT ist also integraler Bestandteil der Infrastruktur des Unternehmens und seiner wertschöpfenden Prozesse. Insofern kann durchaus von der Unternehmens-IT als „one of the threads from which the fabric of the organization is now woven" gesprochen werden [26]. Die umfassende Rolle der Informationstechnologie in vielerlei Aspekten des geschäftlichen Alltags kommt nicht zuletzt durch den Begriff des „Digital Enterprise" zum Ausdruck.

Die zunehmende Digitalisierung des Geschäftsalltags hat bedeutende Implikationen für das Selbstverständnis und die Arbeitsweise von IT-Abteilungen in klassischen Industriezweigen.[2] Das traditionelle „in-house"-Geschäft der IT als Unterstützer der Wertschöpfung in den einzelnen Fachbereichen wächst mit den Kernprozessen des Unternehmens in nie da gewesener Art und Weise zusammen. In der Automobilindustrie beispielsweise wird die IT ein immer bedeutenderer Bestandteil der Wertschöpfungskette. Seien es neue Fahrerassistenzsysteme, die Konnektivität von Fahrzeugen oder Mobilitätsdienstleistungen, wie Car2Go[3] oder Moovel[4] – IT spielt eine zentrale Rolle und dies ist erst der Anfang eines tiefgreifenden Wandels.

Der Paradigmenwechsel hin zu einem digitalen Unternehmen verlangt von der IT ein strategisches Umdenken bezüglich ihrer technologischen Landschaft, ihrer organisatorischen Strukturen sowie der Allokation von Ressourcen. Dies ist eine klassische Aufgabe der IT-Governance im Unternehmen. Hierunter werden „principles, procedures and measures which ensure that with the help of IT business goals are fulfilled, resources are used responsibly and risks are monitored adequately" verstanden [17]. Ein Schlüsselfaktor, um den Herausforderungen des digitalen Unternehmens zu begegnen, sind neue Konzepte und Technologien, die Mehrwerte schaffen, indem Kernprozesse effektiver und in kosteneffizienter Art und Weise unterstützt werden. Linked Data kann eine solche vielsprechende Technologie sein.[5] Um die mögliche Rolle von Linked Data im Transformationsprozess von Unternehmen hin zu einem „Digital Enterprise" näher zu erläutern, geht der vorliegende Artikel wie folgt vor. Zuerst wird eine Einführung in die zentralen Ideen und die Funktionsweise von Linked Data vorgenommen. Im nachfolgenden Abschnitt werden

[1] Vgl. hierzu beispielsweise die verschiedenen Beiträge in Cunha & Maropoulos [10].
[2] So führen beispielsweise Ngai & Gunasekaran [16] aus, welche Konsequenzen die zunehmende Digitalisierung für Unternehmen hinsichtlich des Managements hat.
[3] Siehe http://www.car2go.com, aufgerufen am 17.03.2014.
[4] Siehe http://www.moovel.com, aufgerufen am 17.03.2014.
[5] Zu den grundsätzlichen Motivationen einer Nutzung von Linked Data im Unternehmenskontext und der möglichen Potentiale vgl. beispielsweise die Ausführungen bei Allemang [1].

mehrere konkrete Anwendungsfälle für Linked Data vorgeschlagen und diskutiert. Eine Zusammenfassung am Ende des Artikels diskutiert die Erkenntnisse und stellt kritische Punkte für künftige Arbeiten heraus.

12.2 Was ist „Linked Data" und wie funktioniert es?

Bei Linked Data geht es im Kern um die Verwendung semantischer Netze zur besseren Beschreibbarkeit von Daten im Rahmen der Maschine-Maschine-Kommunikation (vgl. [7]). Die semantische Beschreibbarkeit von Daten stellt dabei ein wesentliches Differenzierungsmerkmal zu anderen Technologien dar, die in der Maschine-Maschine-Kommunikation genutzt werden. Charakterisierend für Linked Data ist nicht nur die Verwendung semantischer Netze, sondern insbesondere auch (vgl. [3]):

- Die durchgängige Anwendung der vier Linked Data Prinzipien, wie sie von Tim Berners-Lee formuliert wurden, um Daten im World-Wide-Web zu beschreiben, veröffentlichen, miteinander zu verknüpfen und abrufen zu können [2],
- Die Verwendung entsprechender, durch das World Wide Web Consortium (W3C) standardisierter Technologien, um eben dies tun zu können [24].

Um zu verstehen, wie Linked Data funktioniert, soll zuerst in die Idee der semantischen Netze eingeführt werden, bevor die oben angesprochenen vier Linked Data Prinzipien und die Verwendung standardisierter Linked Data Technologien näher betrachtet werden.

Das Oxford Dictionary of Philosophy erklärt das Wort „Semantik" wie folgt: „One of the three branches into which semiotics is usually divided: the study of the meaning of words, and the relation of signs to the objects to which the signs are applicable" [4]. Es geht also beim Thema „Semantik" – ganz grob gesprochen – um die „Bedeutung", um die Bedeutung von Worten, von Objekten und ihren Zusammenhang. Schon im antiken Griechenland, allen voran bei Platon und Aristoteles, wurde die Idee, dass sich Wörter direkt auf physikalische Entitäten beziehen, näher thematisiert (vgl. [9]). Dies ist beispielsweise zutreffend für konkrete Wörter wie „Baum" oder „Haus", die beide jeweils physikalische Korrelate besitzen. Schwieriger ist es mit abstrakten Wörtern wie „Glück" oder „Liebe", denen kein konkretes physikalisches Korrelat zugeordnet werden kann.[6] Anhand dieses Vergleichs ist erkennbar, dass eine reine Zuordnung von Wörtern zu Objekten nicht ausreichend ist, um die Wortbedeutung zu erklären. Dieses Problem wird gelöst, indem mit der menschlichen Kapazität zur Bildung mentaler Konzeptualisierungen ein drittes Element hinzugefügt wird. Dabei wird davon ausgegangen, dass keine direkte Beziehung zwischen Wörtern und Entitäten, die sie bezeichnen, existiert, sondern, dass diese Verbindung auf Basis von Interpretationsvorgängen anhand mentaler Konzepte erfolgt. Dieses

[6] Für eine detaillierte Diskussion des Spektrums abstrakter versus konkreter Wörter siehe Snodgras [21].

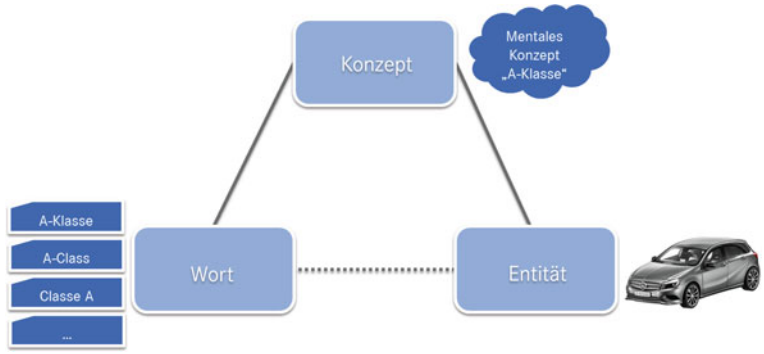


Abb. 12.1 Das semiotische Dreieck anhand der Mercedes-Benz A-Klasse

Denken in Form einer Separierung von Wortformen, mentaler Konzepte und Entitäten ist in der Wissenschaft weit akzeptiert (vgl. [12]). In der Linguistik und in der Philosophie wird die entsprechende Beziehung zwischen Wörtern, Konzepten und Entitäten oft in Form des sogenannten semiotischen Dreieckes diskutiert, welches unter anderem auf den französischen Linguisten Ferdinand de Saussure zurückgeht (vgl. [18]). Der zentrale Punkt dabei ist wie folgt: Ohne mentale Konzepte kann es keine Wortbedeutung geben. Diese Konzepte fungieren als „a kind of mental glue“, welche unsere innere Gedankenwelt „zusammenhält“ [15].

Um die Idee hinter dem semiotischen Dreieck näher zu erklären, ist in Abb. 12.1 ein Beispiel gezeigt. Hierbei wird die Explikation eines semiotisches Dreiecks anhand des Beispiels des Produktes „Mercedes-Benz A-Klasse“ vorgenommen. Die linke Seite des Dreiecks zeigt, dass verschiedene Ausdrücke als Bezeichner für das Produkt existieren. Während in der deutschen Sprache das Wort „A-Klasse“ gebraucht wird, wird im Englischen von „A-Class“ gesprochen, während im Französischem von „Classe A“ die Rede ist. Alle diese Ausdrücke (und alle anderen Ausdrücke, die synonym gebraucht werden) beziehen sich auf das selbe Konzept „A-KLASSE“.[7] Das mentale Konzept „A-KLASSE“ verweist wiederum auf die konkrete physikalische Entität einer A-Klasse, so dass die triangulare Beziehung geschlossen wird.

Eine bedeutende Grundidee von Linked Data liegt in der Explikation solcher semiotischer Dreiecke in Form von Netzen von Konzepten in einem maschinenlesbaren Format. Zur Modellierung solcher Netze wird auf (gerichtete) Graphen als Notationsform zurückgegriffen. Ein Graph ist eine Struktur, die aus Knoten und Kanten besteht, wobei jeder Knoten ein Konzept repräsentiert und die Relationen zwischen Konzepten durch Kanten dargestellt werden. Die kleinste Grundform eines semantischen Netzes, ein Subjekt-Prädikat-Objekt-Triple, wird demnach durch zwei Knoten und eine Kante zwischen ihnen

[7] Um zwischen Ausdrücken und Konzepten zu unterscheiden, werden Konzepte in Großbuchstaben geschrieben. Siehe hierzu beispielsweise Murphy [15].

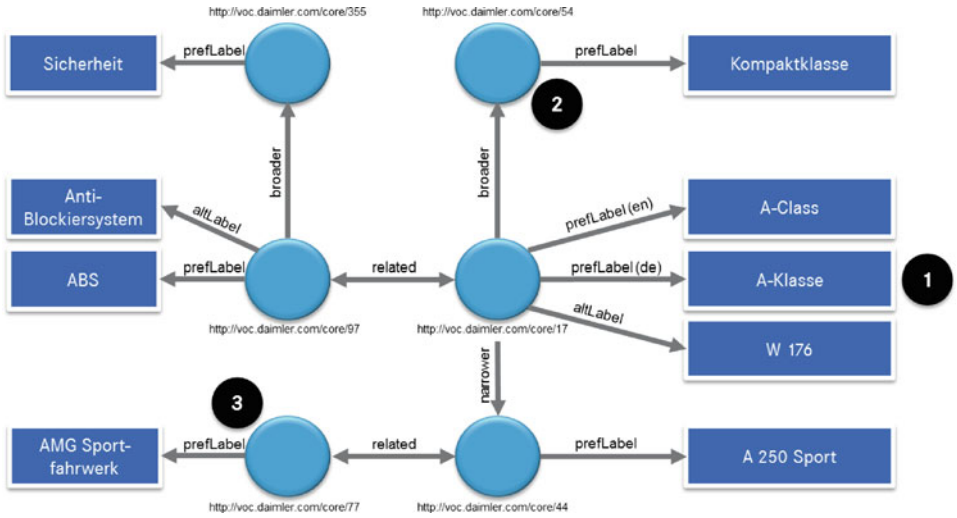


Abb. 12.2 Beispiel eines semantischen Netzes für „A-KLASSE“[8]

abgebildet. Jeder Knoten kann darüber hinaus mehrere Attribute besitzen, die ihn näher beschreiben. Dies trifft auch auf Kanten zu, wodurch eine qualitative Beschreibung von Relationen möglich wird. Eine Spezialform eines Attributes ist der sogenannte Bezeichner („label“), welcher eine bestimmte Form der Bezeichnung notiert, die bei der Nutzung kontrollierter Vokabulare von zentraler Bedeutung ist. Es gibt verschiedene Arten von Bezeichnern: Eine Vorzugsbenennung („preferred label“) zeigt an, dass die Verwendung des Ausdrucks in einer bestimmten Domäne zu nutzen ist, während alternative Bezeichner („alternative label“) synonym genutzte Ausdrücke notieren.

Um die vorstehenden Überlegungen zu veranschaulichen, ist in Abb. 12.2 ein entsprechendes kleines semantisches Netz dargestellt. Dieses Netz enthält mehrere Konzepte wie „KOMPAKTKLASSE“, „A-KLASSE“, „SICHERHEIT“ und so weiter. Wie ersichtlich ist, besitzt jeder Knoten mehrere Bezeichner. So zum Beispiel besitzt das Konzept „A-KLASSE“ die Vorzugsbenennung für die deutsche Sprache, nämlich „A-Klasse“ ❶. Zwischen den verschiedenen Konzepten des Graphs bestehen verschiedene hierarchische und assoziative Beziehungen. So ist das Konzept „COMPACT CLASS“ ein Oberkonzept von „A-CLASS“, welches wiederum ein Oberkonzept von „A 250 SPORT“ ist ❷. Das Konzept „A 250 SPORT“ steht in Beziehung zum Konzept „AMG SPORTFAHRWERK“, da eben dieses Sportfahrwerk in einem A 250 Sport verbaut wird ❸.

Neben der Grundidee von semantischen Netzen gibt es noch weitere Bestandteile, die Kernelemente von Linked Data darstellen. So sind die vier Prinzipien von Linked Data, wie sie Tim Berners-Lee aufgestellt hat, von zentraler Bedeutung [2]:

- *„Use URIs as names for things*
- *Use HTTP URIs so that people can look up those names.*

- *When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)*
- *Include links to other URIs, so that they can discover more things.“*

Um den Gebrauch von HTTP Unique Resource Identifiern (URIs) zu verdeutlichen, sei auf das Beispiel in Abb. 12.2 verwiesen. Hier wird jedes Konzept mittels eines URIs, basierend auf dem bekannten HTTP-Protokoll, referenziert.

Neben semantischen Netzen und den vier genannten Prinzipien ist auch der Gebrauch entsprechender offener W3C Standards, welche nachfolgend aufgelistet sind, ein weiteres wesentliches Kernmerkmal von Linked Data:[9]

- Das Resource Description Framework (RDF) zur Datenrepräsentation
- Die SPARQL Protocol and RDF Query Language (SPARQL) zur Datenabfrage
- Das Simple Knowledge Organization System (SKOS) zur Wissensorganisation
- Die Web Ontology Language (OWL) zur formalen Beschreibung von Daten

Diese offenen Standards bilden eine wichtige Grundlage von Linked Data, da sie eine einheitliche und maschinenlesbare Form zur Entwicklung und zum Betrieb entsprechender semantischer Anwendungen bereitstellen. Durch die Verwendung der beschriebenen W3C Standards, wie RDF, SKOS, OWL und SPARQL, wird die Qualität entsprechender Datenrepräsentation entscheidend verbessert. Daten „erhalten“ mittels dieser Standards eine Bedeutung, die durch Maschinen verarbeitet werden kann. Die Linked Open Data (LOD) Cloud[10] hat in beeindruckender Art und Weise gezeigt, wie dies funktionieren kann.

Bevor zum nächsten Abschnitt übergegangen wird, soll noch eine kurze Klarstellung getroffen werden, um die Verwendung des Begriffes „Semantik“ im Zusammenhang mit Maschinen genauer zu erläutern. In den nachstehenden Ausführungen soll nicht der Eindruck erweckt werden, dass Computer oder Informatiksysteme *verstehen können* in demselben Sinne, wie Menschen Wortbedeutungen verstehen. So hat bereits der Philosoph John Searle in seinem Chinese Room Argument ausgeführt, dass Computer nur auf der Basis vordefinierter Regeln Informationen verarbeiten, so dass in einem geschickt arrangierten Szenario bei einem Beobachter der Eindruck erweckt werden könne, dass ein tatsächliches Verständnis vorliege (vgl. [19]). Tatsächlich findet einzig und alleine eine Reduktion semantischer Informationen in regelbasierter und syntaktischer Form statt, welche naturgemäß „unvollständig“ sein muss. Die Maschine *versteht* daher beispielsweise nicht, *was* eine „A-KLASSE“ ist. Insofern unterscheidet sich die menschliche Semantik von der maschinellen Semantik. Dieses kann – verkürzt gesprochen – auch als Grund gesehen werden, warum vollautomatische Verfahren zum Aufbau und zur Wartung von semantischen Netzen keine befriedigenden Ergebnisse liefern können.

[9] Vgl. W3C [24] und die entsprechenden Unterseiten sowie Stephens [22].
[10] Siehe http://linkeddata.org und http://lod-cloud.net, aufgerufen am 17.03.2014.

12.3 Anwendungsfälle in der Automobilindustrie

Es existiert eine Vielzahl von möglichen Anwendungsfällen für Linked Data in der Automobilindustrie.[11] Im Folgenden wird vertiefend auf einige dieser Anwendungsfälle eingegangen, um die Potentiale und Mehrwerte, aber auch um die Vielseitigkeit von Linked Data, herauszustellen. Die ausgewählten Anwendungsfälle sind wie folgt:

- Semantisch gestützte Suchanwendungen
- Semantisch gestützte Content-Management-Anwendungen
- Semantisch gestützte Systemintegration in Produktentstehungsprozessen
- Semantisch gestützte Einkaufsprozesse
- Semantisch gestütztes Trend-Scouting

12.3.1 Semantisch gestützte Suchanwendungen

Ein sehr illustrativer Anwendungsfall, um die Potentiale von Linked Data zu verdeutlichen, sind semantische Suchanwendungen.[12] Worin unterscheiden sich diese Suchanwendungen von der klassischen Suche, die wir gewohnt sind? Der Hauptunterschied zwischen der semantischen und der klassischen Suche ist wie folgt. Klassische Suchanwendungen arbeiten primär mit Wortformen auf Basis von Wörterbüchern, syntaktischer Algorithmik und quantitativer Methoden. Semantische Suchanwendungen hingegen gehen einen entscheidenden Schritt weiter. Sie nutzen zusätzlich im Hintergrund semantische Wissensnetze, um so die Suchanfragen besser kontextualisieren und entsprechende Resultate liefern zu können. Damit dies funktioniert, werden Suchterme mit entsprechend passenden Konzepten im unterliegenden Graphen in Beziehung gesetzt. Wurde ein entsprechendes Konzept zu einer Suchanfrage gefunden, so wird das Wissen über das Konzept und seinen Kontext, das heißt seine Relationen zu anderen benachbarten Konzepten, genutzt, um bessere Suchresultate liefern zu können. Beispielsweise verwendet Google ein ähnliches Vorgehen im Rahmen der Knowledge Graph Technologie.[13]

Um die Funktionsweise von semantischen Suchanwendungen zu verdeutlichen, ist ein entsprechendes Beispiel in Abb. 12.3 illustriert. Ein Kunde trägt das Wort „A-Klasse“ in ein Suchformular ein und klickt auf „Finde!“ ❶. Die Suchanfrage wird daraufhin zu einem Suche-Server geschickt und dort evaluiert. Nachdem der Suchterm vorverarbeitet wurde (Rechtschreibkorrektur, Stammformreduktion etc.), werden passende Konzepte aus dem unterliegenden Graphen ermittelt. In dem obigen Beispiel wird das Konzept „A-

[11] Für weitere Anwendungsfälle, auch in anderen Kontexten, vgl. beispielsweise Stephens [22] und Allemang [1].

[12] In der Literatur lassen sich eine Reihe von Arbeiten zum Themengebiet der semantisch gestützten Suchanwendungen finden. Vgl. hierzu beispielsweise die angeführte Literatur bei Blanco et al. [5].

[13] Siehe http://www.google.com/insidesearch/features/search/knowledge.html, aufgerufen am 17.03.2014.

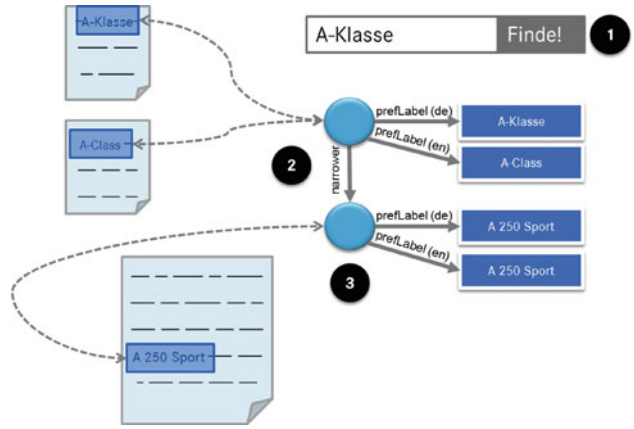


Abb. 12.3 Suchanwendung auf Basis semantischer Netze[14]

KLASSE“ mit allen seinen Inhalten über die Vorzugsbenennung „A-Klasse“ gefunden ❷. Ferner werden auch alle Inhalte, die verwandten Konzepten entsprechen, das heißt Sub-Konzepte oder Konzepte, die in einer assoziativen Beziehung zum Konzept „A-KLASSE“ stehen, gefunden. Bezogen auf das angeführte Beispiel heißt das, dass auch Informationen zum Konzept „A 250 SPORT“ gefunden werden, da der Suche bekannt ist, dass „A 250 SPORT“ ein Sub-Konzept des Konzepts „A-KLASSE“ ist ❸.

Um die prinzipielle Tauglichkeit der vorstehend diskutierten semantischen Suchanwendung und ihre Effekte auf die User Experience zu zeigen, wurde ein entsprechender Prototyp auf der deutschen Produktseite (http://www.mercedes-benz.de) implementiert. Die implementierte Anwendung ermöglicht eine konzeptbasierte Suche. Ein Screenshot dieser Suche ist in Abb. 12.4 dargestellt.

Die Art und Weise, wie diese Suchanwendung funktioniert, ist nachfolgend kurz beschrieben. Ein potentieller Kunde kommt auf die Seite und möchte nach allen Fahrzeugtypen suchen, die Kombicharakter haben. Hierzu tippt er die Buchstaben „Kom“ für das Wort „Kombi“ ein. Schon während des Eintippens der ersten drei Buchstaben erscheint direkt ein entsprechendes Ergebnisfenster, das entsprechende Suchresultate darstellt ❶. Da allerdings die Benennung „Kombi“ keine offizielle Mercedes-Benz-Benennung darstellt, sondern ein Kundensynonym für alle Konzepte mit der Eigenschaft „T-Modell“ ist, muss von der Suche eine entsprechende Assoziation aufgestellt werden. Im vorliegenden Beispiel ist die Benennung „Kombi“ als Synonym für alle T-Modell-Fahrzeugkonzepte gepflegt, so kann diese Assoziation entsprechend von der Suchanwendung aufgelöst werden, um dem Kunden die von ihm gewünschten und intendierten Resultate zu liefern ❷. Durch eine Anreicherung des unterliegenden semantischen Netzes um weitere Kundensynonyme, kann eine zielgruppengerechte Suche entsprechend umgesetzt werden. Anhand des gewählten Beispiels ist auch sichtbar, dass zu dem Suchbegriff „Kom“ nicht nur

[14] Die ursprüngliche Version dieser Abbildung stammt von Andreas Blumauer und wurde im Rahmen eines Projekts zwischen der Daimler AG und der Semantic Web Company erarbeitet.

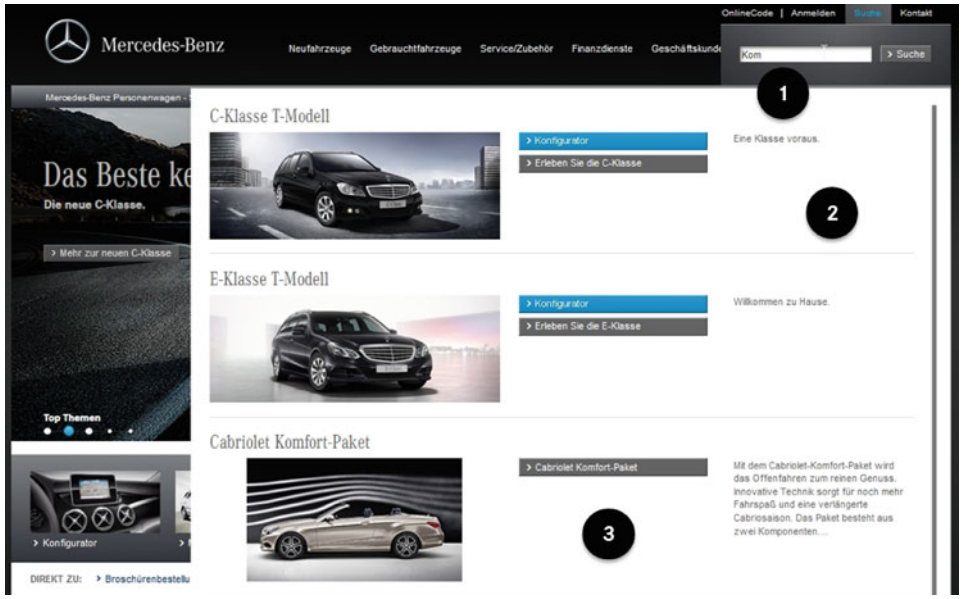


Abb. 12.4 Die semantische Suche auf http://www.mercedes-benz.de

T-Modell-Fahrzeuge gefunden werden, sondern auch Ausstattungsvarianten, hier „Cabriolet Komfort-Paket“ ❸. Dieses Suchresultat verschwindet aus der Ergebnisanzeige, wenn vom Nutzer der Suchbegriff „Kom“ um „bi“ für „Kombi“ erweitert wird.

12.3.2 Semantisch gestützte Content-Management-Anwendungen

Ein zweites Anwendungsszenario für Linked Data, welches mit der vorgestellten semantischen Suche eng verwandt ist, ist der Gebrauch im Rahmen von Content Management Prozessen.[15] Aufgrund der hohen Informationsintensität von unternehmerischen Kernprozessen wie der Entwicklung, der Produktion oder dem Verkauf und Marketing, sind strukturierte Content-Lebenszyklus-Prozesse von großer Bedeutung. Damit Informationen effizient und effektiv eingesetzt werden können, ist es wichtig, dass sie gehaltvolle und präzise Metadaten enthalten und diese geordnet abgelegt und wiedergefunden werden können. Den Ausgangspunkt hierfür bildet wieder ein unterliegendes semantisches Netz, welches die konzeptuelle Struktur der fraglichen Content-Domäne abbildet. Anhand dieses Netzes, welches bereits standardisierte Konzepte enthält, können Content-Bausteine im Erstellungsprozess in Form von Konzept-Tags ausgezeichnet werden. Das heißt, dass

[15] Auch zum Themengebiet der semantisch gestützten Content-Management-Anwendungen lassen sich in der Literatur eine Reihe von Arbeiten finden, so beispielsweise bei Gams & Mittersdorfer [11], Blumauer & Hochmeister [6] oder bei Šimko et al. [20].

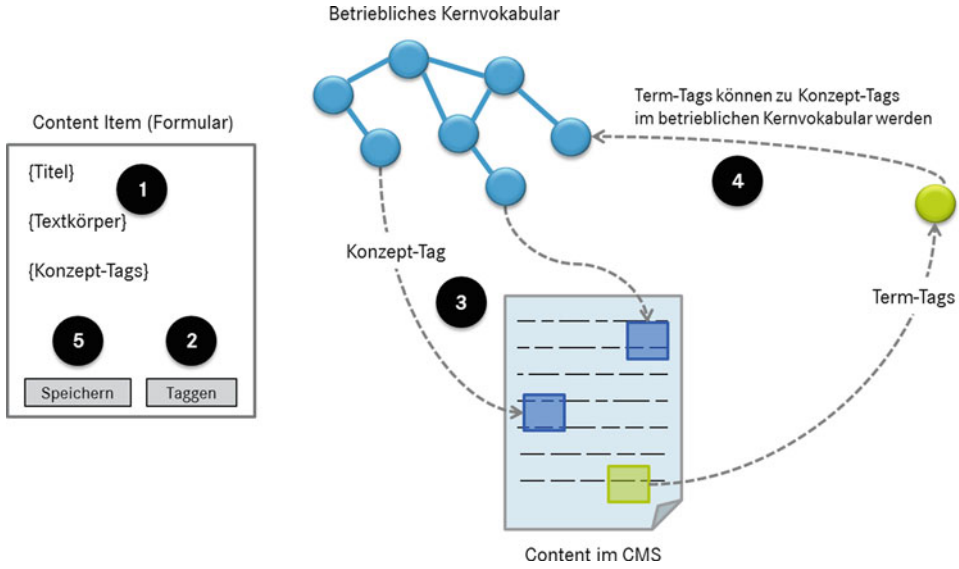


Abb. 12.5 Konzept-Tagging in Content-Management-Anwendungen[17]

zwischen einzelnen Content-Bausteinen und Konzepten eine dauerhafte Zuordnung erfolgen kann, um auf der Metadatenebene entsprechende Auswertungen vornehmen zu können. Diese Arbeit wird durch semi-automatisierte Prozesse unterstützt, indem die Anwendung dem Content-Redakteur anhand automatisiert durchgeführter Content-Analysen entsprechende Konzepte zur Auszeichnung der Inhalte vorschlägt. Ein solches Vorgehen kann als „Conceptual Tagging" bezeichnet werden.[16] Wie genau die Idee der konzeptuellen Auszeichnung von Inhalten funktioniert, ist in Abb. 12.5 dargestellt.

Im vorstehenden Beispiel erstellt ein Redakteur ein Dokument in einem beliebigen Content-Management-System. Dieses Dokument besteht aus verschiedenen inhaltlichen Abschnitten, wie beispielsweise einer Titelzeile und dem eigentlichen Textkörper ❶. Nach der Fertigstellung aller textuellen Inhalte nutzt der Redakteur eine entsprechende Anwendungsfunktion, um eine konzeptbasierte Auszeichnung seines erstellten Dokumentes vorzunehmen ❷. Bei Betätigung der Funktion werden die Inhalte des Dokuments an einen zentralen Web-Service zur Auszeichnung gesendet. Dieser Web-Service nutzt Text-Mining-Algorithmen, um bestimmte Schlüsselterme zu extrahieren und diese anschließend zu den entsprechend hinterlegten Konzepten im semantischen Netz, dem betrieblichen Vokabular, zuzuordnen ❸. Für gefundene Schlüsselterme, die keinen Konzepten zugeordnet werden können, werden sogenannte „Term-Tags" vorgeschlagen, aus denen dann anschließend vollwertige Konzepte im Vokabular angelegt und gepflegt werden kön-

[16] Zur Idee des „Conceptual Taggings" vgl. ferner auch die Ausführungen bei Stojanovic et al. [23] und Allemang [1].

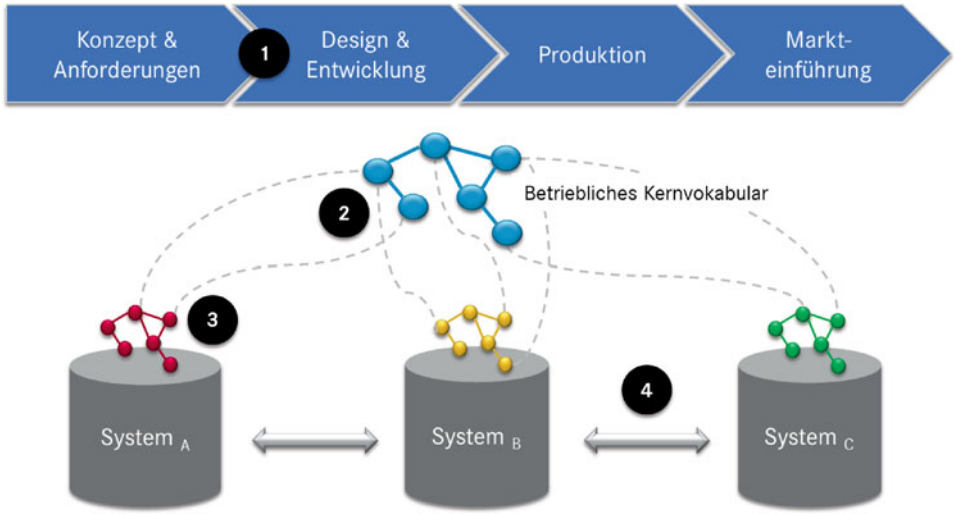


Abb. 12.6 Systemintegration entlang kontrollierter Vokabulare im PEP

nen ❹. In dieser Art und Weise können direkt neue Konzepte zum unterliegenden semantischen Netz hinzugefügt werden. Ist die konzeptbasierte Auszeichnung des Dokumentes erfolgreich beendet worden, so kann der Autor die gewohnte Speichern-Funktion seines Content-Management-Systems nutzen, um das Dokument samt seiner semantischen Metainformationen zu persistieren und in den Veröffentlichungs-Workflow zu geben ❺.

12.3.3 Semantisch gestützte Systemintegration in Produktentstehungsprozessen

Ein weiteres Anwendungsszenario ist die Verwendung von Linked Data in der Integration von Stammdatensystemen im Produktentstehungsprozess anhand kontrollierter Vokabulare.[18] Dieses Szenario ist schematisch in Abb. 12.6 dargestellt.

Produktentstehung ist der Kernprozess schlechthin in der Automobilindustrie. Gewöhnlich bestehen Produktentstehungsprozesse aus mehreren aufeinanderfolgenden Phasen wie Konzept- und Anforderungsanalyse, Produktgestaltung und -entwicklung, Produktion, Verkauf und After Sales ❶. Produktentstehung ist ein hochgradig informationsintensiver und arbeitsteiliger Prozess, sodass Kooperation und Kommunikation der einzelnen Beteiligten wesentliche Erfolgsfaktoren darstellen. Dabei ist es unabdingbar, dass die Beteiligten anschlussfähige Verständnisweisen über das zu entstehende Produkt und seinen Entstehungsprozess bilden. Aus technischer Perspektive erfordert Ko-

[18] Eine Nutzung von Linked Data im Rahmen von Stammdatensystemen wird beispielsweise bei Allemang [1] thematisiert.

operation die Kompatibilität und Integration von Daten verschiedener Quellsysteme. So müssen beispielsweise im Produktentstehungsprozess Zeichnungsdaten aus der Konstruktion mit Stücklisten- und Produktionsdaten verknüpft werden. Menschliche Arbeit wird dann effizienter, wenn die unterliegenden Systeme, die die entsprechenden operativen Prozesse unterstützen, miteinander verbunden sind und medienbruchfrei Informationen austauschen können. So können sich die Mitarbeiter voll und ganz auf ihre inhaltlichen Aufgaben konzentrieren.

Die Nutzung von Linked Data in der Systemintegration beruht auf den folgenden zwei Aspekten:

- Der Erstellung, dem Gebrauch und der Weiterentwicklung eines gemeinsamen betrieblichen Vokabulars ❷
- Der Nutzung offener W3C Standards, um Daten der verschiedenen beteiligten Stammdatensysteme publizieren und austauschen zu können ❸

Genauso wie Menschen, die sinnhaft miteinander kommunizieren wollen, müssen Computersysteme, die gegenseitig Daten austauschen, über ein entsprechendes gemeinsames Vokabular verfügen. Solch ein systemübergreifendes Vokabular bildet die Basis für eine Systemintegration. Ausdrücke, die in einem System genutzt werden, müssen über Konzepte in diesem betrieblichen Vokabular zu Bezeichnern, die in dem Kontext anderer Systeme gebraucht werden, zugeordnet werden ❷. Ist ein solches gemeinsames Vokabular erstellt worden, muss sichergestellt werden, dass die Daten der verschiedenen beteiligten Systeme gegenseitig auch tatsächlich zugreifbar sind, um so sinnstiftende Verknüpfungen zwischen ihnen herstellen zu können. Um diese „Datensilos“ zu öffnen, stehen verschiedene Frameworks und W3C Standards zur Verfügung. Beispielsweise können existierende Systeme erweitert werden, indem diese mit einem SPARQL-Endpoint versehen werden, der als Linked Data Schnittstelle dient. Eine weitere Möglichkeit besteht in der Nutzung von „Mapping-Services“, um die Daten eines Systems in ein gemeinsames offenes Format zu transformieren ❸. Sind ein gemeinsames Vokabular und ein offener Zugang zu den entsprechenden Quellsystemen mittels der W3C Standards implementiert worden, kann eine medienbruchfreie Kommunikation im Sinne von Linked Data zwischen den Einzelsystemen stattfinden ❹.

12.3.4 Semantisch gestützte Einkaufsprozesse

Einen weiteren exemplarischen Anwendungsfall stellt der elektronische Austausch von Informationen in Kunden-Lieferanten-Szenarien im Rahmen von Beschaffungsketten dar. Für Automobilhersteller sind diese Beschaffungsketten von zentraler Bedeutung, da sie auf die Nutzung von Gütern und Dienstleistungen von externen Lieferanten angewiesen sind. Die Beschreibung der angebotenen Güter und Dienstleistungen erfolgt im Rahmen sogenannter Service-Kataloge. In diesen elektronischen Katalogen werden die einzelnen

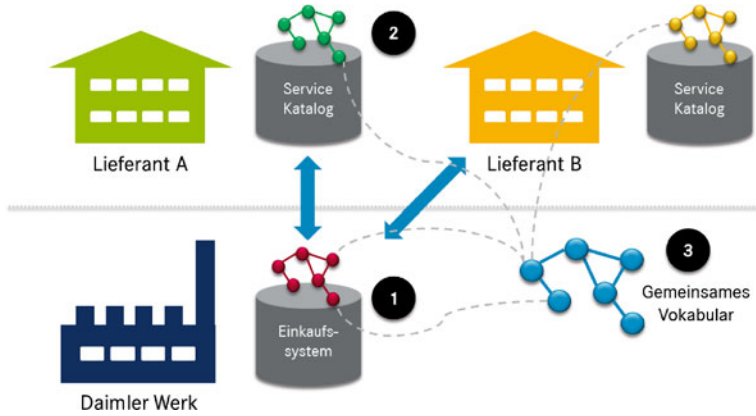


Abb. 12.7 Linked Data in einem Lieferanten-Konsumenten-Szenario

Dienstleistungen und angebotenen Güter beschrieben. Aufgrund sich ändernder Preis- und Produktportfolioinformationen der Lieferanten befinden sich die Kataloginhalte in einem stetigen Wandel, was eine kontinuierliche Aktualisierung auf der Konsumentenseite notwendig macht. Der Abruf und die Synchronisation von Kataloginformationen auf der Seite des Automobilherstellers können entweder in digitaler oder in analoger Form erfolgen. In digitalen Szenarien kommen häufig proprietäre Formate zum Datenaustausch zum Einsatz. Erschwert wird der Datenaustausch ferner dadurch, dass verschiedene Lieferanten unterschiedliche Formate einsetzen und der Automobilhersteller als Konsument sein Einkaufssystem auf diese verschiedenen Formate abstimmen muss. Aufgrund einer oft fehlenden Automatisierung in der Synchronisation von Service-Katalogen, müssen ferner Änderungen in manueller Art und Weise im Einkaufssystem durchgeführt werden. Durch den Einsatz von Linked Data, so die Idee, kann der Datenaustausch- und Datenabgleichsprozess wesentlich verbessert werden.

Die Idee hinter der oben diskutierten Funktionsweise der Verwendung von Linked Data in Lieferanten-Konsumenten-Szenarien ist in Abb. 12.7 skizziert. Auf Seiten des Automobilherstellers wird ein Einkaufssystem zur Beschaffung von Dienstleistungen und Gütern eingesetzt ❶. Dabei publizieren die Lieferanten ihre Informationen auf Basis von RDF entlang eines entsprechenden standardisierten Schemas ❷. Das Einkaufssystem bezieht aus den Service-Katalogen der einzelnen Lieferanten die verschiedenen Lieferantenstammdaten und Bestellinformationen. Damit das Einkaufssystem und die verschiedenen Katalogsysteme der Lieferanten zusammenarbeiten können, ist ein gemeinsames Vokabular notwendig, in dem die verschiedenen Benennungen und Konzepte, die von den unterschiedlichen Teilnehmern verwendet werden, aufeinander abgeglichen sind ❸. Ändert nun ein Lieferant seine Daten, werden diese Änderungen „live“ publiziert und können entsprechend vom Einkaufssystem automatisiert abgerufen und verarbeitet werden. So ist gewährleistet, dass ohne manuelle Schritte die Daten im Einkaufssystem aktuell gehalten werden.

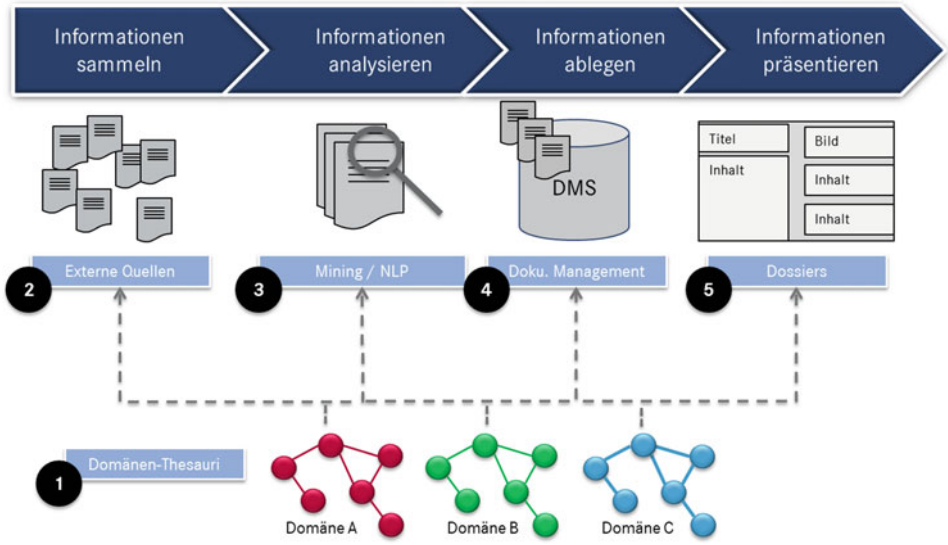


Abb. 12.8 Linked Data im Technologie-Scouting-Prozess

12.3.5 Semantisch gestütztes Trend-Scouting

Der letzte Anwendungsfall, der im Rahmen dieses Artikels thematisiert werden soll, ist die Verwendung von Linked Data im Rahmen des Technologie-Scoutings. Ziel des Technologie-Scoutings ist es, neue Technologien, die im Rahmen von neuen Produkten oder Produktionsverfahren eingesetzt werden können, zu identifizieren und ihre Nutzbarmachung zu analysieren. Aufgrund der Informationsmenge, die hierfür ausgewertet werden muss, ist der Rückgriff auf automatisierbare Verfahren unerlässlich. Das prinzipielle Vorgehen ist in Abb. 12.8 dargestellt. Im Technologie-Scouting-Prozess müssen im ersten Schritt Informationen aus verschiedenen externen Quellen (Internet, Fachzeitschriften, Datenbanken etc.) gesammelt werden. Im nächsten sind die gesammelten Informationen zu analysieren, um die behandelten Inhalte näher zu identifizieren und diese zu klassifizieren. Danach werden die als brauchbar identifizierten Informationen abgelegt, um sie weiter verwenden zu können. Aus der Menge abgelegter Informationen sind im letzten Schritt zielgruppenspezifische Kurzzusammenfassungen bzw. Dossiers zu erstellen, um so die Informationen an die richtigen internen Ansprechpartner zu verteilen.

Auch der vorstehend beschriebene Technologie-Scouting-Prozess kann durch Linked Data an verschiedenen Stellen unterstützt werden. Das Kernelement bei dieser Unterstützung bilden sogenannte Domänen-Thesauri. Diese Thesauri können mit dem W3C SKOS Standard umgesetzt werden und organisieren und repräsentieren das Kernwissen um eine

bestimmte technologische Domäne ❶. Bei diesem Wissen handelt es sich unter anderem um die Konzepte, ihre Beziehungen zueinander sowie um die verschiedenen Benennungen dieser Konzepte, die diese technologische Domäne charakterisieren. In diesem Sinne können die entsprechenden Thesauri als Ordnungs- und Klassifizierungsinstrument aufgefasst werden, das in den verschiedenen Phasen des Technologie-Scouting-Prozesses als elementarer Bestandteil zum Einsatz kommt. Im Rahmen der Sammlung von Informationen können die Thesauri dazu eingesetzt werden, um über entsprechende Schlüsselwörter und deren Synonyme relevante Informationen zu entdecken ❷. Bei der darauffolgenden Analyse der gefundenen Informationen kann wiederum auf die Strukturinformationen der Domänen-Thesauri zurückgegriffen werden ❸.

Durch entsprechende Mining- und Natural-Language-Processing-Algorithmen können entlang der Thesauri-Informationen potentiell interessante Inhalte gefiltert und extrahiert werden. Die extrahierten Informationen werden daraufhin mit entsprechend generierten Metadaten in einem persistenten Speicher, beispielsweise einem Dokumentenmanagement-System, abgelegt ❹. Die abgelegten Inhalte werden dabei mit Konzepten aus dem jeweiligen Domänen-Thesaurus assoziiert, was eine Weiterverwendung der einzelnen Informationsbausteine wesentlich vereinfacht. Dieses zum Beispiel bei der Nutzung der gewonnenen und abgelegten Informationen im Rahmen von zielgruppenorientierten Dossiers bzw. Mashups. Hierbei werden, basierend auf den Strukturinformationen der angeschlossenen Domänen-Thesauri, die abgelegten Technologieinformationen über ihre Metadaten in Form von multimedialen Artikeln aufbereitet und den potentiell interessierten Anwendern zur Verfügung gestellt ❺.

Wie gezeigt wurde, ist das Potential von Linked Data im Umfeld der Automobilindustrie vielfältig. Dies trifft insbesondere auf Themenstellungen zu, in denen es um die Publikation und den Abruf sowie die Verarbeitung von verschiedensten Stammdaten (Produktstammdaten, Lieferantenstammdaten, Technologiestammdaten etc.) geht. Allerdings muss die letztendliche Tauglichkeit von Linked Data für solche Szenarien noch durch konkrete Proof-of-Concepts belegt werden.

12.4 Zusammenfassung und Ausblick

Der vorliegende Artikel hat dargestellt, dass Linked Data eine vielversprechende Technologie sein kann, um den strukturellen Wandel von Automobilherstellern hin zu „Digital Enterprises“ zu unterstützen. Die Ausführungen zu den vorstehend vorgestellten Anwendungsfällen haben dies beispielhaft gezeigt. Allerdings sind abschließend mehrerlei kritische Punkte zu nennen, die Bestandteil einer weitergehenden Evaluation und Diskussion von Linked Data im Enterprise-Kontext sein müssen. Eine Adressierung dieser Punkte ist insofern wichtig, um die Technologie aus ihrem gegenwärtigen innovativen Status herauszuführen und sie als eine echte enterprisefähige Technologie etablieren zu können.

Dies vor allen Dingen vor dem Hintergrund, dass viele Linked-Data-Softwareprodukte aus akademischen Kontexten heraus entstanden sind und nicht primär die Anforderungen geschäftlicher Anwendungsfelder im Fokus hatten.[19] Um den Ansprüchen geschäftlicher Anwendungen gerecht zu werden, sind insbesondere, aber nicht alleine, die folgenden Punkte zu berücksichtigen:[20]

- Linked Data Softwareprodukte müssen auf enterprisefähigen Basisprodukten aufsetzen, die sicher, verfügbar, skalierbar etc. sind. Ferner müssen sich die Produkte in die existierende technologische Landschaft und Infrastruktur des Unternehmens einfügen. Eine ganze Reihe von Linked Data Produkten ist zum heutigen Zeitpunkt leider noch nicht in der Lage den genannten Anforderungen Rechnung zu tragen, was unter anderem mit ihrer primär akademischen Ausrichtung zusammenhängt.
- Verschiedene technische Fragestellungen bedürfen einer näheren Betrachtung. Dies trifft beispielsweise auf die Ableitung kosteneffizienter Methoden und Vorgehensmodelle zu, um existierende Produktivsysteme bereit für Linked Data zu machen. Unter diesem Punkt verbergen sich unter anderem Themen der Konvertierung von Bestandsdaten in entsprechende Linked Data Formate sowie die Ableitung entsprechender Beschreibungsschemata. Auch Themen der Authentifizierung und Autorisierung auf entsprechende Linked Data Graphen sind an dieser Stelle zu nennen.
- Bisherige Betrachtungen zum Thema „Linked Data“ fokussieren primär auf technische Fragestellungen. Die Einführung einer neuen Technologie ist aber selten ein rein technisches Thema, weil sie in die einführende Organisation in vielerlei Art und Weise zurückwirkt. Zu wenig ist über Wege und Mittel bekannt, um mit den verschiedenen nicht-technischen Implikationen dieser Technologie umzugehen, so zum Beispiel kulturelle, rechtliche und ökonomische Fragestellungen, die entsprechend Beachtung finden müssen.
- Es bedarf einer ausgereiften Methodik, um den Lebenszyklus von Linked Data Anwendungen im Rahmen einer IT-Governance zu überwachen und zu steuern. So bestehen beispielsweise offene Fragen, wie unternehmensweit einsetzbare Plattformen und Services basierend auf Linked Data Technologien aufgebaut, gewartet und weiterentwickelt werden können. Hier bedarf es entsprechender Konzepte und Handlungsempfehlungen, um eine durchgängige IT-Governance der Technologie ausüben zu können.
- Bei der Kommunikation von Linked Data Themen muss darauf geachtet werden, dass die Möglichkeiten und Grenzen der Technologie angemessen wiedergegeben werden. Dies trifft insbesondere auf die Aufwände zu, die im Zusammenhang mit der Etablierung der Technologie im Unternehmen stehen. Generell muss einer Überschätzung der

[19] Für eine Diskussion von Linked Data Softwareprodukten für geschäftliche Anwendungen vgl. beispielsweise Stephens [22].

[20] Weitere Herausforderungen werden zum Beispiel auch bei Hyland [13] sowie bei Cardoso et al. [7] diskutiert.

Technologie vorgebeugt werden. Auf die Grenzen von dem, was theoretisch möglich, das heißt durch die Technologie leistbar ist, wurde am Ende des zweiten Abschnittes dieses Artikels eingegangen.

- Der Markt für Implementierungspartner für Linked Data Anwendungen ist vergleichsweise klein. Mehr Wettbewerb in diesem Sektor würde die Kosteneffizienz, Anwendungstauglichkeit und die Weiterentwicklung der Technologie positiv beeinflussen.

Es würde uns freuen, wenn die vorstehenden Punkte im positiven Sinne als Anregung für weitere Forschungen und die praktische Auseinandersetzung mit dem Thema verstanden werden.

Literatur

1. Allemang, D. 2010. Semantic Web and the Linked Data Enterprise. In *Linking Enterprise Data*, Hrsg. D. Wood, 3–23. Springer Science + Business Media
2. Berners-Lee, T. 2009. *Linked Data – Design Issues*. http://www.w3.org/DesignIssues/LinkedData.html. Zugegriffen: 17.03.2014
3. Bizer, C., T. Heath, und T. Berners-Lee. 2009. Linked data – the story so far. *International Journal on Semantic Web and Information Systems* 5(3): 1–22
4. Blackburn, S. 2008. *The Oxford Dictionary of Philosophy*, 2. Aufl., Oxford, UK: Oxford University Press
5. Blanco, R., H. Halpin, D.M. Herzig, P. Mika, J. Pound, H.S. Thompson, und T. Tran. 2013. Repeatable and reliable semantic search evaluation. *Web Semantics: Science, Services and Agents on the World Wide Web* 21: 14–29
6. Blumauer, A., und M. Hochmeister. 2009. Tag-Recommender gestützte Annotation von Web-Dokumenten. In *Social Semantic Web*, Hrsg. A. Blumauer, T. Pellegrini, 227–243. Berlin, Heidelberg: Springer-Verlag
7. Cardoso, J., M. Lytras, und M. Hepp. 2008. The Future of the Semantic Web for Enterprises. In *The Semantic Web – Real-World Applications from Industry*, Hrsg. M.D. Lytras, 3–15. New York: Springer Science + Business Media
8. Cooper, R.G., und E.J. Kleinschmidt. 1991. New Product Processes at Leading Industrial Firms. *Industrial Marketing Management* 20(2): 137–147
9. Crystal, D. 1993. *Die Cambridge Enzyklopädie der Sprache*. Frankfurt a. M.: Campus Verlag
10. Cunha, P.F., und P.G. Maropoulos (Hrsg.). 2007. *Digital Enterprise Technology – Perspectives and Future Challenges*. New York: Springer-Verlag
11. Gams, E., und D. Mitterdorfer. 2009. Semantische Content Management Systeme. In *Social Semantic Web*, Hrsg. A. Blumauer, T. Pellegrini, 207–226. Berlin, Heidelberg: Springer-Verlag
12. Herrmann, T. 1994. Psychologie ohne Bedeutung: Zur Wort-Konzept-Relation in der Psychologie. *Sprache & Kognition* 13: 126–137

13. Hyland, B. 2010. Preparing for a Linked Data Enterprise. In *Linking Enterprise Data*, Hrsg. D. Wood New York: Springer Science + Business Media

14. King, J.L., und K. Lyytinen. 2005. Automotive Informatics: Information Technology and Enterprise Transformation in the Automobile Industry. In *Transforming Enterprise*, Hrsg. W.H. Dutton, B. Kahin, R. O'Callaghan, A. Wycoff, 283–312. Cambridge, MA: MIT Press

15. Murphy, G.L. 2002. *The big book of concepts*. Cambridge, MA: MIT Press

16. Ngai, E.W.T., und A. Gunasekaran. 2007. Managing Digital Enterprise. *International Journal of Business Information Systems* 2(3): 266–275

17. Rüter, A., J. Schröder, und A. Göldner (Hrsg.). 2006. *IT-Governance in der Praxis: Erfolgreiche Positionierung der IT im Unternehmen. Anleitung zur erfolgreichen Umsetzung regulatorischer und wettbewerbsbedingter Anforderungen*. Berlin, Heidelberg: Springer-Verlag

18. Saussure, F. de. 2001 [1916]. *Grundfragen der allgemeinen Sprachwissenschaft*. 3. Auflage. Berlin: de Gruyter

19. Searle, John R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(3): 417–457

20. Šimko, M., M. Franta, M. Habdák, und P. Vrablecová. 2013. Managing Content, Metadata and User-Created Annotations in Web-Based Applications. In *Proceedings of the 13thACM Symposium on Document Engineering*, 201–204

21. Snodgrass, J.G. 2003. Representations, Abstract and Concrete. In *Encyclopedia of Cognitive Science*, Hrsg. L. Nadel London, UK: Nature Publishing Group

22. Stephens, S. 2008. The Enterprise Semantic Web – Technologies and Application for the Real World. In *The Semantic Web – Real-World Applications from Industry*, Hrsg. J. Cardoso, M. Hepp, M.D. Lytras New York, NY: Springer Science + Business Media

23. Stojanovic, L., N. Stojanovic, und J. Ma. 2007. *On the conceptual tagging: An ontology pruning use case* IEEE/WIC/ACM International Conference of Web Intelligence., 344–350

24. W3C 2014. *Semantic Web – W3C*. http://www.w3.org/standards/semanticweb/. Zugegriffen: 17.03.2014

25. Wittgenstein, L. 2003. *Philosophische Untersuchungen. Auf der Grundlage der Kritisch-genetischen Edition neu herausgegeben von Joachim Schulte*. Frankfurt a. M.: Suhrkamp Verlag

26. Zammuto, R.F., T.L. Griffith, A. Majchrzak, D.J. Dougherty, und S. Faraj. 2007. Information Technology and the Changing Fabric of Organization. *Organization Science* 18(5): 749–762

Linked Data als Grundlage der semantischen Videosuche mit yovisto

13

Harald Sack und Jörg Waitelonis

Zusammenfassung

Videodaten sind auf dem besten Wege zur bedeutendsten Informationsquelle im World Wide Web zu werden. Bereits heute werden pro Minute mehr als 100 Stunden Videomaterial (siehe http://www.youtube.com/yt/press/statistics.html, aufgerufen am 01.03.2014.) von den Benutzern auf Videoplattformen wie YouTube eingestellt. Bei dieser gewaltigen Menge an unstrukturierten multimedialen Daten wird auch die gezielte Informationssuche immer schwieriger, da eine inhaltsbasierte Suche mit Hilfe von textbasierten Metadaten realisiert wird, die entweder manuell oder mittels unzuverlässiger automatischer Analyseverfahren gewonnen werden. Hier bietet die semantische Videosuche einen Ausweg, die aufbauend auf einer Vielzahl unterschiedlicher Analyseverfahren versucht, textbasierte Metadaten inhaltlich miteinander in Bezug zu setzen und zielsicher die gewünschten Ergebnisse zu finden. Darüber hinaus ermöglicht es den zu Grunde liegenden Suchraum, d. h. das gesamte Videoarchiv ähnlich dem Stöbern in einem gutsortierten Bücherregal zielstrebig zu durchmustern und auf diese Weise hilfreiche neue Informationen zu finden. Die Videosuchmaschine yovisto.com implementiert zahlreiche visuelle Analyseverfahren und kombiniert diese prototypisch in einer explorativen semantischen Suche.

H. Sack ✉
Hasso-Plattner-Institut für Softwaresystemtechnik GmbH, Universität Potsdam, 14482 Potsdam, Deutschland
yovisto GmbH, 14482 Potsdam, Deutschland
e-mail: harald.sack@hpi.uni-potsdam.de
J. Waitelonis
yovisto GmbH, 14482 Potsdam, Deutschland

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_13

13.1 Was ist yovisto?

Die yovisto GmbH ist ein Unternehmen mit Sitz in den Filmstudios Babelsberg in Potsdam. Das 2012 gegründete Unternehmen entwickelt und vermarktet semantische Multimedia Technologien. Hierzu zählen automatisierte Verfahren zur inhaltsbasierten Analyse von digitalem Video- und Bildmaterial, sowie darauf abgestimmte Such- und Empfehlungssysteme unter Einsatz von Semantic Web und Linked Data Technologien.

Das Unternehmen yovisto GmbH ist aus der Videosuchmaschine yovisto.com hervorgegangen. Diese existiert seit 2006 und verwaltet aktuell ca. 10.000 Videoaufzeichnungen von universitären Lehrveranstaltungen und wissenschaftlichen Vorträgen weltweit, hauptsächlich in deutscher und englischer Sprache. Was yovisto.com von anderen Videosuchmaschinen unterscheidet, ist der verwendete Suchindex und die in diesem Suchindex verarbeiteten Daten. Traditionelle Suchmaschinen können den tatsächlichen visuellen und auditiven Inhalt der Videodaten nicht problemlos verarbeiten. Deshalb werden üblicherweise nicht die Videodaten selbst, sondern nur deren Metadaten, also die textuellen Beschreibungen zum Inhalt der Videos im Suchindex abgespeichert. Dazu zählen z. B. Titelangaben, Schlüsselwörter, ein kurzer Beschreibungstext und andere, meist durch den Autor zur Verfügung gestellte sogenannte autoritative Information. Nicht immer sind diese Informationen in ausreichender Menge vorhanden oder nicht hinreichend aussagekräftig, um vollständige Suchergebnisse zu gewährleisten. Weiterhin besteht insbesondere bei längeren Aufzeichnungen das Problem, dass sich die angegebenen Metadaten meist auf das gesamte Video und nicht auf einzelne, inhaltlich abgeschlossene Szenen oder fixe Zeitpunkte im Video beziehen. Zur Lösung dieser Probleme setzt die Suchmaschine yovisto.com semantische Multimedia-Technologien ein, mit dem Ziel, neue Metadaten automatisiert zu erzeugen, vorhandene Metadaten zu ergänzen, diese mit anderen automatisiert zu verknüpfen, zu aggregieren und zu verteilen. So werden über eine automatische Verschlagwortung mit Hilfe von Schrifterkennung im Videobild (Video OCR, Optical Character Recognition) Metadaten mit Zeitbezug erzeugt. Bei einer Suchanfrage werden auf diese Weise auch einzelne Zeitpunkte im Video hervorgehoben, und das Video kann direkt an der Fundstelle wiedergegeben werden. Um die Qualität der erzielten Suchergebnisse weiter zu erhöhen, werden die meist textbasierten Metadaten mit öffentlichen verfügbaren Wissensbasen, wie z. B. DBpedia[1] verknüpft. Auf diese Weise werden Suchanfragen mit Zusatzinformationen erweitert, um genauere und vollständigere Suchergebnisse zu erzielen oder Zusammenhänge zwischen diesen aufzuzeigen.

Der Einsatz dieser semantischen Technologien führt zu einem erheblichen Mehrwert für den Benutzer. Die automatische Erzeugung neuer zeitabhängiger Metadaten aus den Videodaten ermöglicht qualitativ bessere Suchergebnisse, zeitbezogenes punktgenaues Suchen und damit eine höhere Granularität der erzielten Suchergebnisse sowie neue, zusätzliche Navigationsmöglichkeiten. Die Verknüpfung der Metadaten mit semantischen Wissensrepräsentationen erlaubt die Nutzung semantischer Zusammenhänge bei der Su-

[1] Siehe http://dbpedia.org/, aufgerufen am 01.03.2014.

che. Die Darstellung dieser semantischen Zusammenhänge bei der Visualisierung von Suchergebnissen führt zu neuen Möglichkeiten in der Navigation. Das Anreichern und Ergänzen der eigenen Inhalte mit den verknüpften Daten erhöht deren Vielfalt und damit auch das Potenzial neue Nutzer zu gewinnen. Darüber hinaus wird das Risiko einer Nutzerabwanderung verringert, da die Informationsbedürfnisse besser bedient werden können. Weiterhin können inhaltsbasierte Empfehlungen innerhalb der Suchmaschine (z. B. zu anderen Videos) oder auch zu externen Inhalten (z. B. zu Büchern oder anderen Produkten) einbezogen werden. Darüber hinaus basiert die Suchmaschine auf Linked Data Prinzipien und stellt so eine universelle Schnittstelle zu anderen Systemen und Kunden zur Verfügung.

In den folgenden Abschnitten sollen diese Aspekte technisch detailliert beleuchtet und deren Nutzen sowie Grenzen aufgezeigt werden.

13.2 Schlüsselwortbasierte Suche vs. semantische Suche – ein konzeptioneller Vergleich

Multimediale Informationsressourcen und insbesondere Videodaten bestimmen heute das Erscheinungsbild des World Wide Webs (WWW). Dabei sind Suchmaschinen bei der Suche in multimedialen Daten heute immer noch auf textbasierte Metadaten angewiesen, die aufwändig durch einen menschlichen Bearbeiter manuell annotiert oder direkt über Analyseverfahren aus den Originaldaten gewonnen werden und gemeinsam mit den Mediendaten vorliegen müssen. Die automatische inhaltliche Analyse von multimedialen Dokumenten stößt aktuell allerdings schnell an die Grenzen ihrer Leistungsfähigkeit. Noch immer ist das bild- und informationsverarbeitende System der menschlichen Wahrnehmung jeder automatisierten Analyse überlegen, insbesondere dann, wenn komplexe Transfer- und Abstraktionsleistungen zur Interpretation des Inhalts notwendig werden. Daher werden die zur Informationssuche in Videodaten benötigten Metadaten heute in den meisten Fällen manuell gewonnen. Einfacher ist die Ermittlung von Videometadaten im WWW, das als Netzwerk untereinander verbundener Dokumente einen sogenannten Link-Kontext vorhält, z. B. Text, der im Kontext eines Hyperlinks auf das betreffende Videodokument vorliegt und dieses oft inhaltlich beschreibt. Bevor auf die besonderen Eigenheiten der Suche in Videodaten eingegangen wird, werden zunächst allgemein die traditionelle, schlüsselwortbasierte Suche und die semantische Suche einander gegenübergestellt.

13.2.1 Schlüsselwortbasierte Suche und Volltextsuche

Unabhängig von der Art der Dokumente oder Medien, die durchsucht werden sollen, besteht die grundlegende Aufgabe im Information Retrieval in der Ermittlung von Informationsressourcen, die bezogen auf eine Suchanfrage als relevant eingestuft werden. Dazu

müssen die betreffenden Informationsressourcen zunächst indexiert werden, d. h. sie werden in eine abstrakte Repräsentationsform gebracht, in der eine Ähnlichkeitsbestimmung zwischen den durchsuchten Dokumenten und der Suchanfrage das Suchergebnis liefert. Um die Qualität der erzielten Suchergebnisse zu beurteilen, verwendet man einfache statistische Maßzahlen, die angeben, wie genau (Precision) und wie vollständig (Recall) die gewonnenen Suchergebnisse bzgl. einer vorgegebenen Referenz sind. Handelt es sich bei den betrachteten Informationsressourcen um Textdokumente, lassen sich auf einfache Weise Deskriptoren (Metadaten) bestimmen, indem eine aussagekräftige Untermenge der darin enthaltenen Terme mit geeigneten linguistischen Verfahren ausgewählt wird. Zu diesen zählen das Zerlegen von Fließtext in einzelne Terme (Tokenisierung), die Bestimmung der Wortart (Part-of-Speech Tagging) oder auch die Rückführung von Termen in eine Stamm- oder Normalform (Stemming). Im Gegensatz zum textbasierten Information Retrieval ist die Ableitung inhaltlich aussagekräftiger Deskriptoren aus multimedialen Daten erheblich aufwändiger.

Google als prominentester Vertreter einer WWW-Suchmaschine steht heute vielfach in der Kritik, da Suchanfragen oft zu viele nicht relevante Ergebnisse zurückliefern. Einer der Gründe dafür liegt in der Mehrdeutigkeit jeder natürlichen Sprache. Zahlreiche Terme stehen für eine Vielzahl unterschiedlicher Bedeutungen (Homonymie). So bezeichnet der Term „Bank“ sowohl ein Geldinstitut, ein Sitzmöbel, eine Untiefe im Meer, kann aber auch für den Nachnamen einer Person stehen. Eine traditionelle schlüsselwortbasierte Suche nach dem Begriff „Bank“ resultiert in Ergebnisdokumenten, in denen das Wort „Bank“ in unterschiedlichen Bedeutungen verwendet wird, von denen nicht alle der vom Benutzer beabsichtigten Bedeutung entsprechen. Aber selbst wenn die eindeutige Bedeutung erschlossen werden könnte, so kann ein Wort auch in unterschiedlichem Kontext und mit unterschiedlicher Absicht (Pragmatik) vom Autor des Dokuments verwendet worden sein, die mit den Absichten des Benutzers, der die Suchanfrage stellt, nicht übereinstimmen muss.

Andererseits kann heute kein Mensch mehr beurteilen, ob sich auch tatsächlich alle relevanten Ergebnisse in der Fülle der von Google angebotenen Suchtreffer befinden. So können auch alternative Bezeichner (Synonyme) in einer traditionellen Volltextsuche nicht ermittelt werden. Sucht der Benutzer nach dem Term „Bank“, werden Dokumente, die den Begriff „Bank“ über das Synonym „Kreditanstalt“ verwenden, nicht als Ergebnis ermittelt. In gleicher Weise werden taxonomische Zusammenhänge, d. h. Ober- und Unterbegriffe, Generalisierungen und Spezialisierungen von Begriffen nicht in die Volltextsuche mit einbezogen. Wird nach dem Oberbegriff „Sportler“ gesucht, werden Dokumente, die lediglich eine Spezialisierung des Begriffes enthalten, wie z. B. „Fußballspieler“, nicht als Ergebnis ermittelt. Daneben können Begriffe auch mit Hilfe von Metaphern und anderen sprachlichen Ausdrucksmitteln umschrieben werden, ohne dass der inhaltlich damit bezeichnete Suchbegriff in einem an sich relevanten Dokument auftauchen muss. Daher liegt der Schluss nahe, dass vielmehr die inhaltliche Bedeutung (Semantik) der Bestandteile eines Dokuments und nicht nur die darin verwendeten Zeichenketten im Vordergrund einer inhaltsbasierten Suche stehen müssen [1].

13.2.2 Semantische Suche

Explizite formale Semantik in Form von W3C[2]-Standard konformen semantischen Metadaten (URI[3], RDF[4], OWL[5], SPARQL[6], RIF[7], etc.) können zur Verbesserung des inhaltsbasierten Information Retrievals im Sinne einer semantischen Suche herangezogen werden. Dabei können wir prinzipiell die Verwendung von Wissensrepräsentationen (Ontologien) im Sinne von Klassen, Beziehungen zwischen Klassen (Relationen), sowie Einschränkungen, Bedingungen und Regeln, die an Klassen und Relationen geknüpft sind, unterscheiden von der Nutzung einzelner Instanzen (Entitäten) dieser Ontologien in der Form von Linked Open Data Ressourcen. Diese semantischen Metadaten können das klassische Information Retrieval auf die folgende Weise unterstützen:

- **Sinnvolle und zielgerichtete Präzisierung und Erweiterung von Suchergebnissen (Query String Extension oder Query String Refinement)**
  Die Erweiterung von Suchphrasen wird von WWW-Suchmaschinen wie Google bereits seit geraumer Zeit angeboten. Dabei werden anhand vorhandener Nutzungsdaten populäre Suchphrasen ermittelt, deren Präfix mit der vom Benutzer eingegebenen Suchphrase übereinstimmt (Autocompletion). In gleicher Weise kann eine semantische Erweiterung im Sinne einer Disambiguierung mehrdeutiger Suchphrasen angeboten werden, d. h. bei der Eingabe der Suchphrase „Bank“ werden zur Vervollständigung die Varianten „Bank, Geldinstitut“, „Bank, Sitzmöbel“, „Bank, Meeresuntiefe“, usw. als semantische Präzisierung gemeinsam mit möglichen syntaktischen Vervollständigungen (z. B. Komposita, wie „Bankräuber“, „Bankschalter“, usw.) und deren möglichen Disambiguierungen angegeben. Dabei stellt die Wahl einer geeigneten Benutzerschnittstelle eine kritische Komponente der semantischen Suchmaschine dar, auf die in Abschn. 13.5 näher eingegangen wird.
- **Herleitung von implizit vorhandener, verdeckter Information (Inference)**
  Durch die Ergänzung der ursprünglichen Suchphrase mit Termen, die aus relevanten Ontologien stammen, wird nicht nur eine zielgenauere Suche ermöglicht, sondern vielmehr auch eine assoziativ motivierte Suche, die anhand impliziter Zusammenhänge Naheliegendes erschließt und aufdeckt, und so dem Suchenden einen Einblick in vorhandene Informationen gewährt, die er über eine traditionelle Informationssuche nie entdeckt hätte. Bezieht man z. B. Ober- und Unterbegriffe, d. h. Spezialisierungen und Generalisierungen in die Suche mit ein, so ist es sinnvoll, bei der Suche nach einem

[2] Siehe World Wide Web Consortium, http://www.w3.org/, aufgerufen am 01.03.2014.
[3] Siehe Uniform Resource Identifier, http://tools.ietf.org/html/rfc3986, aufgerufen am 01.03.2014.
[4] Siehe Resource Description Framework, http://www.w3.org/TR/rdf-concepts/, aufgerufen am 01.03.2014.
[5] Siehe Web Ontology Language, http://www.w3.org/TR/owl-ref/, aufgerufen am 01.03.2014.
[6] Siehe SPARQL Query Language, http://www.w3.org/TR/sparql11-query/, aufgerufen am 01.03.2014.
[7] Siehe Rule Interchange Format, http://www.w3.org/TR/rif-core/, aufgerufen am 01.03.2014.

Oberbegriff auch Ergebnisse zu dessen Unterbegriffen oder Spezialisierungen zurückzuliefern. So könnten etwa bei einer Suche nach dem Oberbegriff „Sportler“ auch Ergebnisse zurückgeliefert werden, die lediglich die Begriffe „Fußballer“, „Tennisspieler“ oder „Rennfahrer“ beinhalten, da es sich bei diesen um spezielle Varianten (genauer Unterklassen) von Sportlern handelt. Ebenso könnten explizit genannte Individuen, von denen bekannt ist, dass diese zu „Fußballer“, „Tennisspieler“ oder „Rennfahrer“ zählen, bei der Suche nach dem Oberbegriff „Sportler“ zurückgeliefert werden. Auf diese Weise kann die Einbeziehung einer entsprechenden Ontologie in die Suche die Vollständigkeit der zurückgelieferten Suchergebnisse erhöhen [2].

- **Herstellung von Querverweisen und Assoziationen (Cross Referencing)**
  Ging es bislang um die Ergänzung bzw. Präzisierung von Suchergebnissen, können über die Einbeziehung von Querverweisen und Assoziationen zusätzlich zu den gewünschten Suchergebnissen auch inhaltlich naheliegende, aber nicht explizit verlangte Suchergebnisse zurückgeliefert werden. Dabei sind nicht notwendigerweise nur zur Suchabfrage „ähnliche“ Dokumente gemeint, sondern Dokumente, die mit der ursprünglichen Suchphrase in einem engen inhaltlichen Zusammenhang stehen. Sucht der Benutzer nach „Albert Einstein“, dann ist er eventuell auch an dessen Forschungsthemen „Relativitätstheorie“ oder „Quantentheorie“ interessiert. Um dies zu implementieren, werden entsprechend umfassende Wissensbasen benötigt, z. B. DBpedia.org. Naheliegendes oder Ähnliches als Suchergebnis zu liefern, ist insbesondere auch dann interessant, wenn in einem inhaltlich beschränkten Suchraum, wie z. B. in einem Videoarchiv gesucht wird. Sollte zu einer Suchphrase kein exakt passendes Ergebnisdokument geliefert werden können, so könnten für den Benutzer zumindest inhaltlich naheliegende Dokumente von Interesse sein. Hier verläuft die Grenze zu einem inhaltsbasierten Empfehlungssystem bzw. einer explorativen Suche fließend.
- **Nutzung von semantischen Beziehungen zur Visualisierung und Navigation durch den Such- oder Ergebnisraum der Suche (Explorative Suche)**
  Bei der Suche in einem Bibliothekskatalog sucht der Benutzer nach Autoren, Titeln, Verlagen oder Schlagworten. Entweder kennt der Benutzer den Namen des betreffenden Autors bzw. den Buchtitel, oder aber er versucht sich an einer thematischen Zuordnung des von ihm gesuchten Werkes und schlägt diese im Schlagwortkatalog nach, in dem den Informationsressourcen von autoritativer Stelle, d. h. vom Autor, dem Verleger oder dem Bibliothekar passende Schlagwörter zugeordnet wurden. Da aber Benutzer und Schlagwortautor unterschiedlicher Auffassung über die treffende Zuordnung von Schlagwörtern sein können, ist diese Variante der Suche nicht immer zielführend. Anders ist die Situation, wenn der Benutzer nicht genau weiß, was er sucht. Wenn er sich z. B. erst einmal einen Überblick über die zu einem Themengebiet vorhandenen Informationsressourcen bzw. über den gesamten Suchraum verschaffen möchte. Im WWW ist dies heute schon aufgrund der ungeheuren Dokumentenmenge unmöglich. In der Bibliothek dagegen hat der Benutzer die Möglichkeit, die Bücherregale selbst zu durchforsten, in denen die vorhandenen Informationsressourcen entsprechend einer vorgegebenen Systematik eingeordnet wurden. So kann er innerhalb eines The-

mengebiets „herumstöbern“ und dabei zufällig auf Bücher stoßen, die ihn interessieren, wobei er sich dessen zuvor eventuell gar nicht bewusst war. Diese Möglichkeit der „zufälligen und glücklichen Entdeckung“ wird im Englischen auch als „Serendipity“ bezeichnet. Es geht also darum, Suchergebnisse zu entdecken, nach denen der Benutzer zunächst gar nicht gesucht hatte. Diese Art der zielgerichteten Erkundung des Suchraums ist uns aus unserem täglichen Leben vertraut und wird auch als „explorative Suche“ bezeichnet, die sich mit Hilfe von Domain-Ontologien und Linked Data Ressourcen realisieren lässt.

13.3 Semantische Videosuche

Im vorliegenden Kapitel soll zunächst auf die Eigenheiten der Suche in Videodaten eingegangen werden. Traditionelle Metadaten können entweder manuell oder mit Hilfe automatisierter Analyseverfahren gewonnen werden (vgl. Abschn. 13.3.1). Aufbauend auf diesen traditionellen, meist textbasierten Metadaten kann eine semantische Analyse durchgeführt werden mit dem Ziel, Entitäten in den Textdaten zu identifizieren und diese mit Entitäten aus einer Wissensbasis oder Klassen einer Ontologie zu annotieren (vgl. Abschn. 13.3.2). Die so gewonnenen semantischen Metadaten dienen in erster Linie dazu, die Qualität der erzielten Suchergebnisse zu verbessern. Des Weiteren können sie dazu eingesetzt werden, die Suchergebnisse inhaltlich zu strukturieren (facettierte Suche) und den Benutzer gezielt durch den Suchraum zu führen (explorative Suche, vgl. Abschn. 13.3.3).

13.3.1 Videoanalyse und traditionelle Metadaten

Inhaltsbeschreibende Metadaten zu Videodaten können entweder aufwändig manuell annotiert werden oder mit Hilfe automatischer Analyseverfahren ermittelt werden. Man unterscheidet bei den durch automatische Analyseverfahren gewonnenen Metadaten sogenannte Low-Level Deskriptoren von High-Level Deskriptoren, die sich durch die Höhe ihres Abstraktionsniveaus unterscheiden. Low-Level Deskriptoren lassen sich direkt aus statistischen Analysen der Videodaten gewinnen, wie z. B. Aussagen über die Farbverteilung von Einzelbildern, der Verlauf der Lautstärke einer Audiosequenz oder die Helligkeitsdifferenzen einer Serie aufeinanderfolgender Einzelbilder. Diese Low-Level Deskriptoren erlauben zunächst keine Aussagen über den Inhalt der betrachteten Videodaten. Tatsächlich eignen sie sich gut für eine ähnlichkeitsbasierte Suche, bei der z. B. Bilddateien gefunden werden sollen, deren Inhalt nach visuellen Kriterien gemessen einem vorgegebenen Bild ähneln. Im Gegensatz dazu besitzen High-Level Deskriptoren ein höheres Abstraktionsniveau und repräsentieren nicht direkt visuelle oder auditive Parameter, sondern vielmehr deren inhaltliche Interpretation. Eine inhaltsbasierte Suche lässt sich in Videodaten daher am besten über High-Level Deskriptoren realisieren.

Im Folgenden sollen einige automatische Analyseverfahren kurz vorgestellt werden, die in yovisto.com zum Einsatz gelangen:

Strukturelle Videoanalyse

Die strukturelle Analyse der Videodaten ist der erste Analyseschritt für die inhaltsbasierte Videosuche. Dabei ist es das Ziel der strukturellen Analyse, das Video in inhaltlich zusammenhängende, d. h. kohärente Abschnitte zu zerlegen. Diese Segmentierung der Videodaten liefert wichtige Informationen über die zeitlichen Eigenschaften des Videos, die in daran anschließenden Analyseschritten, aber auch bei der Darstellung von Navigationselementen für den Benutzer Verwendung finden. So kann auf der Grundlage der Videosegmentierung ein visuelles Inhaltsverzeichnis eines Videos angelegt werden, mit dessen Hilfe der Benutzer im Video navigieren kann. Zu diesem Zweck müssen geeignete Einzelbilder, sogenannte Key-Frames ermittelt werden, die einzelne Szenen am besten visuell beschreiben. Andererseits müssen aus jedem Segment auch geeignete Einzelbilder für die spätere Schrifterkennung (OCR) und die visuelle Konzepterkennung (VCD) selektiert werden. Neben den visuellen Eigenschaften können für eine strukturelle Analyse auch die auditiven Eigenschaften eingesetzt werden, z. B. um Sprecher, Sprecherwechsel, Musikeinblendungen und andere signifikante Ereignisse (z. B. Applaus, Schüsse, Schreie, etc.) zu markieren. Darüber hinaus ist die Unterteilung der Videodaten in zusammenhängende Szenen eine wichtige Voraussetzung für die Festlegung des semantischen Kontextes in der anschließenden semantischen Analyse.

Ein Videodatenstrom lässt sich noch weiter strukturell unterteilen. Jede zusammenhängende Szene kann zusätzlich in sogenannte Shots unterteilt werden. Ein Shot markiert den Beginn und Ende einer zusammenhängenden Kameraaufnahme. Zwischen zwei Shots liegt stets ein Schnitt (Cut). Man unterscheidet hier grundsätzlich zwischen abrupten Szenenwechseln (harter Schnitt, Hard-Cut) oder graduellen Übergängen (Soft-Cuts), wie z. B. langsame Ein-/Ausblendungen (Fade-In/Out), das „Einschieben“ des neuen Shots (Wipe) oder die Überblendung (Dissolve). Innerhalb eines Shots lassen sich Sub-Shots ermitteln, die z. B. das Auftauchen und Verschwinden von Objekten innerhalb des Shots markieren. Sub-Shots selbst bestehen aus einer Folge von Einzelbildern (Frames). Abbildung 13.1 gibt einen Überblick über die hierarchische Struktur von Videodaten.

Die Erkennung von harten Schnitten basiert meist auf statistischen Methoden, die Helligkeitsdifferenzen mehrerer aufeinanderfolgender Einzelbilder analysieren und dazu einen adaptiven Schwellenwert bestimmen. Überschreitet die Helligkeitsdifferenz aufeinanderfolgender Einzelbilder diesen Schwellenwert, kann ein harter Schnitt identifiziert werden. Die graduellen Übergänge Fade-In/Out weisen häufig einen langsamen An- oder Abstieg der Beleuchtungsintensität und damit verbunden der Informationsdichte (Entropie) im Bild auf, wenn z. B. zu einem schwarzen oder weißen Bild überblendet wird.

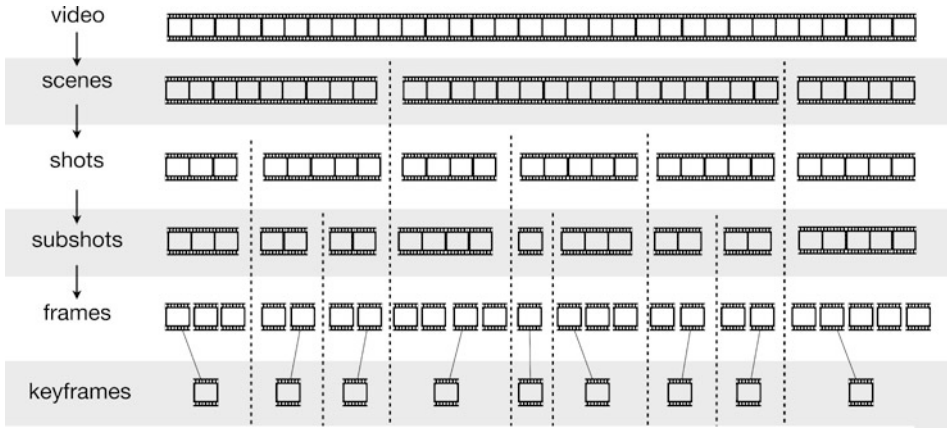


Abb. 13.1 Hierarchische Struktur von Videodaten

Video OCR

Zu jeder erkannten Bildsequenz (Szene) werden repräsentative Einzelbilder (Key-Frames) bestimmt, die den Inhalt der betreffenden Sequenz möglichst gut repräsentieren. So werden z. B. universitäre Vorlesungen und wissenschaftliche Vorträge heute meist von textbasierten Präsentationen (Folien, Desktop-Präsentation, Tafelanschrieb, etc.) unterstützt, in denen die inhaltlich wichtigsten Punkte zusammengefasst werden. Diese Texte werden im Videobild zunächst lokalisiert, anschließend aufwändig vorverarbeitet und mit Hilfe geeigneter Texterkennungsmethoden als textuelle Metadaten extrahiert (Video Optical Character Recognition, OCR). Da nicht jedes einzelne Videobild bei einer Bildfrequenz von aktuell mindestens 25 Bilder/Sekunde aufgrund der Komplexität der Verarbeitung vollständig analysiert werden kann, werden die Einzelbilder zunächst über Kantenerkennungsverfahren in Verbindung mit einer statistischen Analyse der Kantenrichtungsverteilungen im Bild vorgefiltert. Dabei werden die statistischen Eigenschaften typischer Schriften herangezogen, um Schriftkandidaten unter den Videobildern zu identifizieren und herauszufiltern. Diese Schriftkandidaten durchlaufen ein von der Berechnungskomplexität aufwändigeres Verifikationsverfahren, bei dem die verwendete Strichweite der entdeckten Kanten ermittelt und im Bild mit typischen Strichweiten von Schriften abgeglichen werden (Stroke Width Detection). Die so verifizierten Schriftkandidaten werden räumlich zusammengefasst (Connected Component Analysis) und durchlaufen weitere Vorverarbeitungsschritte, deren Ziel es ist, die Schrift vom darunterliegenden Videobild zu separieren [3]. Eingabe für die anschließende Text-OCR ist stets ein vorgefilterter, normalisierter Textkandidat mit schwarzer Schrift auf weißem Hintergrund. Dies ermöglicht anschließend den Einsatz handelsüblicher Text-OCR Software, deren Entwicklung

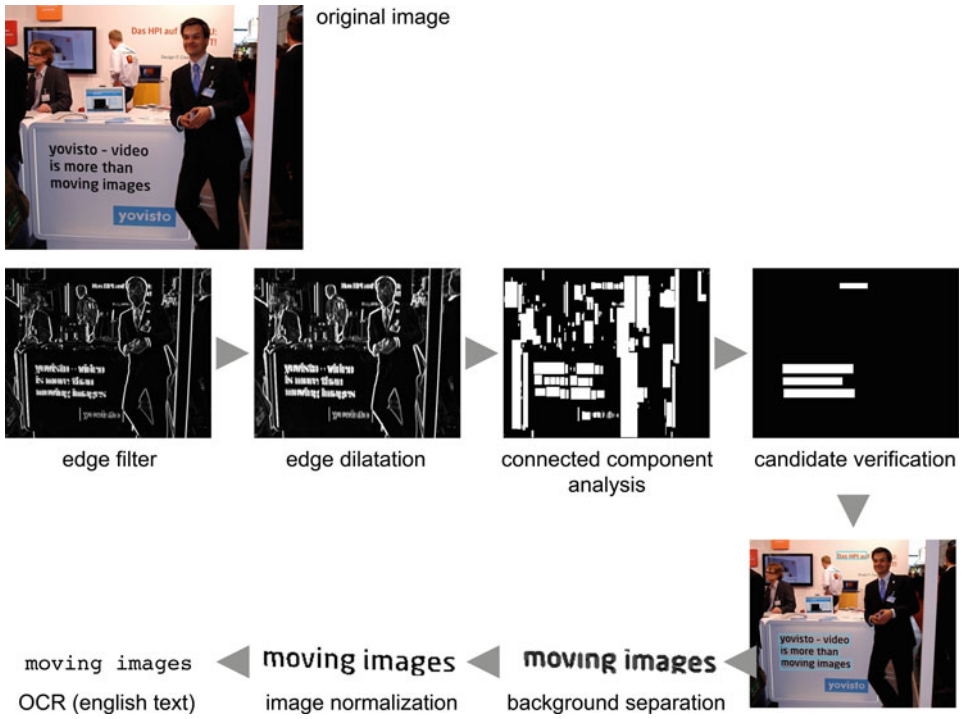


Abb. 13.2 Prozess-Einzelschritte der Video-OCR

mittlerweile ausgereift ist und die zuverlässige Texterkennungsergebnisse gestattet. Allerdings arbeitet die Text-OCR üblicherweise auf qualitativ hochwertigen Scans, deren Texte in schwarzer Schrift auf einheitlich weißem Hintergrund vorliegen. Bei der Video-OCR ist das Ausgangsbild oft von niedrigerer Qualität, d. h. die Auflösung ist geringer, der Hintergrund ist heterogen, schwacher Kontrast und differenzierte Beleuchtungsverhältnisse erschweren zusätzlich die Texterkennung. Sollen neben Texteinblendungen (Overlay Text) auch von der Kamera aufgenommene Texte (Szenentext) erkannt werden, kommt zusätzlich noch das Problem der geometrischen Verzerrung der Texte, sowie Verdeckungen und Abschattungen der Texte zum Tragen. Um dennoch eine zufriedenstellende Texterkennung zu gewährleisten, durchlaufen die von der OCR-Software ermittelten Texte aufwändige syntaktische und semantische Fehlerkorrekturverfahren. Abbildung 13.2 zeigt vereinfacht die einzelnen Prozessschritte der Video-OCR.

Visual Concept Detection

Neben den textbasierten Metadaten bilden auch die visuellen Inhalte in Bild- oder Videodaten eine wichtige Informationsquelle. In der visuellen Konzepterkennung (Visual Concept Detection) werden Einzelbildern automatisch vordefinierte inhaltliche Kategorien zugeordnet, wie z. B. Innen- oder Außenaufnahme, Einzelpersonen oder Menschengrup-

Abb. 13.3 Beispiel für die visuelle Konzepterkennung

pen, Portrait-, Landschafts-, oder Stadtaufnahme, Nahaufnahme, Tiere, Pflanzen, Grafiken, Himmel, Wolken, Wasser, Berge, etc. Abbildung 13.3 zeigt ein Bildbeispiel sowie das zugehörige Klassifikationsergebnis für sieben allgemeine Bildkategorien, wobei für jede Kategorie auch die ermittelte Zuverlässigkeit mit angegeben wird.

Die visuelle Konzepterkennung erfolgt prinzipiell über die folgenden Einzelschritte: Im ersten Schritt der Bildanalyse werden Deskriptoren (Feature-Vektoren) aus dem Einzelbild extrahiert (vgl. Abb. 13.4a). Diese beschreiben die visuellen Basiseigenschaften (Low-Level Features) des Bildes bzgl. Kanten-, Kontrast- und Farbverteilungen (vgl. Abb. 13.4b). Zu den bekanntesten Algorithmen, die zu diesem Zweck eingesetzt werden, zählt die sogenannte skaleninvariante Merkmalstransformation zur Extraktion lokaler Bildmerkmale (Scale-invariant feature transform, SIFT) [5]. Sie wird auf jeden der einzelnen Farbkanäle des Bildes (RGB) angewandt. Die extrahierten Werte werden anschließend durch die sogenannte Bag-Of-Keypoints Methode [6] zu Vektoren zusammengefasst (vgl. Abb. 13.4c). Da für ein Einzelbild sehr viele dieser Vektoren anfallen und nicht alle gleichermaßen stark die Eigenschaften des Bildes beschreiben, wird im nächsten Schritt ein k-Means Clustering Verfahren ausgeführt, das die Anzahl der Vektoren reduziert. Das Ergebnis ist ein visuelles Wörterbuch (Codebuch), das die wichtigsten Repräsentanten der Vektoren (Codewörter) enthält (vgl. Abb. 13.4d). Jeder Vektor wird dem nächst gelegenen Codewort zugeordnet, die Codewörter werden bzgl. ihrer Häufigkeit gezählt. Auf diese Weise lässt sich eine Häufigkeitsverteilung (Histogramm) der Codewörter ermitteln, die als Deskriptor für das Bild verwendet wird (vgl. Abb. 13.4e).

Zur eigentlichen Klassifikation der Deskriptoren werden Verfahren des maschinellen Lernens eingesetzt. Diese Verfahren bestehen im Allgemeinen aus zwei Phasen, einer Lern- und einer Klassifikationsphase. Vor der Lernphase wird für eine bestimmte Klasse ein Trainingsdatensatz manuell zusammengestellt. Dieser besteht aus zahlreichen Bildern, die Positiv- und Negativbeispiele für die gewünschte Klasse visueller Inhalte darstellen. Während der Lernphase wird aus dem Trainingsdatensatz ein Modell berechnet, mit des-

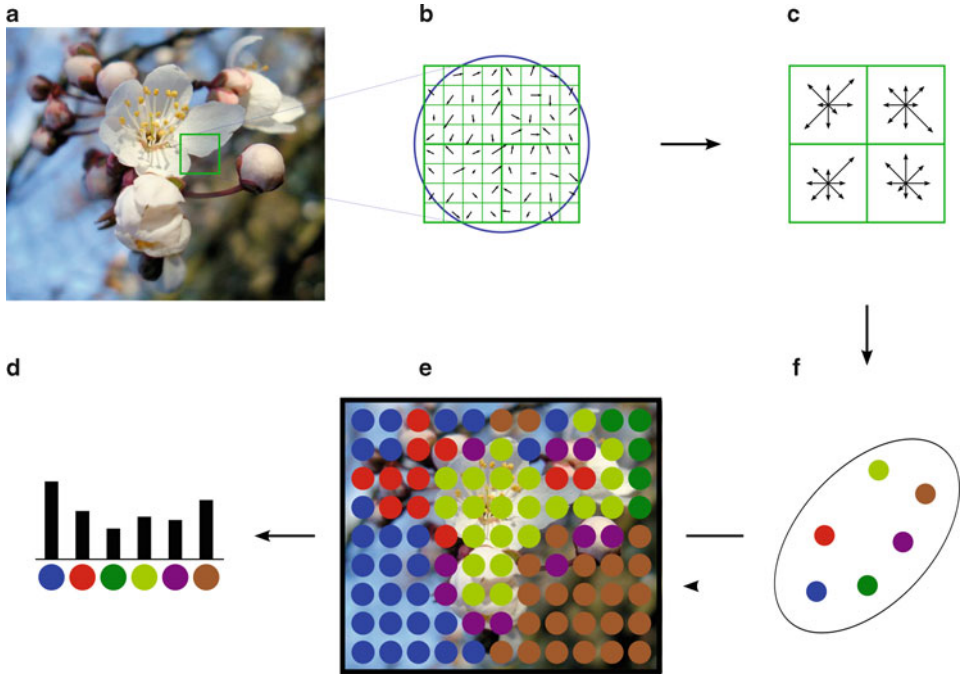


Abb. 13.4 Ablauf der Merkmalsextraktion in der visuellen Konzepterkennung [6]

sen Hilfe bei der Klassifikationsphase entschieden werden kann, ob ein neues bis dahin noch unbekanntes Bild dieser Klasse zugeordnet werden kann oder nicht. Dieses Modell wird als Klassifikator bezeichnet.

In der Visual Concept Detection werden zur Berechnung der Klassifikatoren sogenannte Support Vektor Maschinen (SVM) eingesetzt [7, 8, 9]. Dies sind keine Maschinen im mechanischen Sinne, sondern mathematische Berechnungsvorschriften. Für die Klassifikation von Histogramm-Daten haben sich sogenannte Gauss-Kernel-basierte SVMs als besonders geeignet herausgestellt [9]. Das Ergebnis der Klassifikation ist meist ein Zahlenwert zwischen 0 und 1. Bei Erreichen oder Unterschreiten eines bestimmten Schwellenwerts im Klassifikationsergebnis wird festgelegt, ob das getestete Bild nun zur Klasse gehört oder nicht, bzw. mit welcher Zuverlässigkeit eine solche Aussage getroffen werden kann (vgl. Abb. 13.3).

Der besondere Vorteil des gesamten hier vorgestellten Verfahrens liegt in seiner Robustheit gegenüber Bildverzerrungen, Beleuchtungsunterschieden, Rotation, Verdeckung und erhöhter Varianz innerhalb einzelner Klassen [7]. Der Ansatz ist generisch anwendbar, d. h. neue Klassen können beliebig trainiert werden, ohne dass die Art und Weise der Featureextraktion an die neue Klasse angepasst werden muss. Eine ausreichend große Anzahl an Beispielbildern, die die neue Klasse hinreichend repräsentieren, genügt.

Audioanalyse

Zusätzlich kann eine Spracherkennung (Automated Speech Recognition, ASR) verwendet werden, die eine (oft fehlerbehaftete) Transkription der gesprochenen Inhalte einer Videosequenz ermöglicht. Aufgrund der meist schlechten Aufnahmebedingungen (kein professionell ausgebildeter Sprecher, keine Studiobedingungen, Störgeräusche durch das Publikum, etc.) können die via ASR erkannten Texte zahlreiche Fehler enthalten und sind daher qualitativ im Vergleich zu den mittels OCR erzielten Metadaten nebenrangig. Man muss bei der Spracherkennung zwischen Systemen unterscheiden, die speziell auf einen bestimmten Sprecher und dessen Aussprache hin trainiert wurden, und Systemen, die ein allgemeines Sprachmodell anwenden, um sprecherunabhängig arbeiten zu können. Allerdings ist die Qualität der Erkennungsergebnisse sprecherunabhängiger Systeme den speziell trainierten Systemen unterlegen und weniger robust gegenüber Störgeräuschen und mangelnder Aufnahmequalität. Yovisto verwendet für die Audioanalyse der Videodaten Systeme von Drittanbietern, wie z. B. vetail-X[8].

13.3.2 Semantische Analyse

Die in der Videoanalyse gewonnenen Metadaten liegen meist als textbasierte Metadaten vor und können für eine traditionelle schlüsselwortbasierte Suche verwendet werden. Um eine semantische Suche auf den Videodaten durchführen zu können, müssen diese Metadaten zunächst eine weitere, semantische Analyse durchlaufen, bevor die in Abschn. 13.2.2 genannten Möglichkeiten der semantischen Suche realisiert werden können. Die semantische Analyse textbasierter Metadaten beginnt grundsätzlich mit einer linguistischen Analyse, wie sie auch im traditionellen Information Retrieval Anwendung findet. Natürlichsprachlicher Text wird in seine Wortbestandteile aufgelöst (Tokenization). Stoppwörter, die für eine Suche nichtrelevante Information enthalten, werden entfernt (Stopword Removal, Blacklisting). Zuvor werden noch die einzelnen Wörter im Satz bzgl. ihrer Funktion gekennzeichnet (Part-of-Speech Tagging), um die für die Suche relevanten Hauptwörter bzw. Adjektiv-Hauptwort-Kombinationen herauszufiltern. Danach findet oft noch eine Rückführung auf Wortstammformen statt, damit auch alle möglichen Flexionsformen der Wörter berücksichtigt werden können (Stemming).

In natürlichsprachlichem Text werden dann bedeutungstragende Entitäten ermittelt (Named Entity Recognition), d. h. es wird ermittelt, welche Wörter im Satz Personen, Orte oder andersartige Entitäten repräsentieren. Eine Stufe weiter geht die sogenannte Named Entity Disambiguation. Dabei werden die ermittelten Entitäten den Instanzen einer vorgegebenen Wissensbasis, also z. B. einer DBpedia Entität eindeutig zugeordnet. Beide Verfahren werden nachfolgend kurz vorgestellt:

[8] Siehe http://www.vetail-x.com/, aufgerufen am 01.03.2014.

Named Entity Recognition (NER)

Das Ziel der Named Entity Recognition besteht in der Zuweisung vordefinierter Klassen und Kategorien zu bedeutungstragenden Entitäten (Named Entities) in einem Text. Üblicherweise sind dabei Klassen wie z. B. Personen, Orte, Organisationen oder Datumsangaben von Interesse, wobei die Klassifikation auch wesentlich differenzierter erfolgen kann. Die hierzu eingesetzten Verfahren machen sich statistische Eigenschaften in natürlichsprachigen Texten zu Nutze, indem die lokale Satzstruktur zur Klassifikationsentscheidung herangezogen wird. Neben einfachen regelbasierten Verfahren basieren aktuelle Ansätze auf *Hidden Markov Models* oder auch *Conditional Random Fields*, die aufgrund lokaler Satzeigenschaften Entscheidungen treffen. Es werden aber auch nicht-lokale Ansätze verfolgt, die semantische Abhängigkeiten über größere Distanzen in Texten verfolgen [10]. Zur Durchführung der NER stehen bereits verschiedene, öffentlich verfügbare Bibliotheken, wie z. B. der Stanford Named Entity Recognizer[9] kostenlos zur Verfügung. In yovisto dient die Named Entity Recognition als Unterstützung zur anschließenden Named Entity Disambiguation, in der einem Textabschnitt eine spezifische semantische Entität aus einer Wissensbasis zugewiesen wird.

Named Entity Disambiguation

Das Ziel der Named Entity Disambiguation (auch Word Sense Disambiguation) besteht in der korrekten eindeutigen Zuweisung einer bedeutungstragenden Entität aus einer Wissensbasis zu einem Textabschnitt, der diese Entität inhaltlich referenziert, z. B. die Bedeutung eines Wortes innerhalb eines gegebenen Kontexts richtig zu bestimmen. Die Schwierigkeit bei dieser Aufgabe liegt in der Mehrdeutigkeit natürlicher Sprache begründet. Die Bedeutung eines Wortes ist stets gebunden an einen Kontext, in dessen Zusammenhang das Wort verwendet wird. Mehrdeutige Wörter, sogenannte Homonyme, können entsprechend dem Zusammenhang ihrer Verwendung unterschiedliche Bedeutungen haben. Die „Bank“ kann sowohl das Geldinstitut als auch die Sitzgelegenheit bezeichnen. In einem Satz wie z. B. „Herr Schmidt sitzt auf der Bank“ ist mit hoher Wahrscheinlichkeit das Sitzmöbel und nicht das Geldinstitut gemeint, auch wenn man im gegebenen Kontext nicht absolut sicher sein kann. Je mehr Kontext gegeben ist, d. h. wenn mehr Text gegeben ist, z. B. „Herr Schmidt sitzt auf der Bank. Die Bank steht unter den Bäumen im Park.“ wächst die Zuverlässigkeit, mit der eine eindeutige Aussage bzgl. der Bedeutung des Wortes „Bank“ ermittelt werden kann. Während sich der Kontext in einem natürlichsprachlichen Text strukturell in Satz, Absatz, Seite, Kapitel, etc. untergliedern lässt, stellt die Festlegung eines entsprechenden strukturellen Kontextes in Videodaten ein komplexeres Problem dar. Die strukturelle Untergliederung eines Videos kann in Szenen, Shots, Sub-Shots, etc. erfolgen (vgl. Abschn. 13.3.1). Daher ist es naheliegend, diese Untergliederung auch für die Kontextdefinition zu verwenden. Zeitbezogene Analysedaten, die an ein entsprechendes Segment im Video geknüpft werden, bilden also einen Teil des Kontextes. Daneben existieren oft auch Metadaten, die sich auf das Video als Ganzes be-

[9] Siehe http://nlp.stanford.edu/software/CRF-NER.shtml, aufgerufen am 10.03.2014.

ziehen, wie z. B. der Titel, die Schlüsselwörter oder eine Inhaltsangabe. Diese können ebenfalls als Kontext für jedes Segment herangezogen werden. Daneben kann noch zwischen autoritativen Metadaten, d. h. Metadaten aus einer zuverlässigen Quelle wie z. B. der Autorin oder der Archivarin, und nicht-autoritativen Metadaten, wie z. B. benutzergenerierten Schlagwörtern (Tags) oder Kommentaren unterschieden werden. Letztere sind ebenfalls wie die zeitbezogenen Analysemetadaten stets mit einer gewissen Unsicherheit behaftet und entsprechend weniger zuverlässig. Aus den Metadaten bezogen auf die strukturelle Untergliederung des Videos in Verbindung mit deren ermittelter Zuverlässigkeit wird nun ein statistisches Kontextmodell berechnet, das die wahrscheinlichste Zuordnung einer semantischen Entität aus einer Wissensbasis ermöglicht. Als Wissensbasis dienen hier oft die DBpedia oder auch nationale Normvokabulare, wie z. B. die Gemeinsamen Normdaten der Deutschen Nationalbibliothek[10].

Technische Umsetzung im Suchindex

Jeder Suchprozess startet mit der Formulierung einer Suchanfrage, über die ein Benutzer sein persönliches Informationsbedürfnis möglichst genau auszudrücken versucht. Bei traditionellen schlüsselwortbasierten Suchmaschinen wird die Anfrage durch ein oder mehrere Schlüsselwörter formuliert, wobei sich durch die Verwendung logischer Operatoren (z. B. AND, OR, NOT) eine Exklusion oder Schnittmengenbildung der Suchergebnisse erzielen lässt. Suchmaschinen wie z. B. Google ermitteln die Suchergebnisse meist über eine generelle UND/ODER-Verknüpfung, d. h. mehrere Schlüsselwörter werden zuerst via UND und anschließend via ODER miteinander verknüpft. Bei einer Suche nach „A B C“ werden also alle Dokumente gefunden die „A“, „B“ oder „C“ enthalten, wobei die Dokumente, die alle drei Suchbegriffe enthalten, gegenüber den Ergebnissen, die nur zwei oder eines der gesuchten Schlüsselwörter enthalten, bevorzugt werden.

Komplexere Systeme sind auch in der Lage natürlichsprachliche oder formal-strukturierte Suchanfragen zu verarbeiten. Natürlichsprachliche Anfragen besitzen zwar die größtmögliche Ausdrucksmächtigkeit, jedoch sind sie auf Grund der hohen Komplexität der Interpretation der Anfrage durch eine Maschine und den damit entstehenden Ungenauigkeiten mit heutigen Systemen nur schwer, bzw. nur auf einen sehr eingeschränkten Anwendungsbereich zu realisieren. Formal-strukturierte Anfragen sind hingegen durch ihre klare Definition der Syntax von Maschinen leicht zu interpretieren, jedoch sind nur Experten in der Lage, derartige Anfragen zu formulieren.

Eine besondere Form der Suche ist das sogenannte entitätenzentrierte Retrieval. Hierbei wird nicht nach Schlüsselwörtern oder natürlichsprachigen Formulierungen, sondern nach konkreten semantischen Entitäten gesucht. Die durch die natürliche Sprache bedingten inhärenten Probleme durch Polysemie und Synonymie fallen für das entitätenzentrierte Retrieval nicht mehr ins Gewicht, mit der Folge höherer Genauigkeit und Vollständigkeit der erzielten Suchergebnisse. Zieht man Multilingualität in der Suche in Betracht, kann über eine entitätenzentrierte Suche sogar eine Sprachunabhängigkeit erreicht wer-

[10] Siehe http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html, aufgerufen am 01.03.2014.

den. Allerdings unterscheidet sich die entitätenzentrierte Suche bereits bei der Formulierung der Suchabfrage von einer traditionellen Suchmaschine, da der Benutzer selbst eine Entität nicht manuell über den sie identifizierenden Uniform Resource Identifier (URI) als Adressangabe eingeben kann. Zu diesem Zweck wurden Mechanismen entwickelt, die entweder eine natürlichsprachliche Suchphrase mit Hilfe der Named Entity Disambiguation semantisch analysieren (vgl. Abschn. 13.3.2) oder diese mit der Hilfe von automatisierten Vorschlagsmechanismen in Interaktion mit dem Benutzer auflösen (vgl. Abschn. 13.5.1). Die derart vorprozessierte Suchanfrage besteht dann nicht mehr aus Schlüsselwörtern, sondern aus Entitäten, die jeweils durch ihren eindeutigen URI identifiziert werden.

Bei der Implementierung einer entitätenbasierten Suche müssen die zu indexierenden Dokumente nicht mehr zuerst die komplexe linguistische Analyse (Tokenization, Stemming, etc.) durchlaufen, da diese bereits vorab im Zuge der Named Entity Disambiguation durchgeführt wurde. Im Suchindex wird lediglich gespeichert, an welcher Stelle im Dokument eine bestimmte Entität gefunden wurde. Dies erfolgt einfach durch das Abspeichern des URIs der Entität, der ID des Dokuments sowie der zugehörigen Position innerhalb des Dokuments, bzw. Videos, die dieser Entität zugeordnet werden konnten.

Das entitätenzentrierte Retrieval ist als Vorstufe bzw. als Voraussetzung zur semantischen Suche zu sehen. Semantische Suche in diesem Sinne bedeutet, dass während des Suchprozesses nicht nur die in der Suchanfrage angegebenen Entitäten berücksichtigt werden, sondern auch die Beziehungen, wie diese mit anderen Entitäten in Zusammenhang stehen. In Abhängigkeit von der verwendeten Wissensbasis, aus der die Entitäten stammen, können direkte und indirekte Relationen zwischen den Entitäten im Suchindex identifiziert werden. Wird z. B. nach der DBpedia-Entität „Albert Einstein“[11] gesucht, so existiert eine direkte Beziehung zwischen der DBpedia-Entität Albert Einsteins und der DBpedia-Entität der Stadt „Ulm“, da Albert Einstein dort geboren wurde. Eine indirekte Beziehung besteht z. B. zur DBpedia-Entität „Max Planck“, da beide Entitäten Albert Einstein und Max Planck eine (direkte) Verbindung zur DBpedia-Entität „Physiknobelpreisträger“ besitzen. Derartige Relationen werden bei der Suche genutzt, um einerseits mehr relevante Dokumente zu bestimmen und andererseits das Ranking (z. B. die Suchergebnisreihenfolge) zu optimieren. Eine Suche nach der Entität „Physiknobelpreisträger“ liefert auf diese Weise auch Dokumente über „Albert Einstein“ und „Max Planck“ zurück, während diese Dokumente bei einer traditionellen schlüsselwortbasierten Suche notwendigerweise immer auch den Begriff „Physiknobelpreisträger“ beinhalten müssen.

Eine besondere Stellung nimmt das entitätenzentrierte Retrieval in Kombination mit der traditionellen schlüsselwortbasierten Suche ein (Hybrides Retrieval). Hierbei werden die URIs der Entitäten und deren zugehörige Kontextinformationen gemeinsam mit den Indextermen der klassischen linguistischen Analyse im Suchindex verwaltet. Das zur Named Entity Disambiguation eingesetzte Verfahren wird in diesem Fall zur Fehlervermei-

[11] Die Entität „Albert Einstein“ hat in der DBpedia folgende URI: http://dbpedia.org/resource/Albert_Einstein.

dung zugunsten der Genauigkeit (Precision) hin optimiert, sodass die Fehlerrate auf ein Minimum reduziert werden kann. Dies geschieht allerdings auf Kosten der Vollständigkeit (Recall), also der Anzahl der ermittelten Entitäten im zugrundeliegenden Dokument. D.h. wenn das System nicht sicher ist, welche Entität einem Term aus dem Dokument zugeordnet werden soll, wird sie eher verworfen, als eine potenziell falsche Zuordnung zuzulassen. Dieser durch die Named Entity Disambiguation nicht erkannte Term kann dann zwar nicht über die entitätenzentrierte Suche gefunden werden, jedoch wird er durch die in Kombination verwendete schlüsselwortbasierte Suche gefunden. Allerdings treffen bei diesem kombinierten Verfahren zwei völlig unterschiedliche Paradigmen aufeinander, deren effiziente Verknüpfung aktuell noch Gegenstand der Forschung ist. Nicht nur aus diesem Grund bleibt die schlüsselwortbasierte Suche auch heute noch relevant. Die Mehrheit der Suchmaschinenbenutzer ist mit der Arbeitsweise traditioneller Suchmaschinen vertraut. Daraus ergibt sich eine Erwartungshaltung des Benutzers an die Funktionsweise einer Suchmaschine, die vor allem im hier vorgestellten kombinierten Ansatz nur schwer zu erfüllen ist. Dem Benutzer fällt es schwer zwischen der Suche nach Entitäten und der Suche nach Schlüsselwörtern zu unterscheiden, erst recht, wenn beides in Kombination auftritt. Wird z. B. im Text „Er sitzt auf der Bank vor der Sparkasse.“ nach der Entität „Geldinstitut“ gesucht, erwartet der Benutzer in vielen Fällen auch die (fehlerhafte) Hervorhebung des Terms „Bank“ (als synonyme Bezeichnung für Geldinstitut). Dennoch ermöglichen es hybride Ansätze, den Benutzer schrittweise an die neuen semantischen Suchparadigmen heranzuführen.

13.3.3 Explorative Suche

Zur explorativen Suche werden explizite und implizite inhaltliche Zusammenhänge einzelner Entitäten genutzt, d. h. zu den jeweils vorhandenen semantischen Metadaten einer Informationsressource werden zusätzliche Metadaten bestimmt, die mit diesen inhaltlich zusammenhängen. Ist also z. B. ein Videosegment mit dem Schlüsselwort „Stephen King“ annotiert, wird über eine Verknüpfung mit den DBpedia-Daten die DBpedia-Entität des US-amerikanischen Autors „Stephen King“ bestimmt und mit dem Schlüsselwort verknüpft. Über die enzyklopädischen Daten der DBpedia werden zusätzliche Informationen, wie z. B. das bevorzugte literarische Genre des Autors („Fantasy“, „Science Fiction“), sein Geburtsort („United States“) oder auch andere, inhaltlich verwandte Autoren („Edgar Allan Poe“) sowie weitere assoziativ verbundene Entitäten bestimmt (z. B. „Maine“, „Desperation“, „Author“, „Pseudonym“, etc.). Zusätzlich erfolgt ein automatischer Abgleich, ob zu diesen verknüpften Begriffen überhaupt Videosegmente in der zugrundeliegenden Datenbank vorhanden sind und wieviele Suchtreffer diesbezüglich erzielt werden. Assoziierte Begriffe, zu denen in der aktuellen Datenbasis keine Videosegmente gefunden werden können, werden sofort ausgefiltert.

Eine einfache explorative Navigationshilfe wird in Abb. 13.5 dargestellt. Links neben der eigentlichen Trefferliste für den Suchbegriff „Stephen King“ werden weiterführende

Abb. 13.5 Explorative Videosuche im semantischen yovisto-Prototyp mit dem Suchbegriff „Stephen King“ und einer Detailvergrößerung des explorativen Navigationselements (links). Auf der rechten Seite des Suchergebnisses werden Suchfacetten (Facets) angezeigt, über die die erzielten Suchergebnisse gefiltert werden können

Suchbegriffe und die dazu vorhandene Anzahl an Informationsressourcen angezeigt, für die ihrerseits durch Anklicken erneut eine Suche ausgelöst werden kann. Dabei werden qualifizierte Assoziationen, bei denen die Beziehung zwischen Suchbegriff und assoziiertem Begriff benannt werden kann, von unqualifizierten Assoziationen ohne Nennung der verknüpfenden Beziehung unterschieden. So werden die mit dem ursprünglichen Suchbegriff in Bezug stehenden Begriffe als Navigationselement verwendet, mit dem eine explorative Suche im vorhandenen Gesamtdatenbestand realisiert wird [4].

13.4 yovisto und Linked Open Data

Linked Open Data bildet im Zusammenhang mit der Videosuchmaschine yovisto.com eine fundamentale Grundlage für die semantische Suche. Als zentrale Wissensbasis wird die DBpedia herangezogen, mit der die in den Videometadaten erkannten Entitäten verknüpft

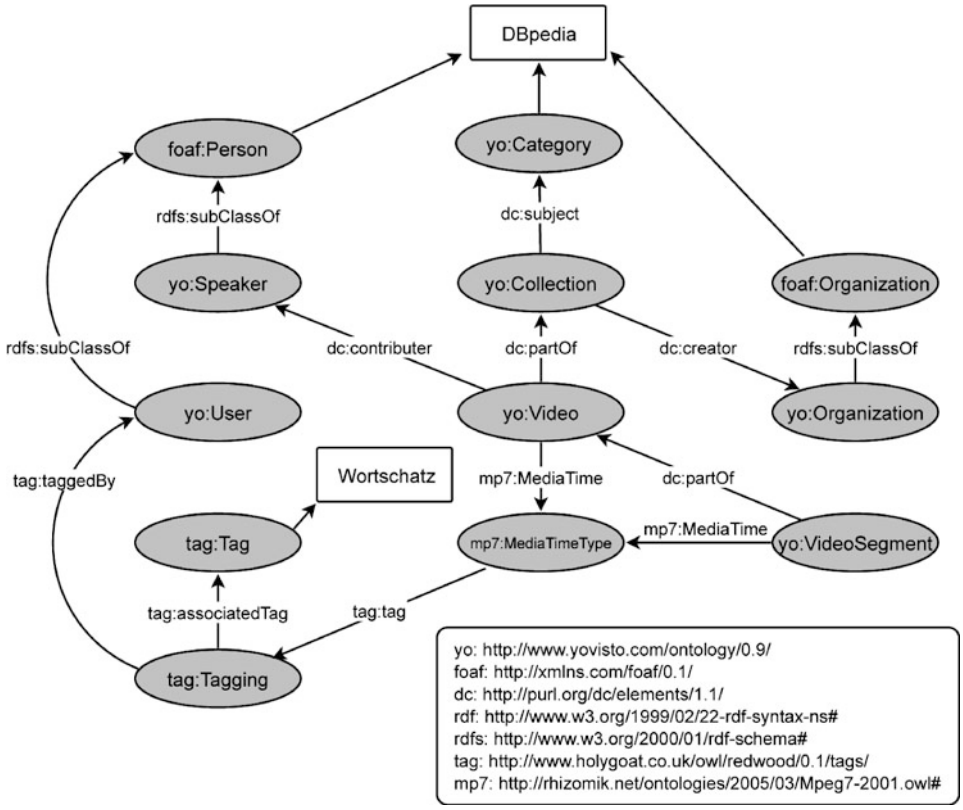


Abb. 13.6 Yovisto Ontologie (vgl. http://yovisto.com/ontology/)

werden. Im Gegenzug stellt yovisto.com eigene Metadaten und Analysedaten selbst auch als Linked Open Data zur Verfügung. Über einen öffentlich zugänglichen SPARQL Endpunkt kann die yovisto Wissensbasis abgefragt werden (vgl. Abschn. 13.4.1). Zusätzlich enthalten die via yovisto.com produzierten HTML-Webseiten zur Darstellung von Suchergebnissen semantisch annotierte Daten eingebettet via RDFa[12] (vgl. Abschn. 13.4.2).

13.4.1 RDF und SPARQL Endpunkt

In Abb. 13.6 wird das der Suchmaschine yovisto.com zu Grunde liegende Datenmodell als OWL-Ontologie[13] schematisch dargestellt.

Beim Entwurf der in yovisto.com verwendeten Ontologie lag der Fokus auf Einfachheit bei gleichzeitiger Wiederverwendung bereits existierender Vokabulare. Sämtliche Rela-

[12] Siehe http://www.w3.org/TR/xhtml-rdfa-primer/, aufgerufen am 01.03.2014.
[13] Siehe http://www.w3.org/TR/owl-ref/, aufgerufen am 01.03.2014.

tionen zwischen den definierten Klassen werden durch die bekannten Vokabulare DublinCore[14], Mpeg7-Ontology[15], Tags-Ontology[16] sowie das RDF-Schema realisiert. Die Definition von neuen Klassen erfolgte bei yo:Speaker und yo:Organization als Spezialisierung der bereits existierenden Klassen aus dem FOAF-Vokabular[17]. Eine Verlinkung in die Wissensbasis DBpedia.org erfolgt in der aktuell verwendeten Version über Instanzen von Personen, Kategorien und Organisationen. Von den Nutzern vergebene Tags werden mit dem Wortschatz Leipzig[18] verknüpft. Sämtliche Ressourcen sind über HTTP Content-Negotiation (Hypertext Transfer Protocol) de-referenzierbar sowohl in ihrer RDF Repräsentation als auch in einer zugehörigen HTML (Hypertext Markup Language) Repräsentation.

Yovisto stellt für den direkten Zugriff auf die semantischen Metadaten auch einen SPARQL Endpunkt bereit[19], über den automatisiert bzw. auch interaktive Suchanfragen über die RDF(S)-Abfragesprache SPARQL gestellt werden können [11]. Damit bietet yovisto einen unkomplizierten und frei zugänglichen Anknüpfungspunkt zur Nutzung der Ergebnisse der automatisierten Analyse bzw. der benutzergenerierten Tags, die sich auf diese Weise einfach in Form von Mashups in anderen Webanwendungen weiterverwenden lassen.

13.4.2 Eingebettete semantische Information mit RDFa

Mit Hilfe von RDFa[20] lassen sich RDF-Daten direkt in HTML Webseiten einbetten. Zwar werden diese Annotationen durch den Browser nicht explizit dargestellt, allerdings können sie auf einfache Weise automatisiert aus der Webseite ausgelesen und weiterverarbeitet werden. Dies ist besonders interessant für (semantische) Suchmaschinen im WWW, die auf diese Weise Webseiteninhalte besser „verstehen“ können. Bei yovisto.com werden die dargestellten Webseiten zu diesem Zweck, wie in Abb. 13.7 dargestellt, um folgende semantischen Metadaten ergänzt:

- Der Dokument-Typ (DOCTYPE) wird festgelegt auf „XHTML+RDFa“ mit Angabe der entsprechenden Document Type Definition (DTD).
- Im HTML-Tag werden alle verwendeten Namensräume (Namespaces) aufgeführt.
- Im Head-Tag der HTML-Webseite wird das externe Metadaten-Profil für RDFa angegeben.

[14] Siehe http://dublincore.org/, aufgerufen am 01.03.2014.
[15] Siehe http://www.w3.org/2005/Incubator/mmsem/XGR-mpeg7/, aufgerufen am 01.03.2014.
[16] Siehe http://www.holygoat.co.uk/projects/tags/, aufgerufen am 01.03.2014.
[17] Siehe http://www.foaf-project.org/, aufgerufen am 01.03.2014.
[18] Siehe http://wortschatz.uni-leipzig.de/, aufgerufen am 01.03.2014.
[19] Siehe http://sparql.yovisto.com/, aufgerufen am 01.03.2014.
[20] Siehe http://www.w3.org/TR/xhtml-rdfa-primer/, aufgerufen am 01.03.2014.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
      xmlns:dcmitype="http://purl.org/dc/dcmitype/"
      xmlns:lom="http://ltsc.ieee.org/rdf/lomv1p0/lom#"
      xmlns:lomvoc="http://ltsc.ieee.org/rdf/lomv1p0/vocabulary#"
      xmlns:vcard="http://www.w3.org/2006/vcard/ns#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:rss="http://purl.org/rss/1.0/"
      xmlns:skos="http://www.w3.org/2008/05/skos#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:yo="http://www.yovisto.com/ontology/0.9/"
      xmlns:y="http://www.yovisto.com/resource/"
      xmlns:addthis="http://www.addthis.com/help/api-spec">
<head  profile="http://ns.inria.fr/grddl/rdfa/">
```

Abb. 13.7 Kopf einer Webseite mit RDFa Einbettung

```
<div about="/resource/organization/939" typeof="yo:Organization">
 ▾ <a href="/resource/organization/939">
    <img alt="Hasso Plattner Institut" src="logos/hpi.png">
    <span property="foaf:name">Hasso Plattner Institut</span>
  </a>
</div>
```

Abb. 13.8 Einfache RDFa-Einbettung in HTML

Die Einbettung der RDF Metadaten in die Webseite erfolgt wie im Beispiel in Abb. 13.8 dargestellt. Im HTML `div` - Element wird über das HTML-Attribut `about` festgelegt, auf welche RDF Ressource sich der über `div` - Element strukturierte Inhalt bezieht.

Aus dem in Abb. 13.8 angegebenen RDFa Beispiel lassen sich die folgenden RDF-Tripel extrahieren (in Präfixdarstellung):

```
y:organization/939 rdf:type yo:Organization .
y:organization/939 foaf:name "Hasso Plattner Institut".
```

RDFa ist ein nützliches Werkzeug, um semantische Metadaten ohne großen Aufwand in bereits existierende Webseiten einzubinden.

13.5 Graphische Benutzeroberflächen

Wie bereits ausgeführt, stellt die semantische Suche ebenfalls neue Anforderungen an die Benutzerschnittstellen für Suchmaschinen. Wurde zuvor eine einfache Autovervollständigung von eingegebenen Suchphrasen unterstützt, muss diese im Zuge der entitätenzentrierten Suche auch die Möglichkeit der Disambiguierung von mehrdeutigen Begriffen bieten (vgl. Abschn. 13.5.1). Andererseits gestatten inhaltsbasierte Suchfacetten eine einfache Strukturierung und Filterung der erzielten Suchergebnisse nach inhaltlichen Kriterien (vgl. Abschn. 13.5.2).

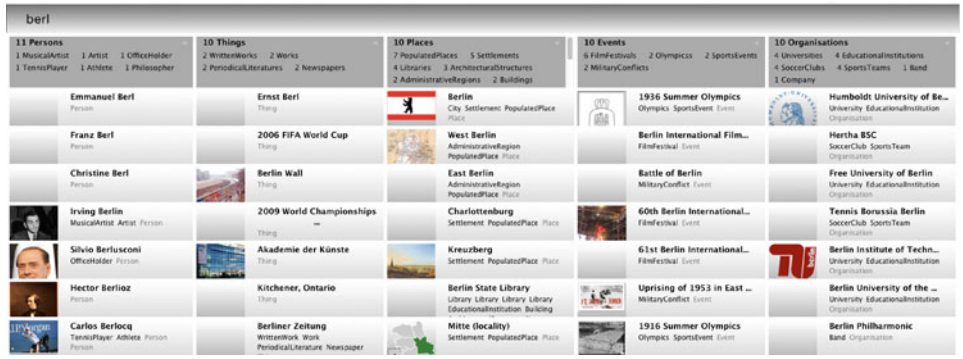


Abb. 13.9 Möglichkeit der Implementierung einer Autovervollständigung für entitätenzentrierte Suche bei der Eingabe des Suchterms „berl“

13.5.1 Autovervollständigung und Suchvorschläge

Wie bereits in Abschn. 13.3.2 diskutiert, kann die Disambiguierung der Terme einer Suchanfrage schon während der Eingabe der Suchphrase mit Hilfe eines Vorschlagsmechanismus erfolgen. Viele Suchmaschinen erleichtern dem Nutzer die Eingabe der Suchanfrage durch eine automatische Vervollständigung der Suchbegriffe. Üblicherweise wird dabei rein statistisch vorgegangen, indem die am häufigsten verwendeten Suchphrasen, deren Präfix mit der aktuellen Eingabe des Benutzers übereinstimmt, zur Vervollständigung vorgeschlagen werden. Bei einer entitätenzentrierten Suche werden im Gegensatz dazu semantische Entitäten vorgeschlagen, deren Bezeichner mit der Eingabe des Benutzers übereinstimmen, wobei Synonyme und Homonyme Berücksichtigung finden.

Abbildung 13.9 zeigt eine prototypische Implementierung der Autovervollständigung für die entitätenzentrierte Suche[21]. Im dargestellten Beispiel wurde vom Benutzer die Suchphrase „berl“ eingegeben. Die Autovervollständigung komplettiert diese Eingabe zu vollständigen Bezeichnern, deren zugehörige Entitäten aus der DBpedia ermittelt und vorgeschlagen werden. Zur besseren Übersicht werden die vorgeschlagenen Entitäten gruppiert nach den Kategorien Personen, Orte, Ereignisse, Organisationen und Sonstiges. Jede Gruppe wird sortiert nach der am besten zur Eingabe passenden Entität, wobei die Popularität der Entität für die Platzierung innerhalb der dargestellten Reihenfolge ausschlaggebend ist, um den Nutzererwartungen besser zu entsprechen. Mit jedem weiteren vom Benutzer eingegebenen Buchstaben des Suchterms werden die Auswahlmöglichkeiten entsprechend verfeinert. Dem Vorschlagssystem liegt ein Index zu Grunde, der alle Entitäten der DBpedia bereithält. Zu jeder Entität muss der Name, alternative Schreibweisen (Synonyme, Übersetzungen und Falsch-Schreibweisen), der zugehörige URI und, falls vorhanden, ein Vorschaubild indexiert werden [12].

[21] Für eine Demo siehe http://www.yovisto.com/labs/autosuggestion, aufgerufen am 01.03.2014.

Die Besonderheit dieses Ansatzes liegt in der Unterstützung des Benutzers bei der Formulierung seiner Anfrage:

- indem alternative Bezeichnungen und Übersetzungen berücksichtigt werden, so werden z. B. die Anfragen nach „Kosmonaut“, „Astronaut“ oder „Taikonaut“ auf dieselbe semantische Entität dbpedia:Raumfahrer abgebildet, und
- indem durch die vom Benutzer vorgenommene manuelle Auswahl einer vorgeschlagenen Entität automatisch andere Entitäten mit gleichlautender Bezeichnung (Homonyme) von der Suche ausgeschlossen werden.

13.5.2 Semantische Suchfacetten

Eine weitere Möglichkeit zur Nutzung semantischer Metadaten in der Suche besteht in der Verwendung von inhaltsbasierten Suchfacetten. Suchfacetten dienen zur Filterung der Suchergebnisse und beziehen sich oft auf technische Metadaten, wie z. B. Dokumentendatierung, Dokumentenkodierung oder Dokumentengröße. Traditionell können auf diese Weise z. B. Dokumente eines bestimmten Alters oder einer bestimmten Größe aus dem Suchergebnis herausgefiltert werden. Ein bekanntes Beispiel zur facettierten Suche ist die Google Bildsuche[22], die die Möglichkeit bietet, Bilder bestimmter Größe, Farbe oder Dateityps zu filtern. Semantische Suchfacetten beziehen sich auf inhaltliche Filtermöglichkeiten der Dokumente. Werden in den zugrundeliegenden Dokumenten einer Suchmaschine Personen, Ortsangaben oder auch Ereignisse identifiziert, können die darin erkannten semantischen Entitäten gemäß ihrer Oberbegriffe aggregiert als Filterfacette zum Suchergebnis passend dargestellt werden. Suchfacetten bieten auf diese Weise einen besseren Einblick in die Inhalte der erzielten Suchergebnisse und erlauben so eine Art inhaltsbasierter Navigation durch den Suchergebnisraum. Abbildung 13.4 zeigt ein Suchergebnis des yovisto Prototyps zur semantischen Suche, in der auf der rechten Seite des Bildschirms Suchfacetten (Facets) angezeigt werden, nach denen die erzielten Suchergebnisse inhaltlich bzgl. Sprache, Kategorien, publizierender Organisation, usw. gefiltert werden können.

13.6 Evaluation semantischer und explorativer Videosuche

Ausgehend von der Definition der semantischen Suche können potenziell qualitativ hochwertigere und vollständigere Suchergebnisse erzielt werden. Wird die vom Benutzer eingegebene Suchphrase mit Hilfe der Autosuggestion (vgl. Abschn. 13.5.1) zu einer semantischen Entität ergänzt, werden bei der semantischen Suche nach dieser eindeutig zugeordneten Entität vormals nicht zutreffende Ergebnisse vermieden, bedingt durch die

[22] Siehe http://www.google.com/imghp, aufgerufen am 01.03.2014.

Auflösung von Mehrdeutigkeiten – sowohl in der Suchphrase als auch in den zugrundeliegenden Dokumenten in der Suchmaschine. In der gleichen Weise können Synonyme eindeutigen semantischen Entitäten zugeordnet werden und so vormals unvollständige Ergebnisse komplettiert werden. Dies lässt sich quantitativ evaluieren (vgl. Abschn. 13.6.1). Andererseits lässt sich der Nutzen, den eine veränderte semantische Suche dem Benutzer tatsächlich bringt, nur qualitativ ermitteln (vgl. Abschn. 13.6.2)

13.6.1 Quantitative Evaluation

Quantitative Evaluation wird im Information Retrieval mit Hilfe der informationstechnischen Kenngrößen Recall (Vollständigkeit) und Precision (Genauigkeit) durchgeführt. Recall bezeichnet dabei das Verhältnis zwischen den korrekt gefundenen Ergebnissen zu den tatsächlich korrekten Ergebnissen. Precision dagegen berechnet sich aus dem Verhältnis der korrekt gefundenen Ergebnisse zu den insgesamt gefundenen Ergebnissen. Beide Maßzahlen setzen voraus, dass zuvor manuell eine korrekte Ergebnismenge bzgl. einer durchzuführenden Suche ermittelt wurde. Während dies bei einer beschränkten Informationsmenge, wie z. B. einem kleineren Videoarchiv problemlos möglich ist, können diese Maßzahlen bei einer webbasierten Suche aufgrund der riesigen Informationsmenge im Web stets nur abgeschätzt werden.

Für eine explorative Suche kann eine solche quantitative Abschätzung aber nur schwer erfolgen, da es schwierig ist, diesbezüglich eine „korrekte“ Ergebnismenge zu ermitteln. Die explorative Suche schlägt gleich einem Empfehlungssystem „passende“ bzw. naheliegende weiterführende Ergebnisse vor, die nicht zum direkt erzielten Suchergebnis zählen. Der ausschlaggebende Zusammenhang aufgrund dessen der Ergebnisvorschlag der explorativen Suche ermittelt wurde, kann nur schwer objektiv eingeschätzt werden, da dessen subjektiver Nutzen stets vom Betrachter abhängt. Bezugnehmend auf das in Abschn. 13.3.3 angegebene Beispiel zur Suche nach „Stephen King“ könnten für den einen Betrachter weitere Suchergebnisse zu anderen, ähnlichen Autoren relevant sein, während ein anderer Betrachter den lokalen Bezug in den Vordergrund stellt und Suchergebnisse zum U.S. Bundesstaat Maine bevorzugt, in dem Stephen King lebt. Um die Qualität der explorativen Suche beurteilen zu können, muss daher eine qualitative Evaluation durchgeführt werden.

13.6.2 Qualitative Evaluation

Die qualitative Evaluation einer Suche setzt die Zufriedenheit des Benutzers in den Mittelpunkt der Beurteilung. Diese ist davon abhängig, wie gut, wie vollständig und wie komfortabel die Informationsbedürfnisse des Benutzers befriedigt werden können. Für yovisto wurde die explorative Suche qualitativ evaluiert, indem einer Gruppe von Benutzern eine Reihe von komplexen Suchaufgaben gestellt wurde. Dazu wurde bestimmt, in

welcher Zeit es dem Benutzer gelang, die gestellte Aufgabe zu lösen und wie zufrieden der Benutzer mit der Lösung der Aufgabe war im Vergleich zur Lösung derselben Aufgabe mit einer traditionellen Suche [13]. Beispiele für diese komplexen Suchaufgaben in der Videosammlung der yovisto Suchmaschine waren z. B. „Mit welchen Physikern war Albert Einstein in den 1920er Jahren bekannt und anlässlich welcher Ereignisse könnte er sie kennengelernt haben?“ oder „Welche Philosophen bauen auf den Theorien des griechischen Philosophen Platon auf?“. Diese Aufgaben sind nicht mit den konventionellen Suchabfragen vergleichbar, in denen stets nach einem oder mehreren Schlüsselbegriffen gesucht wird, sondern es ist eine Transferleistung des Benutzers erforderlich, zunächst die passenden Schlüsselwörter für eine Ausgangsbasis zu finden (z. B. „Albert Einstein“) und dann über die Möglichkeiten der explorativen Suche den Bestand des Videoarchivs gezielt zu durchstöbern. In [12] konnte gezeigt werden, dass mit Hilfe der explorativen Suche in kürzerer Zeit zugleich eine größere Zahl für die Suchabfrage tatsächlich relevanter Videos ermittelt werden konnte bei gleichzeitiger deutlicher Verbesserung der Nutzerzufriedenheit.

Literatur

1. Sack, H. Feb. 2010. Semantische Suche – Theorie und Praxis am Beispiel der Videosuchmaschine yovisto.com. *Web 3.0 & Semantic Web, HMD – Praxis der Wirtschaftsinformatik, Nr. 271* Hrsg. U. Hentgartner und A. Meier. Heidelberg: dpunkt Verlag. ISSN 1436-3011
2. Sack, H. 2005. NPBibSearch: An Ontology Augmented Bibliographic Search, *Proc. of SWAP 2005, the 2nd Italian Semantic Web Workshop*. Trento, Italy, December 14–16, 2005, CEUR Workshop Proceedings, ISSN 1613-0073
3. Yang, H., B. Quehl, und H. Sack. 2012. A framework for improved video text detection and recognition, Multimedia Tools and Applications, Springer, US, pp. 1–29 (2012). ISSN: 1380-7501, DOI: 10.1007/s11042-011-0971-2
4. Waitelonis, J., und H. Sack. 2009. *Towards Exploratory Video Search by Using Linked Data* Proc. of 2nd IEEE International Workshop on Data Semantics for Multimedia Systems and Applications (DSMSA2009), San Diego, December 14–16. California: Multimedia Tools and Applications, 59 (2): 645–672.
5. Lowe, D.G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Jrnl. Of Computer Vision* 60(2): 91–110
6. Hentschel, C., S. Gerke, und E. Mbanya. 2013. *Classifying images at scene level: comparing global and local descriptors* AMR 2011 – 9th International Workshop on Adaptive Multimedia Retrieval., 72–82
7. Csurka, G., C.R. Dance, L. Fan, J. Willamowski, C. Bray, und D. Maupertuis. 2004. *Visual Categorization with Bags of Keypoints* WS. on Statistical Learning in Computer Vision, ECCV., 1–22
8. Snoek, C.G.M., und M. Worring. 2009. Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval* 2(4): 215–322
9. Zhang, J., M. Marszalek, S. Lazebnik, und C. Schmid. 2006. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *Int. Jrnl. Of Computer Vision* 73(2): 213–238

10. Nadeau, D., und S. Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30(1): 3–26. doi:10.1075/li.30.1.03nad

11. Prud'hommeaux, E., S. Harris, und A. Seaborne (Hrsg.). 2013. SPARQL 1.1 Query Language. W3C, URL: http://www.w3.org/TR/sparql11-query

12. Osterhoff, J., J. Waitelonis, und H. Sack. 2012. *Widen the Peepholes! Entity-Based Auto-Suggestion as a rich and yet immediate Starting Point for Exploratory Search* 2. Workshop Interaktion und Visualisierung im Daten-Web (IVDW 2012), im Rahmen der INFORMATIK 2012, Braunschweig, Sep.16–21

13. Waitelonis, J., H. Sack, Z. Kramer, und J. Hercher. 2010. *Semantically Enabled Exploratory Video Search* Proc. of Semantic Search Workshop (SemSearch10) at the 19th Int. World Wide Web Conference (WWW2010), Raleigh, NC, USA, 26–30 April

Linked Data als integraler Bestandteil der Kernprozesse bei Wolters Kluwer Deutschland GmbH

14

Christian Dirschl und Katja Eck

Zusammenfassung

Das Linked Data Paradigma zielt darauf ab, Informationen im Web einfacher und nachhaltiger auffindbar zu machen. Das Strukturieren und das zielgruppenspezifische Aufbereiten von Informationen ist gleichzeitig Kernaufgabe eines Medienunternehmens wie Wolters Kluwer Deutschland. Somit ergibt sich grundsätzlich eine natürliche Übereinstimmung zwischen den Anforderungen der Medienbranche und den Semantic Web Technologien, die das Fundament von Linked Data bilden. Dieser Beitrag zeigt anhand von Businessanforderungen, wie sich die Wertschöpfungskette innerhalb eines Medienhauses unter Einbeziehung von Linked Data weiterentwickeln kann. Insbesondere die systematische Trennung von textlichem Content und Metadaten eröffnet völlig neue Möglichkeiten im Gesamtprozess. Die strukturierte Einbindung externer Wissensquellen stellt dabei ein nicht zu vernachlässigendes Potential dar. Die dynamische Entwicklung in diesem Bereich erfordert die Analyse und Abschätzung der technischen Konzepte und Werkzeuge um mittel- bis langfristig neue wertschöpfende Geschäftsmodelle zu etablieren.

14.1 Die wachsende Bedeutung von Metadaten in Medienunternehmen

Die fortschreitende Digitalisierung der Gesellschaft und der technologische Fortschritt, insbesondere in den letzten beiden Jahrzehnten, haben eine fast schon groteske Situation geschaffen: wertvolle Informationen sind ubiquitär verfügbar, können jedoch oft nur durch

C. Dirschl ✉ · K. Eck
Content Strategy and Architecture, Wolters Kluwer Deutschland GmbH, Freisinger Strasse 3, 85716 Unterschleißheim, Deutschland
e-mail: cdirschl@wolterskluwer.de

T. Pellegrini, H. Sack, S. Auer (Hrsg.), *Linked Enterprise Data*, X.media.press,
DOI 10.1007/978-3-642-30274-9_14

hohen Ressourceneinsatz gefunden werden. Dies stellt ein Problem für alle Industrien dar, aber gerade das Verlags- und Mediensegment verlangt nach skalierbaren Lösungen, um die geleistete Service-Qualität des Kerngeschäfts – die Erstellung, Aufbereitung und Weitergabe von Informationen – erhalten und weiterhin gewährleisten zu können.

Neben den erforderlichen Datenressourcen ergeben sich auch Herausforderungen auf technologischer Ebene. Die Nutzung mobiler Medien und damit die Notwendigkeit auf verschiedenste Endgeräte abgestimmte und optimierte Informationsdienste anbieten zu müssen, haben großen Einfluss auf die Datenaufbereitung, insbesondere im Hinblick auf die Granularität von Informationen.

Einen weiteren Aspekt stellt die zunehmende Globalisierung dar, der sich ein weltweit agierender Konzern wie Wolters Kluwer stellen muss. Als Beispiele seien hier Themen wie Mehrsprachigkeit und Lokalisierung erwähnt (z. B. müssen generisch entwickelte Entscheidungsunterstützungstools an die jeweiligen Gegebenheiten einzelner Länder und deren Rechtshistorien angepasst werden). Daraus ergeben sich eine hohe Erschließungstiefe und flexible Optionen zur Aufbereitung von Informationen für unterschiedliche Anwendungen und Produktionskanäle.

Software-Applikationen und Konzepte rund um das Linked Data Paradigma stellen hier den benötigten technologischen Rahmen für die nachhaltige Umsetzung dar. Metadaten – basierend auf kontrollierten Vokabularen wie Schlagwortlisten, Taxonomien oder Thesauri – sind eine wesentliche Komponente und bilden die Voraussetzung für den Einsatz von Semantic Web Technologien.

14.2 Die Nutzung von Semantic Web Technologien als Lösungsansatz

Die Idee des Semantic Web entstand aus einer technologischen Vision: Man wollte eine Infrastruktur bereitstellen, die das Web mit so viel Intelligenz und Hintergrundwissen ausstattet, damit es selbst für komplexe Fragestellungen und die zukünftig erwarteten Datenmengen eine unverzichtbare Kernkomponente bleiben konnte.

Diese Grundidee adressiert die oben genannten Anforderungen, allerdings ist das Semantic Web in großen Teilen noch Gegenstand der Forschung. Derzeit gibt es wenige kommerzielle Produkte, die operativ im industriellen Umfeld genutzt werden können.

Um die Forschung verstärkt für die Bedürfnisse eines Informationsdienstleisters zu sensibilisieren, hat sich Wolters Kluwer Deutschland 2010 entschlossen, als Use Case Partner im von der EU geförderten Leuchtturmprojekt LOD2[1] zu agieren. Das Projekt bietet durch einen Mix aus akademischen und kommerziellen Partnern bis Sommer 2014 den idealen Rahmen für technologische Entwicklungen, um Semantic Web Technologien umfassend einsetzen zu können [1]. Erste Prototypen sowie auch die Nutzung einiger Komponenten in der operativen Contentverarbeitung legen den Schluss nahe, eine erfolgversprechende strategische Ausrichtung gewählt zu haben.

[1] Siehe http://lod2.eu, aufgerufen am 20. Februar 2014.

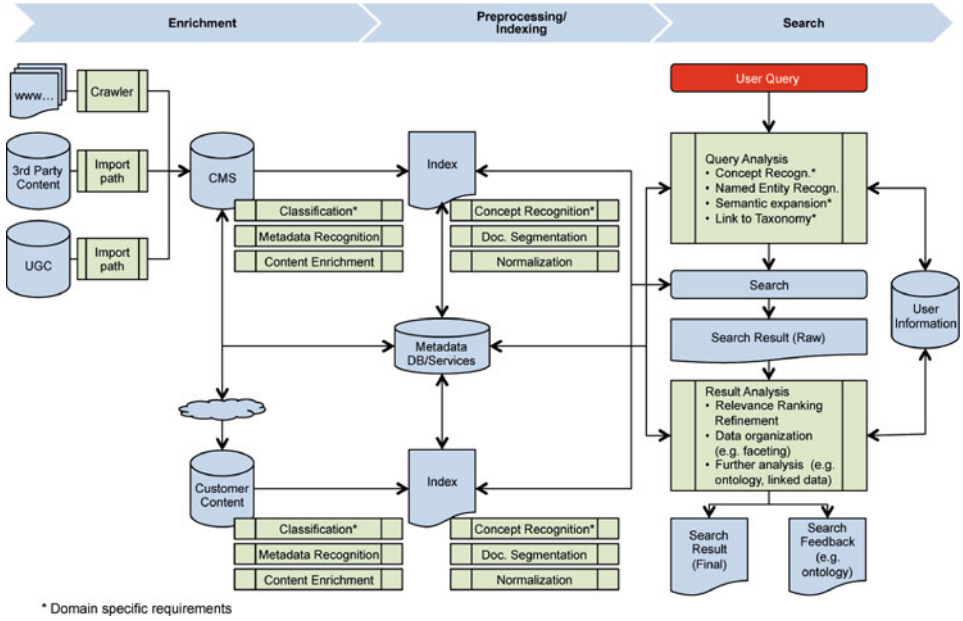


Abb. 14.1 Zukünftiger Workflow der Contentverarbeitung bei Wolters Kluwer Deutschland

Folgende Herausforderungen sollen durch die Technologie adressiert werden:

1. **Flexible Granularitätsstufen:** Die textliche Information soll in unterschiedlicher semantischer Granularität für die verschiedenen Medien genutzt werden können. Eine Online-Fachdatenbank möchte z. B. eine Information im Kontext eines ganzen Buchkapitels anzeigen, eine App dagegen nur auf Absatzebene.
2. **Einsatz unterschiedlicher Wissensdomänen:** Die textliche Information soll mit Informationen aus unterschiedlichen Wissensdomänen sowie weiteren Metadaten angereichert werden können.
3. **Anreicherung von Textressourcen:** Es sollen nicht nur eigene Texte angereichert werden, sondern auch Texte aus dem Internet, Social-Media-Inhalte oder Informationen aus LOD-Quellen.
4. **Gewährleistung von Interoperabilität:** Die Metadaten sollen in einem offenen standardisierten Format vorgehalten werden, um sie interoperabel zu halten und eine Abhängigkeit von proprietären Standards oder Werkzeugen zu vermeiden.
5. **Wiederverwendbarkeit von Ressourcen:** Externe Wissensbasen sollen über ein standardisiertes Verfahren mit den internen Metadaten verknüpft werden, um existierende Ressourcen wiederverwenden zu können.
6. **Gewährleistung der Datenkonsistenz:** Externe Wissensbasen sollen auch direkt mit den textlichen Informationen verknüpft werden können, um zeitlich unbegrenzte Datenkonsistenz zu gewährleisten.

Daraus ergeben sich folgende systemarchitektonische Grundentscheidungen (siehe Abb. 14.1):

1. Trennung von textlichem Content und Metadaten
2. Speicherung der Metadaten in einem offenen Standardformat
3. Aufbau einer Domänenwissensbasis, die für alle Prozesse innerhalb der Contentverarbeitung und -nutzung verwendet werden kann

14.3 Das Umsetzungsprojekt

Diese Anforderungen führen zu einer Grundarchitektur, die sich vor allem dadurch auszeichnet, dass das ehemals geschlossene System CMS (jeder Text und dessen Metadaten vereint in einer XML-Datei) aufgebrochen und drei Teilsysteme etabliert wurden:

1. Kern-CMS mit den Textinformationen als XML
2. Einsatz kontrollierter Vokabulare im SKOS[2] Format
3. Weitere Metadaten sowie Dokumentstrukturinformationen in RDF[3]

Diese Aufteilung gibt die Flexibilität und auch den strukturellen Rahmen, sukzessive die genannten Anforderungen umsetzen zu können.

Zu Projektbeginn wurde eingehend diskutiert, welche Mächtigkeit das zu Grunde liegende Wissensmodell haben sollte. Letztendlich fiel die Entscheidung für SKOS und gegen OWL[4], obwohl man damit bestimmte Einschränkungen in Kauf nimmt. Die Hauptgründe hierfür waren:

1. SKOS basiert auf dem Wissensmodell Thesaurus. Thesauri sind weit verbreitet und intuitiv verständlich, ohne dass umfangreiches, spezifisches Vorwissen im Bereich der Wissensmodellierung vorausgesetzt werden muss.
2. SKOS-Thesauri lassen sich relativ einfach erstellen und warten.
3. Werkzeuge zur Modellierung von OWL wie Protégé[5] sind in der Nutzung sehr komplex und aufwändig.
4. Die eigentliche Stärke von OWL – die Möglichkeit Schlüsse (Inferenzen) aus Wissensbasen zu ziehen – wird in diesem Kontext in absehbarer Zeit nicht genutzt werden.

[2] Siehe http://www.w3.org/2004/02/skos/, aufgerufen am 20. Februar 2014.
[3] Siehe http://www.w3.org/RDF/, aufgerufen am 20. Februar 2014.
[4] Siehe http://www.w3.org/2001/sw/wiki/OWL, aufgerufen am 20. Februar 2014.
[5] Siehe http://protege.stanford.edu/, aufgerufen am 20. Februar 2014.

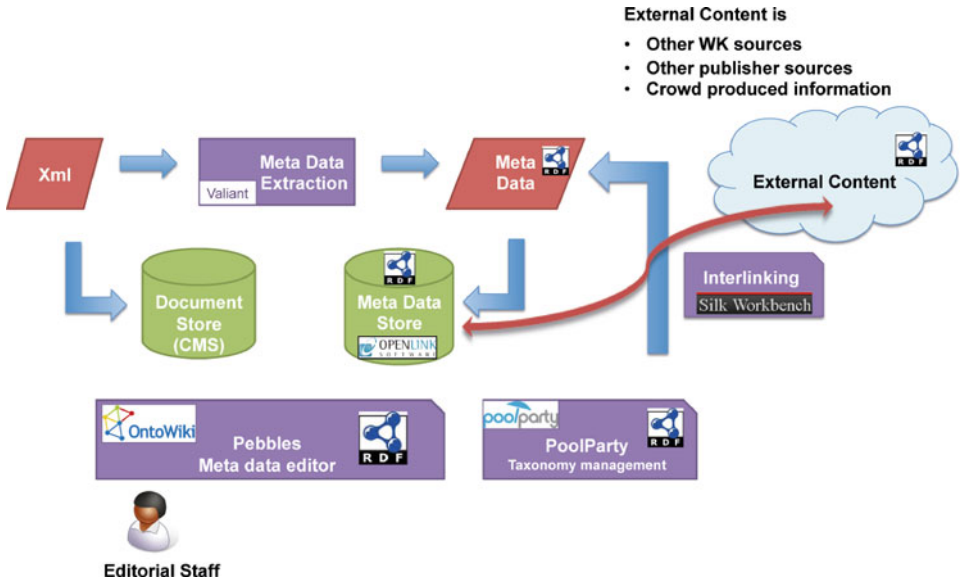


Abb. 14.2 Zukünftige Datenflüsse bei Wolters Kluwer Deutschland

In der Umsetzung wurden, wie in Abb. 14.2 gezeigt, folgende Schritte durchgeführt:

1. Analyse der XML-DTD zur Identifikation der relevanten Metadaten
2. Dokumentation der Semantik der relevanten Metadaten
3. Mapping der Metadaten zu Schemata
4. Festlegung der kontrollierten Vokabulare
5. Extraktion der kontrollierten Vokabulare und Laden in den SKOS-Thesaurus-Manager PoolParty[6]
6. Bereinigung der kontrollierten Vokabulare
7. Extraktion aller relevanten Metadaten und Ablage im Triple Store Virtuoso[7] unter Nutzung der kontrollierten Vokabulare
8. Verbindung der kontrollierten Vokabulare in PoolParty mit externen Datenquellen über das Mappingframework SILK[8]
9. Verbindung weiterer Metadaten in Virtuoso mit externen Datenquellen über das Mappingframework SILK
10. Aufbau einer prototypischen Nutzerschnittstelle zum Zugriff auf und zur Verwaltung von Metadaten in Virtuoso basierend auf OntoWiki[9]
11. Publizierung von juristischen Thesauri als LOD-Daten mit Hilfe von PoolParty

[6] Siehe: http://poolparty.biz/, aufgerufen am 20. Februar 2014.
[7] Siehe: http://www.openlinksw.com/, aufgerufen am 20. Februar 2014.
[8] Siehe: http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/, aufgerufen am 20. Februar 2014.
[9] Siehe: http://aksw.org/Projects/OntoWiki.html, aufgerufen am 20. Februar 2014.

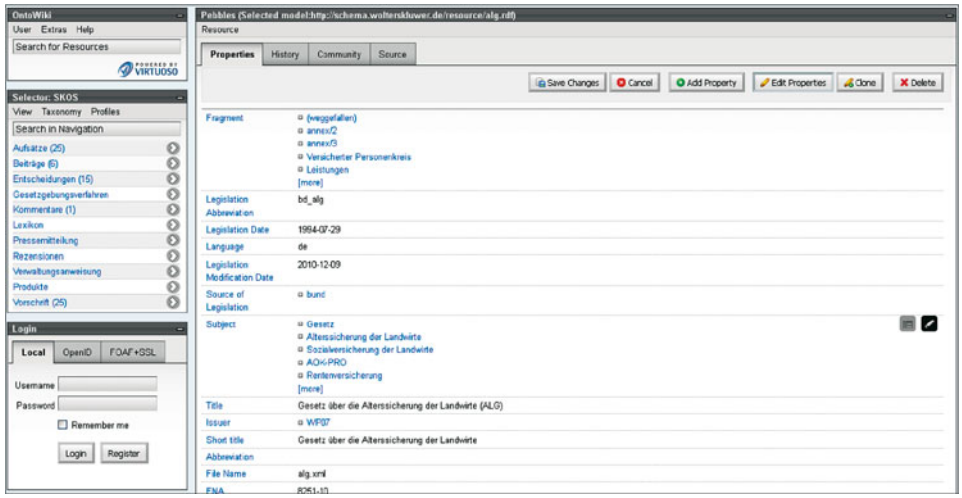


Abb. 14.3 Prototypische Redaktionsschnittstelle zur Verwaltung der Metadaten

Mit diesen Schritten wurde ein völlig neues Metadatenökosystem geschaffen, das es sukzessive ermöglicht, die alte Redaktionsinfrastruktur zu modernisieren und neue Anforderungen aus den Produkten wie Personalisierung oder semantische Suche zu adressieren.

Die Verwaltung der kontrollierten Vokabulare in PoolParty erfolgt so wie die Metadatenverwaltung in Virtuoso bereits operativ innerhalb der erweiterten existierenden Infrastruktur. Die Redaktionsschnittstelle ist prototypisch realisiert (vgl. Abb. 14.3) und wird in einem eigenen Implementierungsprojekt im Jahr 2014 sukzessive umgesetzt.

14.4 Zusammenfassung und Ausblick

Das Semantic Web soll Inhalte besser erschließbar und damit nutzbar machen. Dieses Ziel deckt sich mit einer Hauptaufgabe eines Medienunternehmens wie Wolters Kluwer.

Aus diesem Grund lag die Hauptherausforderung zur Durchführung des Projektes weniger darin, das Management von der grundsätzlichen Notwendigkeit zu überzeugen, sondern darin, dass die eingesetzten Technologien für die beteiligten Kollegen völlig neu waren. Somit mussten und müssen auch weiterhin unternehmensintern neue Fertigkeiten aufgebaut werden. Die Einbettung in das LOD2-Projekt und damit die Nutzung externer Ressourcen für den Basisprototypen hat die Aufgabe zwar etwas erleichtert, aber die Umsetzung als Realsystem bleibt trotzdem eine große Herausforderung.

Insgesamt wurden in den letzten drei Jahren Systeme prototypisch entwickelt, die das Potenzial haben, vorhandene Anforderungen an zukünftige semantische Systeme besser zu adressieren als klassische Lösungen. In Teilbereichen wie der Verwaltung und Pflege von kontrollierten Vokabularen, beispielsweise domänenspezifischen Taxonomien und

Thesauri, gibt es industriell nutzbare Tools wie PoolParty auf dem Markt. Ebenso ist die Speicherung und flexible Verarbeitung von Metadaten in Datenbanken wie Virtuoso inzwischen performant und sicher möglich. Die Schaffung eines Gesamtökosystems für Metadaten mit Schnittstellen zu verschiedenen Quellen, den damit verbundenen Fragen zur Lizenzierung und zur Herkunft sowie der Nachhaltigkeit und Pflegbarkeit der Vernetzungen stellt noch eine echte Herausforderung für die nächsten Jahre dar. Wenn man allerdings mit ausgesuchten Quellen arbeitet und die Verbindungen jeweils in einer 1 : 1-Relation qualitätsgesichert manuell oder semi-automatisch schafft, können auch heute schon wesentliche Netzwerkeffekte generiert werden.

Die Verankerung von Linked Data in Geschäftsmodellen muss als ganzheitliche Aufgabe betrachtet werden: Zukünftige Systeme und Lösungen erfordern eine koordinierte Zusammenarbeit von Lieferanten und Konsumenten von Wissensbasen. Nur so können Konzepte und Applikationen von einem rein wissenschaftlichen Betätigungsfeld auf ein business-orientiertes Niveau gehoben werden. Die Publizierung der juristischen Fachthesauri durch Wolters Kluwer[10] sollte als ein aktiver Beitrag für anstehende Diskussionen und Lösungsansätze angesehen werden.

Literatur

1. Auer, Sören, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N. Mendes, Bert van Nuffelen, Claus Stadler, Sebastian Tramp, und Sebastian Williams. 2012. Managing the Life-Cycle of Linked Data with the LOD2 Stack. *The Semantic Web – ISWC 2012*. Proceedings of the 11th International Semantic Web Conference, Boston, MA, USA, November 11–15, 2012, Lecture Notes in Computer Science Volume 7650, pp 1–16

[10] Siehe: http://lod2.wolterskluwer.de/, aufgerufen am 20. Februar 2014.