15 Metadaten

Im vorigen Kapitel wurde beschrieben, wie die Struktur von Dokumenten bei der Suche verwendet werden kann. Ein anderer Ansatz, die Suche zu verbessern, besteht darin, zusätzliches Wissen zu nutzen, das *über* die Dokumente existiert. Das geschieht auch, wenn Literaturdatenbanken oder Kataloge verwendet werden. Diese Hilfsmittel sind aber zugleich zentralisierte Verweise auf die Dokumente einer Sammlung, mit denen der Zugriff organisiert wird, indem z. B. ein Standort in einer Bibliothek oder die genauen Veröffentlichungsdaten angegeben werden.

Bei der automatischen Verarbeitung von Dokumenten (oder allgemeiner: digitalen Objekten) kann dieser Zugriff gut über Programme organisiert werden, wenn die Objekte in einer Form beschrieben sind, die von Rechnern verarbeitet werden kann. Wenn solche Beschreibungen auch Daten über die Qualität, die Art der Veröffentlichung und die Zugriffsmöglichkeiten enthalten, können auch Dokumente, die nicht in einer extra aufgebauten und gepflegten Sammlung enthalten sind, gefunden werden. Daher ist es für digitale Objekte wichtiger, dass gute maschinenlesbare Beschreibungen existieren und gefunden werden können, als dass sie bereits als Kataloge organisiert sind. Solche Beschreibungen von Objekten bezeichnet man als *Metadaten*, also Daten über das eigentliche Dokument.

Mit SGML, XML und anderen Strukturierungsverfahren können verschiedene Textteile unterschieden werden. So können auch Metadaten, die ja eigentlich nicht zum Inhalt eines Textes gehören, in einem Dokument abgelegt werden, wie z. B. die Daten im `Head`-Element eines HTML-Dokuments. Durch die lange Tradition von Archiven und Bibliotheken gibt es insbesondere für wissenschaftliche Artikel und Bücher eine Vielzahl von bibliografischen Formaten, die Metadaten beschreiben. Dabei steht die formale Erfassung bibliografischer Daten im Vordergrund. Sie enthalten aber auch Daten, die den Inhalt beschreiben, wie Stichwortlisten oder Klassifikationen. Solche Daten werden übrigens auch bei gedruckten Büchern häufig in den Büchern selbst abgedruckt, z. B. als *Library of Congress Cataloging-in-Publication Data*. Die klassischen Bibliotheksformate sind allerdings für digitale Objekte oder Dokumente nicht unbedingt geeignet.

15.1 Dublin-Core-Metadaten

Als ersten Schritt hin zur einheitlichen Beschreibung von digitalen Objekten durch Metadaten hat man sich Mitte der 1990er Jahre zunächst auf so genannte *Document Like Objects* (DLO) beschränkt. Ein DLO steht für die digitale Form dessen, was man bisher als Dokument kannte. Es wurde zunächst ganz bewusst nicht genauer definiert. Diese Überlegungen zur Entwicklung eines Metadatenformats für DLO gehen auf einen Workshop zurück, der 1995 in Dublin, Ohio stattfand (Weibel, Godby, Miller und Daniel, 1996 [125]), um über Wege zur Beschreibung von Objekten im Internet nachzudenken.

Nach diesem Ort ist eine Sammlung von Metadatenelementen als *Dublin Core* benannt worden, die – wie der Name sagt – einen Kern von Angaben bilden soll, mit dem ein DLO beschrieben werden kann. Ziel der Überlegungen war es, eine Menge von einfachen Elementen zu definieren, deren Namen möglichst intuitiv sind oder durch kurze Definitionen erläutert werden können. Dahinter stand die Überlegung, dass die große Mehrzahl der Web-Dokumente nicht von Expertinnen und Experten beschrieben werden können, sondern dass die Metadaten durch die Autorinnen und Autoren zur Verfügung gestellt werden müssen. Das *Dublin Core Element Set* wird von der *Dublin Core Metadata Initiative* (DCMI) [26] entwickelt. Es enthält zurzeit die folgenden Elemente:

- **Title**: Namen des Objekts.
- **Creator**: Personen, Organisationen oder Dienste, die in erster Linie für den Inhalt des Objekts verantwortlich sind, z. B. Autorinnen oder Autoren.
- **Subject**: Thema (*topic*) des Objekts, typischerweise Stichwörter, Deskriptoren oder Elemente eines Klassifikationssystems.
- **Description**: Beschreibung des Inhalts des Objekts als Text, z. B. als Abstract oder Inhaltsverzeichnis.
- **Publisher**: Personen oder Organisationen, die dafür verantwortlich sind, das Objekt zugänglich zu machen.
- **Contributor**: Personen oder Organisationen, die wesentliche Beiträge zum Inhalt des Objekts geleistet haben, aber nicht unter Creator genannt sind (Herausgebende, Übersetzerinnen, Illustratoren).
- **Date**: Datum von Ereignissen, die mit dem Objekt verbunden sind, wie das Veröffentlichungsdatum.
- **Type**: Art oder Genre des Objektinhalts (z. B. Erzählung, Gedicht oder Lexikon).
- **Format**: Physisches oder digitales Format des Objekts (wie PostScript oder ausführbares Programm, aber auch Dauer und Größe).
- **Identifier**: Zeichenkette oder Nummer, mit der das Objekt in einem Kontext eindeutig identifiziert werden kann (URL, URI, ISBN, DOI).
- **Source**: Informationen über Objekte, aus denen das zu beschreibende Objekt abgeleitet wurde.

- **Language**: Sprache des Inhalts des Objekts.
- **Relation**: Beziehung zu anderen, verwandten Objekten.
- **Coverage**: Räumliche und zeitliche Charakteristika des Objektinhalts.
- **Rights**: Informationen über (Urheber- und Verwertungs-)Rechte an dem Objekt und seinem Inhalt bzw. die Inhaber dieser Rechte.
- **Audience**: Zielgruppe, für die das Objekt erstellt wurde oder für die es nützlich ist.

Die aktuelle Version findet sich auf der Dublin Core Homepage [26]. Die erste Version und ihre Entstehung wurde von Weibel, Godby, Miller und Daniel (1996) [125] beschrieben. Abbildung 15.3 zeigt ein Beispiel für eine Dublin-Core-Beschreibung.

Man sieht, dass sich viele der Elemente an bibliografischen Daten für »Papierdokumente« orientieren, wobei die möglichen Einträge für digitale Dokumente angepasst sind (Format, Identifier). Die Definition der Metadatenelemente liegt unterdessen in zahlreichen Sprachen vor, wobei die Elementnamen nicht verändert werden. Bei der Entwicklung der Beschreibung in anderen Sprachen (*internationalization*) muss entschieden werden, wieweit der ursprünglich englische/amerikanische Text an lokale Traditionen und Bedingungen der Länder, in denen die Sprachen gesprochen werden, angepasst werden kann und muss (*localization*).

Neben der Benennung der Elemente wurde bereits früh eine Reihe von Eigenschaften der Elemente bzw. Prinzipien ihres Auftretens in einer Metadatenbeschreibung festgelegt:

- Dublin-Core-Elemente beschreiben *intrinsische* Eigenschaften von Objekten, also Eigenschaften des Objekts selbst. Dieses Prinzip wurde allerdings nicht vollständig durchgehalten.
- *Erweiterbarkeit* der Elementsammlung. Weitere Elemente müssen nicht von allen Systemen, die mit Dublin Core arbeiten, verstanden werden. Die Systeme müssen die Existenz unbekannter Elemente aber tolerieren.
- *Unabhängigkeit von einer spezifischen Syntax*: Hier war die Überlegung vor allem, dass es zum damaligen Zeitpunkt noch zu früh war, eine spezielle Syntax festzulegen. Wieweit eine genaue Spezifikation von Formaten und zu verwendenden inhaltlichen Beschreibungen sinnvoll ist, hängt von der Zielsetzung und Anwendung ab. Je genauer die Formate spezifiziert werden, desto besser können sie verglichen werden, desto größer ist aber auch die Gefahr, dass andere Systeme sie nicht mehr interpretieren können (mangelnde Interoperabilität).
- *Optionalität*: Dublin-Core-Elemente können benutzt werden, sie müssen es aber nicht. Es gibt keine Elemente, die immer angegeben werden müssen. Das hat zwei Gründe: Zum einen können für neuartige Objekte manche Elemente, die jetzt noch sehr zwingend erscheinen, wenig sinnvoll sein, zum anderen wollte man Autorinnen und Autoren nicht durch

komplizierte Vorschriften abschrecken. Eine kurze Beschreibung ist besser als gar keine.

- *Wiederholbarkeit*: Alle Elemente können mehrmals in einem Datensatz auftreten. So können mehrere Autorinnen aufgeführt oder verschiedene Relationen zu verwandten Objekten angegeben werden.
- *Veränderbarkeit*: Jedes Element kann durch die Angabe eines Attributs verändert werden, wenn dadurch eine spezifische Interpretation des Inhalts vorgegeben wird. So können z. B. in einem Gebiet bewährte Inhaltsklassifikationen in Dublin Core übernommen werden. Dazu wird ein Verweis auf das Klassifikationsschema angegeben.

Durch diese Prinzipien soll Dublin Core klein und flexibel gehalten werden. Gleichzeitig soll es möglich sein, vorhandene Metadatenschemata zu integrieren. Dublin Core hätte damit eine Integrationsfunktion für verschiedene Auszeichnungs- und Metadatenschemata.

Nach dem ersten Workshop in Dublin (Ohio) haben weitere Workshops stattgefunden, die sich mit spezifischen Problemen bei der Entwicklung des Formats befasst haben. Dabei hat sich gezeigt, dass die Ansichten über einige der oben aufgeführten Prinzipien durchaus auseinander gehen. Insbesondere bei der Frage, wie weit genaue Formate und ausgearbeitete Strukturen bei den einzelnen Elementen vorgegeben werden sollen, gibt es (mindestens) zwei Positionen: Während die eine vor allem die einfache Anwendbarkeit und die Austauschbarkeit (Interoperabilität) betont und deshalb keine komplexen Strukturen einführen will, betont die andere die Notwendigkeit, durch genauere Vorgaben den Nutzen der Beschreibungen zu erhöhen. Insbesondere soll es möglich sein, vorhandene und bewährte Beschreibungsstandards auch in Dublin Core zu benutzen.

Die erste Position sieht Dublin-Core-Metadaten als eine kleinste gemeinsame Basis zwischen verschiedenen Auszeichnungsformaten, mit der deren Gemeinsamkeiten genutzt werden sollen. Baker (1998) [8] vergleicht diese Rolle mit dem Pidgin-Englisch, das sich entwickelte, als sich Sklaven mit verschiedenen Muttersprachen bei der Arbeit verständigen mussten. Die so entstandene Sprache beschränkte sich zunächst auf die notwendigsten Begriffe, entwickelte sich im Laufe der Zeit aber zu einem immer vollständigeren Verständigungsmittel.

Die zweite Position legt mehr Wert auf die Nutzung bewährter Beschreibungsmittel und -methoden. Solche komplexen Beschreibungen können zwar in der Regel nur von geschulten Personen erstellt und von spezialisierten Suchprogrammen genutzt werden; sie können dafür aber wesentlich genauer sein und vorhandene Beschreibungen lassen sich weiterverwenden.

Aus diesen zwei Positionen haben sich zwei Sichten oder Versionen gebildet: Als *unqualified Dublin Core* wurden nur die Elementbeschreibungen ohne weitere Vorgaben über die Formate und zu verwendenden Beschreibungen bezeichnet. Bei *qualified Dublin Core* wurden *Spezialisierungen* (*Refinements*) und Inhalts- bzw. Formatvorgaben (*encoding schemes*) vorgegeben. Die Dublin-

Core-Empfehlung *Dublin Core Qualifiers* vom 11. Juli 2000 [27] enthielt die im Folgenden dargestellten Spezifizierungsempfehlungen. Sie wurden unterdessen als *Refinements* zur Liste der DCMI-Elemente hinzugefügt (siehe [28]):

- **Title**: *Alternative* (zweiter Titel, der als Ersatz für den »Haupttitel« verwendet werden kann)
- **Description**: *Table of Contents* (Inhaltsverzeichnis), *Abstract* (Zusammenfassung)
- **Date**: *Created* (erzeugt), *Valid* (gültig), *Available* (verfügbar), *Issued* (veröffentlicht), *Modified* (geändert)
- **Format**: *Extent* (Größe oder Dauer), *Medium* (Material oder Datenträger)
- **Relation**: *Is Version Of* (ist eine Version von), *Has Version* (hat eine Version), *Is Replaced By* (wurde ersetzt durch), *Replaces* (ersetzt), *Is Required By* (wird benötigt von), *Requires* (benötigt), *Is Part Of* (ist Teil von), *Has Part* (hat Teil), *Is Referenced By* (es wird darauf verwiesen von), *References* (verweist auf), *Is Format Of* (der gleiche Inhalt wird in anderem Format dargestellt von), *Has Format* (ist eine Darstellung in anderem Format von),
- **Coverage**: *Spatial* (räumlich), *Temporal* (zeitlich)

und folgende Formatempfehlungen:

- **Subject**: *LCSH* (Library of Congress Subject Headings), *MeSH* (Medical Subject Headings), *DDC* (Dewy Decimal Classification), *LCC* (Library of Congress Classification), *UDC* (Universal Decimal Classification)
- **Date**: *DCMI Period* (Dublin-Core-Format für Zeiträume), *W3C-DTF* (W3C-Format für Datum und Zeit, basierend auf ISO 8 601)
- **Type**: *DCMI Type Vocabulary* (Dublin-Core-Vokabular für Dokumenttypen)
- **Format**: *IMT* (Medientyp (Internet Media Type))
- **Identifier**: *URI* (Universal Resource Identifier)
- **Language**: *ISO 639-2* (ISO-Norm mit Buchstaben-Codes für Sprachen), *RFC 1766* (Erweiterung der ISO-Norm mit Ländercodes)
- **Relation**: *URI* (Uniform Resource Locator)
- **Coverage (räumlich)**: *DCMI Point* (Dublin-Core-Spezifikation nach räumlichen Koordinaten), *ISO 3166* (ISO-Spezifikation für Ländernamen), *DCMI Box* (Dublin-Core-Definition einer geografischen Fläche), *TGN* (Getty-Thesaurus geografischer Namen)
- **Coverage (zeitlich)**: *DCMI Period* (Dublin-Core-Format für Zeiträume), *W3C-DTF* (W3C-Format für Datum und Zeit basierend auf ISO-8601)

Zu Dublin-Core-Metadaten gab es bereits 1996 einen DTD-Entwurf, mit dem eine Metadatenbeschreibung als SGML-Dokument notiert werden konnte. Häufiger werden Dublin-Core-Metadaten aber in den Meta-Tags im Head

von HTML-Seiten als Attribut-Wert-Paare verwendet. Dabei wird als Namensraumkennung das Kürzel `dc:` verwendet. Dublin Core ist auch eine Referenzanwendung in der Entwicklung des Resource Description Framework (RDF), das am Ende dieses Kapitels beschrieben wird (Abschnitt 15.4).

15.2 Hierarchisch strukturierte Metadaten

Während Dublin Core lediglich eine Liste von Elementen zur Beschreibung von Document Like Objects zur Verfügung stellt, gibt es Metadatenstandards, die wesentlich komplexer strukturierte Beschreibungsformate definieren. Als Beispiel soll hier die von der IEEE und anderen entwickelte *Learning-Object-Metadata-Spezifikation* (LOM-Spezifikation) dienen [75]. Dabei handelt es sich um eine hierarchisch strukturierte Beschreibung mit neun Top-Level-Elementen, die sich aus Unterelementen zusammensetzen, also eine Datenstruktur, wie sie durch eine DTD beschrieben wird.

Eine solche DTD (auch *XML-Binding* genannt) für LOM wurde vom IMS – Global Learning Consortium [60] entwickelt, einer Organisation, in der sich namhafte Interessengruppen und Software-Produzenten aus dem Bereich computergestütztes Lernen (*CBT*, *Computer Based Training*) zusammengeschlossen haben, um gemeinsame Modelle und Spezifikationen zu vereinbaren.

Abbildung 15.1 zeigt die neun Top-Level-Elemente und gibt jeweils eine kurze Inhaltsbeschreibung. Jedes dieser Elemente besteht aus Unterelementen, die sich ihrerseits aus weiteren Unterelementen zusammensetzen können. Die Elemente in LOM sind bis zu vier Stufen tief geschachtelt. Abbildung 15.2 zeigt exemplarisch einen Zweig des Baums, der von LOM definiert wird.

Durch diese Datenstruktur werden die Metadaten in kleine, wohl definierte Einheiten – im Folgenden wieder Blattelemente genannt – zerlegt. Diese Zerlegung ist für die maschinelle Verarbeitung der Daten wichtiger als für das Verständnis durch Menschen, da Menschen in der Regel wesentlich besser in der Lage sind, aus dem Inhalt und dem Kontext zu schließen, um welche Art von Daten es sich bei einem Eintrag handelt. So können Menschen z. B. Adressen meist in ihre Bestandteile wie Vorname, Nachname, Straße, Hausnummer, Postleitzahl, Ort, etc. zerlegen (zumindest, wenn es sich um Angaben aus einem vertrauten Land oder Kulturkreis handelt), während Maschinen im Allgemeinen eine genaue Spezifikation dieser Bestandteile brauchen, um sie zu erkennen.

Für die Suche bedeutet eine solche stark strukturierte Darstellung der Einträge, dass Anfragen und Einträge in den einzelnen Blattelementen mit jeweils genau auf den Datentyp und die enthaltene Datenart abgestimmten Methoden verglichen werden können. So lassen sich Schwierigkeitsgrade z. B. auf eine Ordinalskala abbilden, die es erlaubt zu bestimmen, dass ein Kurs schwieriger ist als der andere oder als ein in einer Anfrage angegebener Grad.

Learning Object Metadata	
1. General	*identifier, title, location, language, aggregation*
2. Life Cycle	*author, version, contribution(s)*
3. Meta MetaData	*information on this metadata description*
4. Technical	*size, format, technical requirements, etc.*
5. Educational	*interactivity-type, difficulty-level, target group*
6. Rights	*copyright, cost, conditions of use*
7. Relation	*references to other learning objects*
8. Annotation	*remarks, comments, third party evaluations*
9. Classification	*descriptions using existing external classifications*

Abbildung 15.1 – Die neun Top-Level-Elemente der LOM-Spezifikation: Unter den neun Top-Level-Elementen ist in Stichworten die Art der Information angegeben, die im jeweiligen Element zusammengefasst wird. Die einzelnen Elemente setzen sich aus bis zu vier Stufen von Unterelementen zusammen.

Struktur des 2. LOM-Elements LifeCycle
2. LifeCyle
2.1 Version
2.2 Status
2.3 Contribute
2.3.1 Role
2.3.2 Entity
2.3.3 Date

Abbildung 15.2 – Ein Zweig der LOM-Spezifikation: Als Beispiel eines tief strukturierten Top-Level-Elements ist das zweite Element `LifeCycle` dargestellt. Es enthält zunächst Angaben über die Version und den Status (also beispielsweise `Draft`, `Final`, `Published` oder `Unavailable`). Mit dem Element `2.3 Contribute` können über das Unterelement `Role` Personen oder Institutionen aufgeführt werden, die zum Entstehen beigtragen haben, wie Autorinnen, Herausgeber oder Übersetzerinnen. Es können aber auch Institutionen wie Institute oder Verlage angegeben werden.

Kosten können auf einer Rationalskala dargestellt werden, mit der bestimmt werden kann, dass ein Kurs doppelt so teuer ist wie ein anderer. Die Kosten mehrerer Kurse lassen sich aufsummieren und es kann geprüft werden, ob die Summe einen Maximalbetrag, der in einer Anfrage angegeben ist, nicht übersteigt.

Namen von Autoren haben in der Regel nur Nominalskalenniveau, es kann also nur gesagt werden, ob sie gleich oder verschieden sind. Bei der Suche können allerdings auch hier Ähnlichkeitsfunktionen sinnvoll sein, die z. B.

Schreibfehler ausgleichen oder gleich bzw. ähnlich klingende Namen (Mayer, Maier oder Meier) finden können. Ein häufiges Problem bei Namen sind auch unterschiedliche Übertragungen (Transliterationen) von Namen aus anderen Alphabeten. So wurden z. B. für den russischen Mathematiker Tschebyschew in der Literatur über 20 verschiedene Schreibweisen mit lateinischem Alphabet gefunden (von denen die hier benutzte natürlich nur eine ist). Es kann für die Suche sehr hilfreich sein, diese Schreibweisen als Synonymmengen zu betrachten.

In der Regel soll durch eine Ähnlichkeitssuche auf Namen festgestellt werden, ob zwei Namen dieselbe Person bezeichnen oder nicht. Die Ähnlichkeit der Schreibweisen gibt also allenfalls die Sicherheit an, mit der ein Name gefunden wurde, sie gibt keine Ähnlichkeit zwischen Personen an. Sie würde also sinnvollerweise in einem Ansatz verwendet, der mit unscharfen Mengen arbeitet.

Vergleiche oder Ähnlichkeiten in den einzelnen Blattelementen sind aber nur die einzelnen Bestandteile, aus denen eine komplexere Suche zusammengesetzt werden muss. Sie entsprechen den elementaren Anfragen, die in Abschnitt 3.1.1 über die Logik der booleschen Suche beschrieben wurden. Sie können auch in diesem Sinne verwendet werden, wenn es sich um elementare boolesche Anfragen handelt. Man kann aber auch elementare Ähnlichkeitsanfragen definieren, die entsprechend einen Ähnlichkeitswert liefern. Diese elementaren Ähnlichkeitsanfragen können zu komplexen Ähnlichkeitswerten kombiniert werden, indem sie z. B. gewichtet aufsummiert werden. Es sind aber auch komplexere Formeln oder Algorithmen möglich, mit denen z. B. ganze (in sich wiederum strukturierte) Metadatenelemente (also Teilbäume oder Äste der durch die DTD definierten hierarchischen Datenstruktur) gewichtet werden, wie sie z. B. in Abschnitt 14.4.4 beschrieben wurden.

In der LOM-Spezifikation sind viele der Werte der Elemente als Zeichenketten oder Auswahlen definiert. Die Einträge können in geordneten und ungeordneten Listen verknüpft werden. Bis auf die geordneten Listen sind also alle Daten auf Nominalskalenniveau definiert. Auch die von IMS entwickelte LOM-DTD geht kaum über diese Skalenniveaus hinaus, was letztlich daran liegt, dass genauere Datentypen in SGML und XML nicht vorgesehen sind und die Möglichkeiten von XML Schema noch nicht genutzt wurden. Höhere Skalenniveaus können also gegenwärtig zwar bei Zahlen oder Ordinaldaten angenommen werden, sind aber nicht Teil der LOM- oder IMS-Spezifikation. Das heißt aber auch, dass sie zwar innerhalb einer Anwendung oder Sammlung, die einheitlich gepflegt wird, verwendet werden können, dass aber, sobald es um Austauschbarkeit (Interoperabilität) zwischen Anwendungen geht, eine einheitliche Nutzung dieser höheren Skalenniveaus noch nicht vorausgesetzt werden kann.

Es zeigt sich, dass stark strukturierte Metadatenmodelle zwar enorme Möglichkeiten bieten, komplexe Suchverfahren zu definieren, sie benötigen dafür aber auch konsistente Datensammelungen, die entsprechend aufwändig zu erstellen und zu pflegen sind. Nur wenn die einzelnen Elemente mit der

gleichen Semantik und Syntax verwendet werden, kommen die Vorteile der Datentyp-spezifischen Vergleichsmöglichkeiten zum Tragen. Detailliert strukturierte Metadaten sind damit näher am Konzept der Faktendatenbanken als an dem durch Vagheit bestimmten Konzept des Information Retrieval.

Anfragen, die die Strukturierung der Daten wirklich nutzen, müssen sehr spezifisch formuliert werden. Das heißt, dass entweder die Nutzenden diese Struktur selbst gut kennen müssen, oder dass Suchmaschinen den Informationsbedarf der Nutzenden entsprechend auf die Metadatenstruktur abbilden müssen. Das ist insbesondere bei unstrukturierten Anfragen, wie sie von ungeübten Nutzenden gestellt werden, häufig schwierig. In vielen Fällen sollten solche Anfragen auf viele Elemente der Metadaten angewendet werden, sodass die detaillierte Struktur in diesen Fällen nicht genutzt wird.

15.3 PICS

Einer der ältesten Ansätze, Web-Dokumente mit Metadaten auszuzeichnen, ist *PICS*, die *Platform for Internet Content Selection* [91]. Diese Spezifikation wurde zunächst vor allem entwickelt, um Seiten im Web zu kennzeichnen, deren Inhalt als nicht für Kinder und Jugendliche geeignet eingeschätzt wurde. Sie ermöglicht aber als standardisierte Plattform prinzipiell, beliebige Kennzeichnungsschemata einzusetzen. PICS ist in dieser Hinsicht ein Vorläufer des im nächsten Abschnitt beschriebenen Resource Description Framework (RDF) und hat dessen Entwicklung wesentlich beeinflusst.

Das Ziel der Kennzeichnung ist es, mit einem entsprechenden Filter bei den Endnutzenden oder auch bei einem Zugangspunkt zum Web Angebote blockieren zu können, die z. B. als jugendgefährdend gekennzeichnet sind (Negativauswahl), oder aber auch nur solche Angebote weiterzuleiten bzw. anzuzeigen, die – eventuell von einer bestimmten Organisation – als nicht jugendgefährdend eingestuft sind (Positivauswahl).

Mechanismen, die dazu dienen, bestimmte Seiten für bestimmte Gruppen zu sperren, sollten in demokratischen Gesellschaften natürlich kritisch beurteilt werden. Man kann bei solchen Beurteilungen zwischen der *Selbstbeurteilung* (*self-rating*), bei der die Anbieter ihre eigenen Inhalte einschätzen und kennzeichnen, und der *Beurteilung durch Dritte* (*third-party-rating*), bei der Personen, Agenturen oder Organisationen Angebote im Web beurteilen und kennzeichnen, unterscheiden. Die Verfechter dieses Ansatzes sind der Überzeugung, dass eine offene Plattform für Kennzeichnungen, die den Nutzenden (bzw. deren Eltern) die Möglichkeit gibt, am eigenen Rechner eine Auswahl der anzeigbaren Angebote zu treffen, am besten dazu geeignet ist, die Vielfalt der Angebote im Web zu sichern. Nutzende (bzw. deren Eltern) hätten so die Möglichkeit, ihrer Auswahl die Beurteilungen der Institutionen zugrunde zu legen, von denen sie glauben, dass sie für sie (oder ihre Kinder) am besten geeignet sind.

PICS kann aber auch von *Providern* (Anbietern von Internet-Zugang und -Speicherplatz) dazu genutzt werden, »vorgefilterte« Seiten anzubieten. Diese Tendenz wird z. B. dann zunehmen, wenn Provider für die Seiten, auf die man über sie zugreifen kann, strafrechtlich verantwortlich gemacht werden. Es gibt auch öffentliche Büchereien, die Internet Terminals zur Verfügung stellen, mit denen nur »freigegebene« Seiten erreicht werden können, die also eine Positivauswahl treffen. Das kann man in Analogie zur Beschaffungspolitik einer Bücherei sehen, die bestimmte Bücher auswählt, man kann es aber auch als Zensur verstehen. Die Interpretation als Analogie zur Beschaffungspolitik ist insofern problematisch, weil nicht finanzielle oder räumliche Restriktionen zur Auswahl zwingen, sondern explizit bestimmte Inhalte ausgeschlossen werden sollen.

15.4 RDF und das Semantische Web

In Abschnitt 15.2 über strukturierte Metadaten wie LOM waren die Vor- und Nachteile einer detaillierten Datenstrukturierung bereits angesprochen worden. Für die maschinelle Verarbeitung sind wohl definierte Datentypen und Formate häufig nützlich, zumindest wenn es sich um Daten handelt, auf die mit Hilfe der Formate spezielle Vergleichsverfahren angewendet werden können. Um solche Verfahren aber sinnvoll einsetzen zu können, muss auch die Semantik der Daten, also ihre Rolle oder Bedeutung in einer Metadatenbeschreibung, bekannt sein.

15.4.1 Resource Description Framework

Ein allgemeines Modell, das eine Syntax zur Beschreibung der Semantik von Metadaten zur Verfügung stellt, ist vom W3C mit dem *Resource Description Framework* (RDF) definiert worden. »Resource« wird dabei als ein unscharfer Begriff für physische oder konzeptuelle Objekte verwendet. Das Glossar der W3C RDF Syntax Recommendation (1999) [72] definiert Resource als »*An abstract object that represents either a physical object such as a person or a book or a conceptual object such as a color or the class of things that have colors. (. . .)*«. Im Folgenden wird der Begriff als Objekt übersetzt. Diese Übersetzung kollidiert allerdings leider mit dem Begriff Objekt, wie er in der Grammatik verwendet wird (siehe unten). Daher wird auch der Begriff *Ressource* verwendet, auch wenn das dem deutschen Sprachgebrauch eigentlich nicht entspricht, oder das Wort Objekt wird durch das englische »resource« spezifiziert, um es vom Objekt aus der Grammatik zu unterscheiden.

RDF ist unter anderem mit dem Ziel entwickelt worden, Metadaten möglichst eindeutig und unabhängig von einem bestimmten Wissensgebiet zu beschreiben, ihre maschinelle Verarbeitung und Nutzung zu unterstützen und den Austausch von Metadaten zwischen unterschiedlichen Gruppen und Anwendungen zu fördern. RDF wurde mit Hilfe von XML definiert, wobei XML

aber lediglich ein Hilfsmittel zur Beschreibung und Unterstützung sein soll, das prinzipiell auch durch andere Methoden ersetzt werden kann.

Das Datenmodell von RDF setzt sich aus drei Komponenten zusammen (siehe W3C RDF Model and Syntax Specification, 1999 [72]):

- Als *Objekte* oder *Ressourcen* (*resources*) werden alle »Dinge« bezeichnet, die mit RDF beschrieben werden. In der Regel sind das »Web-Objekte«, die mit einer URL und gegebenenfalls weiteren Anker- oder Elementangaben spezifiziert werden. Genauer genommen werden sie durch einen URI, also einen *Uniform Resource Identifier*, bestimmt. Mit diesem erweiterten Adressierungskonzept [10] können nicht nur Web-Seiten, sondern auch beliebige Objekte bezeichnet werden.
- Objekte können *Eigenschaften* (*properties*) haben, die ihre Charakteristika oder Aspekte beschreiben. Eigenschaften sollten eine festgelegte Bedeutung und wohl definierte zulässige Werte haben. Auch der Unterschied oder die Beziehung zu anderen Eigenschaften sollte klar abgegrenzt sein.
- Mit Objekten und Eigenschaften können in RDF *Aussagen* (*statements*) formuliert werden, indem einer Eigenschaft eines Objekts ein Wert zugewiesen wird. Dieser Wert kann entweder aus einer vorgegebenen Wertemenge stammen oder wiederum ein Objekt sein. Eine Aussage kann in diesem Sinne auch als ein Tripel aus *Subjekt*, *Prädikat* und *Objekt* aufgefasst werden. Dabei bezeichnet hier »Objekt« (wie oben angekündigt) den grammatikalischen Begriff, nicht die Ressource. Das Prädikat bezeichnet die Eigenschaft oder die Art der Aussage und das Subjekt schließlich die Ressource (oder eben das Objekt), über die etwas ausgesagt wird. Eine Aussage oder mehrere Aussagen über das gleiche Objekt werden zu einer *Beschreibung* (*description*) zusammengefasst.

Zu diesem Grundmechanismus stellt das Datenmodell von RDF drei verschiedene *Container* zur Verfügung, also Konzepte, mit denen Objekte oder Werte zusammengefasst werden können: *Bag* beschreibt eine ungeordnete Menge, wobei aber einzelne Objekte mehrmals vorkommen können (also keine Menge im mathematischen Sinne). *Sequence* ist eine geordnete Liste von Objekten und *Alternative* ist eine Menge, aus der ein Objekt ausgewählt werden muss. Diese Container können anstelle von einzelnen Objekten (Ressourcen) oder Werten verwendet werden. Dabei kann auch spezifiziert werden, ob sich eine Aussage auf den Container oder auf die einzelnen Objekte im Container bezieht.

Schließlich erlaubt das RDF-Datenmodell auch Aussagen höherer Ordnung – also Aussagen über Aussagen – , indem eine Aussage selbst als Objekt definiert wird. Der Inhalt einer solchen Aussage wird dann nicht mehr als durch die Metadaten beschriebene Tatsache (oder als wahr) angenommen, sondern als ein Objekt, über das Aussagen gemacht werden können.

Sowohl die Definition von Containern als auch die Definition einer Aussage als Objekt werden dadurch spezifiziert, dass der entsprechenden Beschreibung ein in RDF definierter Typ zugewiesen wird. Bei den Containern kann das durch die abgeleiteten XML-Elemente `bag`, `seq` (für Sequence) und `alt` (für Alternative) getan werden. Um eine Aussage als Objekt zu definieren, wird ihr eine Eigenschaft `type` mit dem Wert `statement` hinzugefügt.

Darstellungsformen von RDF

Mit den RDF-Mechanismen lassen sich einfache Attribut-Wert-Paare definieren, also z. B. eine Dublin-Core-Metadatenbeschreibung (siehe Abbildung 15.3). Wenn Aussagen aber entsprechend geschachtelt werden, können auch sehr komplexe Beschreibungen entstehen. Um diesen komplexen RDF-Beschreibungen gerecht zu werden, gibt es weitere Beschreibungsformate. Eine mögliche Darstellungsform ist ein semantisches Netz oder *RDF-Graph*, bei dem Objekte als Knoten und Eigenschaften als Kanten eines gerichteten Graphen gezeichnet werden. Diese Darstellungsform findet sich z. B. in den W3C-Dokumenten, bei Miller (1998) [84] und in Abbildung 15.4.

Die beschriebenen Mechanismen stellen zunächst nur die Syntax der Metadatenbeschreibung zur Verfügung. Die Bedeutung oder Semantik der Metadatenbeschreibung wird durch die Definition der Eigenschaften und der Werte, die die Eigenschaften annehmen können, bestimmt, also z. B. durch die Dublin-Core- oder LOM-Definitionen. Wie weit diese Bedeutung maschinell erfasst und verwendet werden kann, hängt von vielen Faktoren ab, insbesondere natürlich von den Methoden, mit denen das so formal repräsentierte Wissen verarbeitet wird. Dabei sind die in diesem Buch beschriebenen Information-Retrieval- und Wissensextraktionsmethoden sicherlich einfache Modelle, die dafür aber mit heterogenen Sammlungen noch verhältnismäßig gut umgehen können.

Bei der Entwicklung von Methoden zur Beschreibung der Inhalte können weitere Konzepte zur Inhaltsrepräsentation verwendet werden. In Frage kommen z. B. Ansätze, die auch schon in den Datenmodellen von XML Schema verwendet wurden, wie das Konzept der Vererbung oder Ansätze der unscharfen Wissensrepräsentation durch Fuzzy Sets sowie das Vektorraummodell.

RDF-Schema

RDF selbst soll mit der *RDF Vocabulary Description Language RDF Schema* eine Beschreibungssprache zur Verfügung stellen, in der nach ihrer Bedeutung strukturierte Vokabularien definiert werden können. Das geschieht dadurch, dass grundlegende Klassen und Eigenschaften definiert werden, mit denen die inhaltlichen Strukturen beschrieben und weitere Klassen und Eigenschaften (*properties*) definiert werden können. Die folgende Darstellung stützt sich auf den W3C Working Draft 1.0 vom 30. April 2002 [15], beschreibt

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/metadata/dublin_core#"
xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#">
<rdf:Description
about="http://www.dlib.org/dlib/may98/miller/05miller.html">
<dc:Title> An Introduction to the Resource Description Framework </dc:Title>
<dc:Creator> Eric J. Miller </dc:Creator>
<dc:Description> The Resource Description Framework (RDF) is an infrastructure
that enables the encoding, exchange and reuse of structured metadata. rdf is an
application of xml that imposes needed structural constraints to provide unambiguous
methods of expressing semantics. rdf additionally provides a means for publishing both
human-readable and machine-processable vocabularies designed to encourage the reuse
and extension of metadata semantics among disparate information communities. the
structural constraints rdf imposes to support the consistent encoding and exchange of
standardized metadata provides for the interchangeability of separate packages of
metadata defined by different resource description communities. </dc:Description>
<dc:Publisher> Corporation for National Research Initiatives </dc:Publisher>
<dc:Subject>
<rdf:Bag>
<rdf:li> machine-readable catalog record formats </rdf:li>
<rdf:li> applications of computer file organization and access methods
</rdf:li>
</rdf:Bag>
</dc:Subject>
<dc:Rights> Copyright @ 1998 Eric Miller </dc:Rights>
<dc:Type> Electronic Document </dc:Type>
<dc:Format> text/html </dc:Format>
<dc:Language> en </dc:Language>
<dc:Relation rdf:parseType="Resource">
<dcq:RelationType
rdf:resource=
"http://purl.org/metadata/dublin_core_qualifiers#IsPartOf"/>
<rdf:value
resource="http://www.dlib.org/dlib/may98/05contents.html"/>
</dc:Relation>
</rdf:Description>
</rdf:RDF>
```

Abbildung 15.3 – Dublin-Core-Beschreibung eines Artikels mit dem Resource Description Framework (RDF): Der Datensatz beschreibt Eric Millers Artikel über RDF im D-Lib Magazin (Miller, 1998 [84]). In den ersten vier Zeilen wird das Element `rdf:RDF` geöffnet. Dabei werden drei Namensräume spezifiziert. Als nächstes Unterelement wird mit `RDF-Description` eine RDF-Beschreibung spezifiziert, die die eigentlichen Dublin-Core-Elemente enthält. Im Subject Element `dc:Subject` wird der RDF-Container `rdf-Bag` verwendet, um zwei Einträge zu machen. Innerhalb des nicht qualifizierten Relation-Elements `dc:Relation` wird ein qualifiziertes Unterelement `dcq:RelationType` verwendet, das seinerseits wieder RDF-Elemente nutzt. Das Beispiel wurde der RDF-Model and Syntax Specification (1999) [72] entnommen.

also »work in progress« und keine verabschiedete Spezifikation oder W3C-Empfehlung.

Der theoretische Hintergrund der Vorschläge liegt in der Modelltheorie und wird von Hayes (2002) [55] in einem W3C Working Draft näher beschrieben. Dabei geht es darum, ein logisches Modell mit Hilfe eines Graphen (eben dem RDF-Graphen) zu beschreiben, das die »reale Welt« abbilden soll (bzw. genauer: in das die reale Welt abgebildet werden kann). Aus diesem Modell können dann Aussagen abgeleitet und Schlüsse gezogen werden. Das Modell ist also erheblich mehr als nur ein Format für die Beschreibung von Metadaten: Es kann zur Modellierung von Realitätsausschnitten genutzt werden, wendet dabei aber in seiner gegenwärtigen Form lediglich binäre Logik an. Die Inferenzmechanismen werden für die Dokumentsuche allerdings nicht genutzt. Wie weit sie in anderen Anwendungen sinnvoll eingesetzt werden können, scheint noch weitgehend offen.

Nach den Vorschlägen des RDF-Schema-Entwurfs (Bricklay und Guha, 2002 [15]) sollen die bereits in der RDF-Syntax eingeführten Objekte (*resources*), Eigenschaften (*properties*), Aussagen (*statements*) und Container `Bag`, `Seq` und `Alt` als *RDF-Klassen* definiert werden. Weitere neue Klassen sind u. a. *Literal*, eine Klasse, die Werte wie die Menge aller Zeichenketten umfasst, *Container* als »Oberklasse« der genannten Container und die Klassen *class* selbst. (Hayes (2002) [55] begründet, wieso diese rekursive Definition unproblematisch ist. Darauf soll hier aber nicht eingegangen werden.) Durch die Klassen werden sozusagen die grundlegenden Datentypen definiert. Diese Datentypen werden einem Objekt durch die Eigenschaft `type` zugewiesen.

Weiter werden die Eigenschaften (*properties*) nicht als Teile der Objekte oder Ressourcen, sondern als eigenständige Einheiten definiert, die einen Definitionsbereich (*domain*) und einen Wertebereich (*range*) haben. Sie sind als »Verbindungen« oder »Kanten« im RDF-Graphen eingezeichnet. Dabei ist der Ausgangsknoten der Kante ein Element des Definitionsbereichs und der Endpunkt ein Element des Wertebereichs. Die oben definierte Eigenschaft `type` hat z. B. als Definitionsbereich die Klasse der Objekte (Ressourcen) und als Wertebereich die Klasse der Klassen – sie ordnet einem Objekt also seine Klasse zu.

Da die Eigenschaften (*properties*) selbst aber auch eine Klasse bilden, können sie auch als Knoten im Netz auftreten. In Abbildung 15.4 wird z. B. im oberen Teil die Eigenschaft `eg:author` als Knoten beschrieben und im unteren Teil als Kante verwendet. (Formal handelt es sich bei dem RDF-Graphen um einen bezeichneten Graphen, bei dem jeder Kante eine Bezeichnung zugeordnet werden kann. Als eine solche Bezeichnung kann dann auch ein Knoten gewählt werden. Dadurch wird der Widerspruch aufgehoben, dass ein Objekt als Knoten und an anderer Stelle im selben Graph als Kante auftritt.) Dadurch, dass Eigenschaften als Kanten und Knoten auftreten können und die Menge der Knoten selbst als Knoten auftritt, können RDF-Graphen zunächst etwas verwirrend aussehen. Wie weit sie und die Inferenzmöglichkeiten letztendlich nützlich sind, wird sich zeigen müssen.

Die grundlegenden Eigenschaften (*properties*), die zur Strukturierung von Vokabularien definiert werden, sind *SubClassOf* und *SubPropertyOf*, die Teilmengen und Spezialisierung beschreiben. Beide werden im Entwurf der Sprachdefinition als transitiv bezeichnet. Das heißt, ein Objekt, das durch mehrere `SubClassOf`-Eigenschaften von einem anderen Objekt aus »erreicht« werden kann, ist selbst eine Unterklasse dieses Objekts. Ist eine Eigenschaft eine Spezialisierung (`SubPropertyOf`) einer anderen Eigenschaft, so sind ihre Definitions- und Wertebereiche Unterklassen (`SubClassOf`) der Definitions- und Wertebereiche dieser Eigenschaft. Mit dieser Definition können hierarchische Klassifikationen beschrieben werden, in denen Vererbungsmechanismen gelten.

15.4.2 Pläne für ein Semantisches Web

Die Entwicklung von RDF-Schema ist Teil der *Semantic Web Initiative* des W3C, die sich zum Ziel gesetzt hat, Beschreibungsstandards und Technologien zu entwickeln, mit denen im Web nicht nur die Suche nach Informationen und Dokumenten verbessert werden kann, sondern auch die automatische Verarbeitung von Daten und Wissen aus unterschiedlichen Quellen unterstützt wird. Dadurch sollen automatisierte Dienste in den unterschiedlichsten Bereichen wie digitale Bibliotheken, E-Business und Gesundheitsdienstleistungen angeboten werden können. Dazu soll auf den vorhandenen bzw. zurzeit entwickelten XML- und RDF-Anwendungen aufgebaut werden. Zusätzlich soll mit Verschlüsselung und digitaler Signatur bzw. sicherer Identifizierung die Verlässlichkeit der Kommunikation im Web erhöht werden.

Für die Entwicklung dieses »Semantischen Web« (*Semantic Web*) geht die Initiative von einem Schichtenmodell aus, dessen Grundlage der URI (*Universal Resource Identifier*) zur Benennung von (Web-)Objekten und die Verwendung von Unicode als sprachen- und schriftenübergreifende Kodierung bilden. Darauf setzten als nächste Schicht XML, XML Schema und das Namensraumkonzept von XML auf, mit denen Dokumente und Datensätze formatiert werden können. Über dieser Schicht liegt mit RDF und RDF-Schema die Schicht, in der Aussagen über (Web-)Objekte formuliert und Typinformationen über Objekte und ihre Beziehungen untereinander beschrieben werden können. Bis zu dieser Schicht wird im Wesentlichen ein Datenkonzept aufgebaut, mit dem die Beschreibung und Bearbeitung von Inhalten ermöglicht werden soll. Der erste Schritt dazu soll in einer Vokabular- und Ontologieschicht getan werden, in der Inhaltskonzepte und Beziehungen zwischen diesen Konzepten entwickelt werden sollen.

Für das Semantische Web sind noch drei weitere Schichten geplant: In einer *Logikschicht* (*logic layer*) sollen Regeln zur Verarbeitung der in der Ontologieschicht beschriebenen Inhalte definiert werden, die dann in der darüber liegenden *Berechnungsschicht* (*proof layer*) ausgeführt werden. Als letzte Schicht ist schließlich eine *Bewertungsschicht* (*trust layer*) vorgesehen, in der entschie-

den werden soll, ob die Ergebnisse vertauenswürdig genug sind, um angewendet zu werden.

Für die Entwicklung formuliert die W3C-Initiative eine Reihe von Prinzipien (Koivunen und Miller, 2001 [65]), die natürlich auf dem Schichtenmodell und anderen Aktivitäten des W3C aufbauen und vieles von dem widerspiegeln, was bereits an anderer Stelle eingeführt oder erwähnt wurde:

- *Alles kann durch einen URI (Universal Resource Identifier) identifiziert werden:* Für physische Objekte wie Personen oder Orte kann das auch durch Beschreibungen im Web oder die E-Mail-Adresse geschehen.
- *Objekte und Verweise (Links) können einen Typ haben.*
- *Unvollständige und Teilinformationen müssen toleriert werden:* Das heißt, dass Anwendungen im »Semantic Web« auch funktionieren müssen, wenn nur unvollständige Informationen vorliegen, Links nicht funktionieren oder Seiten sich geändert haben.
- *Es muss auch ohne die »absolute Wahrheit« gehen:* Das Web ist auch jetzt ein offenes System, in dem im Prinzip jede und jeder Dokumente veröffentlichen kann. Einige Server sind dabei vertrauenswürdiger als andere, es gibt verschlüsselte Übertragung und vereinzelt auch digitale Signaturen. Auch heute liegt die Entscheidung, welcher Quelle oder welchem Server man vertraut, bei den Nutzenden oder den jeweiligen Anwendungen. Auch das Semantische Web wird selbst keine Garantien auf Vertrauenswürdigkeit der Quellen und Richtigkeit der Informationen geben können. Das bleibt nach wie vor die Angelegenheit der Anwendungen und Nutzenden.
- *Weiterentwicklungen sollen unterstützt werden:* Das Semantische Web nutzt beschreibende Konventionen und Regeln, die weiterentwickelt und an die Bedürfnisse einzelner Gruppen angepasst werden können. So können existierende Daten später durch zusätzliche Attribute in den Kontext gestellt werden, in dem sie gültig waren, oder es können Erweiterungen und Ergänzungen vorgenommen werden, ohne die Daten selbst zu ändern.
- *So wenig Festlegungen wie möglich:* Dadurch, dass unnötige Spezifikationen und Festlegungen vermieden werden, werden einfache Anwendungen nicht behindert, und auch für komplexe Anwendungen bleibt der notwendige Raum für Entwicklungen.

Betrachtet man die Entwicklung der Forschung zur künstlichen Intelligenz, die Entwicklung des Web (siehe Abschnitt 16.1) und die (kurze) Geschichte von XML, scheinen die Pläne zum Semantischen Web recht ambitioniert. In allen Fällen hat sich gezeigt, dass die Entwicklungen, solange sie sich im technischen, d. h. syntaktischen Bereich bewegt haben, durchaus erfolgreich waren. Expertensysteme, die mit streng formatierten und wohl definierten Daten arbeiten, wie die anfangs beschriebenen Fahrplan-Auskunftssysteme, sind erfolgreich; der Erfolg von HTML und HTTP zum Layout und Verknüpfen von

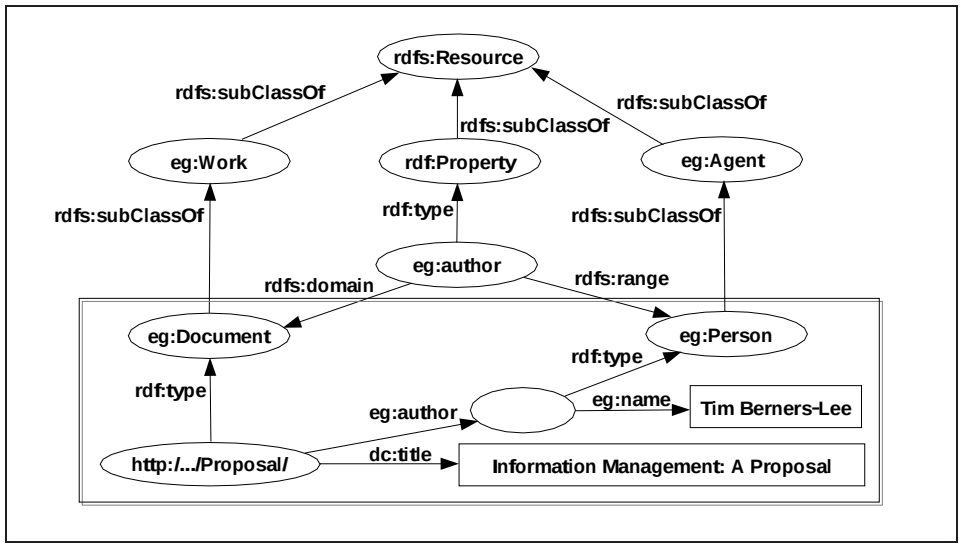

Abbildung 15.4 – Ein RDF-Graph, mit dem die Eigenschaft `eg:author` definiert wird: Im oberen Teil des Diagramms sind die allgemeinen Objekte dargestellt, von denen die spezifischen Objekte und Eigenschaften des Beispiels abgeleitet werden. Letztere sind durch das Präfix `eg:` gekennzeichnet, während `rdf:` für den Namensraum von RDF-Syntax steht, `rdfs:` für den von RDF-Schema und `dc:` für den von Dublin Core. Der Knoten `eg:author` ist eine Eigenschaft (*property*) mit dem Definitionsbereich `eg:Document` und dem Wertebereich `eg:Person`, weist also einem Dokument eine Person als Autor oder Autorin zu. Im unteren Bereich findet sich `eg:author` noch einmal als Label an der Kante zwischen dem Knoten http: ... und dem unbenannten Knoten vom Typ `eg:Person`. Hier weist er dem Objekt vom Typ `eg:Document`, das die Dublin-Core-Eigenschaft `dc:title` mit dem Wert »Information Management: A Proposal« hat, ein Objekt vom Typ `eg:Person` mit dem Namen Tim Berners-Lee als `eg:author` zu. Die Abbildung ist aus Bricklay und Guha (2002) [15] entnommen.

Web-Seiten ist unbestritten, und XML bewährt sich als Austauschformat zwischen Datenbanken und anderen Anwendungssystemen nicht nur im Web.

Sobald es aber darum geht, menschliche Informationsverarbeitung mit automatischen Systemen zu simulieren oder auch nur darum, Informationen zwischen Maschinen und Menschen auszutauschen, werden die Erfolge erheblich seltener. Viele der hochtrabenden Pläne, die zu diesen Fragen in den letzten 40 Jahren gemacht wurden, mussten immer wieder verschoben und relativiert werden. Wo sie erfolgreich waren, lag das oft eher an der Anpassung der Menschen an die Maschinen als umgekehrt an der Anpassung der Maschinen an die Menschen. Die Ausgangslage für das Semantische Web als weltweites, dezentrales und dynamisches Medium, in dem auch noch viele kommerzielle und politische Interessen die Entwicklung beeinflussen, ist sicherlich nicht sonderlich günstig, um diesmal erfolgreicher zu sein.