SEMANTIC WEB – EINE KURZE EINFÜHRUNG

Klaus Meyer-Wegener

Auf Wunsch der Mitglieder des „Interdisziplinären Zentrums für Editionswissenschaft“[1] an der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) habe ich am 22. Oktober 2018 in der Mitgliederversammlung einen Vortrag über das „Semantic Web“ gehalten. Dieser Beitrag zum Magazin ist nun die schriftliche Ausarbeitung dieses Vortrags.

Ausgangspunkt für das Semantic Web[2] war das ubiquitäre World-wide Web (WWW), oft auch einfach als „Internet“ bezeichnet, obwohl das eigentlich nur die darunterliegende Rechnernetz-Infrastruktur ist. Nachdem man dort mit dem Klicken auf Links als navigierende Suche nach Information begonnen hatte, zeichnete sich sehr bald ein Bedarf nach deklarativer Suche ab, der dann von den verschiedenen Suchmaschinen befriedigt wurde. Das ist allerdings immer eine Volltextsuche, d.h. sie findet Texte, wenn ein vorgegebenes Wort oder mehrere vorgegebene Wörter in ihnen vorkommen. Sie findet sie aber nicht, wenn diese Wörter nicht vorkommen, auch wenn der Text inhaltlich viel mit dem Begriff zu tun hat. So ist in älteren Texten noch das Wort „Negerkuss“ zu finden, aktuell wird dafür aber „Schaumkuss“ verwendet – auch bei der Suche. Deshalb stellte sich die Frage, ob man statt mit Wörtern nicht auch nach „Bedeutung“ suchen könnte. Das kann man, aber dazu muss sie den WWW-Texten hinzugefügt werden. Ob das dann von Hand geschieht oder durch einen Text-Analyse-Algorithmus, ist dann noch eine andere Frage. Zuerst muss man klären, wie diese „Bedeutung“ (Semantik) überhaupt aussehen kann.

Und es gab noch einen weiteren Anlass, das WWW weiterzuentwickeln: Es enthält zunächst einmal nur Texte, inzwischen auch Bilder, Tonaufnahmen und Videos, aber das sind allesamt unstrukturierte oder bestenfalls schwach strukturierte Daten. Daneben gibt es aber auch noch in großem Umfang strukturierte Daten, die man sich als Formulare mit Feldern und Inhalten, als Tabellen, als Karteikartensammlungen und dergleichen vorstellen kann. Alle Datenbanken enthalten solche strukturierten Daten – und können sie sehr flexibel auswerten, was mit Texten leider nicht geht. Auch Tabellenkalkulation (in Microsoft Excel und ähnlichen Programmen) verwendet solche strukturierten Daten. Man möchte sie in vielen Fällen in den eigenen Datenbestand integrieren und mit ihm zusammen auswerten, man denke nur an Wirtschaftsentwicklung oder Raumpla-

1 https://www.ized.fau.de/ (letzter Zugriff am 14.08.2019).

2 Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph und York Sure: *Semantic Web – Grundlagen*. Springer : Berlin, Heidelberg, 2008. – ISBN 978-3-540-33993-9.

nung. Während bestimmte Daten dem Datenschutz unterliegen (Personalverwaltung, Gesundheitswesen, Banken, Versicherungen), sind andere im Prinzip öffentlich (Wetter, Landkarten, amtliche Statistiken). Bevor man diese nutzen kann, muss man sie aber verstehen. Das hat zwei Aspekte: die Struktur der Daten und ihre Bedeutung. Als Struktur (oder Format) kann ein sehr breites Spektrum vorkommen: CSV, XML, JSON und viele andere mehr. Es ist hier nicht von Interesse, was diese Abkürzungen bedeuten. Wichtig ist, dass sie alle ganz verschieden sind und dass man sich auf jedes dieser Formate erst einstellen muss. Noch schwieriger ist es mit der Bedeutung: Diese strukturierten Daten weisen Bestandteile auf, die wir hier als „Felder" bezeichnen wollen, wie die Felder, die man in Formularen ausfüllen soll. Sie können auch geschachtelt sein; das Feld „Adresse" kann aus den untergeordneten Feldern „Straße", „Hausnummer", „Wohnort" usw. bestehen. Hier habe ich schon vorausgesetzt, dass die Felder eine Bezeichnung haben, was leider auch nicht immer der Fall ist. Und selbst wenn – was ist eigentlich ein „Preis"? Ist die Mehrwertsteuer schon enthalten oder noch nicht? Ist es der Preis pro Stück oder der Gesamtpreis? Solche Frage treten immer auf, wenn man es mit einem fremden Datenbestand zu tun hat, und es ist alles andere als einfach, sie zufriedenstellend zu beantworten. Andererseits muss man das aber tun, wenn man zu vernünftigen Auswertungen dieser Daten kommen möchte.

Die Idee von Tim Berners-Lee, dem „Vater" des WWW, war nun gegen Ende der neunziger Jahre, die inzwischen ubiquitäre Infrastruktur des WWW auch für die Verbreitung von strukturierten Daten zu nutzen[3]. Man soll sie herunterladen, abfragen und auswerten können wie in einer Datenbank. Im Zusammenhang mit „Big Data" und „Data Science" gewinnt das gerade jetzt erst so richtig an Bedeutung. Es bedeutet aber auch, dass die Daten dann sozusagen von überall her kommen können. Deshalb muss ihre Bedeutung erklärt werden – vom Vertrauen in die Korrektheit der Daten einmal ganz abgesehen. Da sind zunächst die Anbieter der Daten in der Pflicht, die etwas dazu sagen müssen. Vermutlich haben sie ja auch ein Interesse daran, dass ihre Daten verwendet werden[4]. Zur Beschreibung von Daten verwendet man heute sog. Ontologien – eine der vielen unreflektierten Übernahmen attraktiv klingender Begriffe in die Informatik, bei der die ursprüngliche Bedeutung in der Philosophie großzügig angepasst wurde. Hier sind es eigentlich nur Begriffssystematiken oder -systeme (Taxonomien), die eine Festlegung von Begriffen ermöglichen, um sie dann bei der Kennzeichnung von Daten zu verwenden. So wird eben auch festgelegt, was mit „Preis" bei diesen Daten gemeint sein soll. Wichtig

3 Tim Berners-Lee, James Hendler und Ora Lassila: The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. In: *Scientific American*, 284 (5), Mai 2001, S. 34–43 (dt.: Mein Computer versteht mich. In: *Spektrum der Wissenschaft*, August 2001, S. 42–49).

4 Das sollte bei guter wissenschaftlicher Praxis immer mit einer ordentlichen Quellenangabe erfolgen, die dann auch dem Anbieter zugutekommt.

ist dabei, dass auch Beziehungen zwischen Begriffen definiert werden, die sehr zu ihrer konsistenten Verwendung beitragen können. So werden Oberbegriffe und Unterbegriffe einander zugeordnet und es werden Synonyme und Homonyme benannt, u.U. sogar Gegensätze. Ganz zentral ist auch die Zuordnung von Typ und „Instanz", wobei letzteres leider eine direkte Übernahme aus dem Englischen ist und auf Deutsch eigentlich „Exemplar" heißen müsste. Dazu gehören beispielsweise „Säugetier" und „Löwe", „Flugzeug" und „Airbus A380" sowie „Fußballspieler" und „Ronaldo". Bevor ich näher auf die Nutzung solcher Ontologien für die Kennzeichnung der Bedeutung von Daten eingehe, muss ich aber noch die Struktur der Daten etwas genauer beschreiben.

Resource Description Framework (RDF)

Dafür hat sich das „Resource Description Framework" (RDF) etabliert. Es ist sehr einfach und gerade deshalb sehr allgemein. Alles wird dargestellt in Form von Tripeln aus Subjekt, Prädikat und Objekt. Das „Prädikat" ist dabei meist ein Verb. Damit kann man elementare Aussagen formulieren wie: Ein Professor hält einen Vortrag. Nun ist das natürlich noch sehr simpel, seine Wirkung entfaltet es erst dadurch, dass das Subjekt des einen Tripels das Objekt eines anderen Tripels sein kann und umgekehrt. Damit kann man dann schon recht komplexe Zusammenhänge ausdrücken. Wenn man sich wieder ein Formular vorstellt, so kann man das nun z.B. darstellen in der Form:

```
Müller – ist – der Name,
Erlangen – ist – der Geburtsort,
17. Mai 1965 – ist – der Geburtstag
```

Damit kann man das ganze Formular mit allen seinen Daten beschreiben. Für Tabellen geht das analog. Und man kann das fortsetzen mit:

```
Erlangen – ist – eine Stadt
```

So ergibt sich ein immer größeres Netz von Aussagen, das man als Graph darstellen kann: Subjekte und Objekte sind die Knoten, Prädikate die Kanten zwischen ihnen. In Anlehnung an das WWW hat Tim Berners-Lee das „Giant Global Graph" (GGG) genannt, aber das hat sich nicht durchgesetzt.

Der Oberbegriff für Subjekt, Prädikat und Objekt ist Ressource, daher die Bezeichnung RDF. Nun ist die Frage, was alles als eine solche Ressource verwendet werden darf. Wenn man einfach jedes beliebige Wort zulässt, ist wenig gewonnen, denn dann werden die unterschiedlichen Verwendungen dieses Worts zu unterschiedlichen und wahrscheinlich sogar widersprüchlichen Aussagen führen. Jede Ressource muss also eindeu-

tig benannt werden. Dabei schadet es nicht, wenn eine Ressource mehrere Namen hat (Synonyme), denn das kann in RDF problemlos konstatiert werden:

```
Name1 – meint dasselbe wie – Name2
```

Homonyme darf es aber nicht geben, da muss ggf. umbenannt oder präziser benannt werden.

Erfreulicherweise stellt das WWW auch dafür schon einen Mechanismus bereit, denn auch Web-Seiten müssen einen weltweit eindeutigen Namen haben. Das sind die sog. „Uniform Resource Identifiers" (URIs), deren bekannteste Form der „Uniform Resource Locator" (URL) ist. Die sind hierarchisch aufgebaut, damit man sie auch dezentral vergeben kann. Eine URI kann auf eine Web-Seite führen, die das benannte Konzept genauer erläutert, muss es aber nicht.

Somit sind nun die drei Ressourcen in einem RDF-Tripel grundsätzlich URIs. Die einzige Ausnahme macht das Objekt, das auch ein einfaches Literal sein kann, also eine Zahl, ein Wort oder auch ein Satz:

```
Preis-URI – ist-URI – 15 Euro
```

Ein Beispiel aus dem Buch von Hitzler et al.[2] zeigt den Einsatz von richtigen URIs:

```
[Subjekt: „http://example.org/SemanticWeb",
Prädikat: „http://example.org/VerlegtBei",
Objekt: „http://www.springer.com/Verlag"]
```

Und nun auch noch mit einem Literal (einem Wert) als Objekt:

```
[Subjekt „http://www.springer.com/Verlag",
Prädikat: „http://example.org/Name",
Objekt: „Springer-Verlag"]
```

Wie man deutlich sieht, wird das Hinschreiben dadurch zwar präziser, aber auch etwas mühsam. Zur Vereinfachung hat man die sog. *Turtle-Syntax* entwickelt. Sie setzt die URIs in spitze Klammern und die Literale in Anführungszeichen. Am Ende steht ein Punkt.

```
<http://www.springer.com/Verlag> <http://example.org/Name> „Springer-
Verlag" .
```

Der Punkt ist keine Spielerei, sondern erlaubt die Abkürzung bei wiederholt vorkommenden Teilen. Oft will man zu ein und demselben Subjekt mehrere Aussagen machen. Dann verwendet man statt des Punkts ein Semikolon, dem dann nur noch Prädikat und Objekt folgen. Und man darf immer wieder vorkommende Teile von URIs als Präfix deklarieren und mit einem Kürzel versehen:

```
@prefix ex: <http://example.org/>
@prefix springer: <http://springer.com/>
ex:SemanticWeb ex:VerlegtBei springer:Verlag ;
        ex:Titel „SemanticWeb – Grundlagen“ .
springer:Verlag ex:Name „Springer-Verlag“ .
```

So etwas kann man mit ein wenig Übung auch als Mensch verstehen. Zugleich kann es problemlos von Computern verarbeitet werden.

Für die Speicherung großer Mengen von Tripeln bietet sich eine Tabelle mit vielen Zeilen an:

Subjekt	Prädikat	Objekt
ex:SemanticWeb	ex:VerlegtBei	springer:Verlag
ex:SemanticWeb	ex:Titel	„SemanticWeb – Grundlagen“
springer:Verlag	ex:Name	„Springer-Verlag“

Solche Strukturen kann man problemlos in Datenbanken abspeichern – und dann sehr flexibel auswerten! SPARQL[5] ist eine Sprache, in der man diese Auswertungen formulieren kann. Es würde zu weit führen, hier auch nur die wichtigsten Bestandteile von SPARQL zu beschreiben. Ich versuche es mit einem Beispiel für eine Anfrage:

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE { ?x abc:cityname ?capital ;
    abc:isCapitalOf ?y .
 ?y abc:countryname ?country ;
    abc:isInContinent abc:Africa . }
```

Die Präfix-Definition wurde oben schon eingeführt; sie hat nur eine leicht andere Syntax. In der SELECT-Zeile schreibt man die Ausgaben, die man erhalten möchte, hier also Paare von einer Hauptstadt und einem Land. Die hinter WHERE stehenden Zeilen definieren Muster für Tripel. Nur Tripel, die zu einem der Muster passen, kommen

5 SPARQL Protocol And RDF Query Language.

überhaupt in Frage. Ressourcen, die in einem Tripel-Muster mit einem Fragezeichen beginnen, sind nicht vorgegeben. Was immer ein Tripel an dieser Stelle enthält, passt. Ressourcen ohne Fragezeichen dagegen müssen genau so im Tripel enthalten sein. Hinter dem Fragezeichen steht dann noch eine Bezeichnung. Kommt die zusammen mit dem Fragezeichen noch in einem anderen Muster vor, müssen zwei Tripel an dieser Stelle in beiden Fällen dieselbe Ressource benennen. Wenn man es in normale Sprache übersetzt, sucht die oben genannte Anfrage nach einer Stadt („?x"), die die Hauptstadt von „?y" ist, und „?y" muss in Afrika liegen. Die Namen dieser Stadt und ihres Landes sollen ausgegeben werden.

RDF Schema

In der Nutzung von RDF trat sehr bald ein weiterer Bedarf zutage, nämlich die Festlegung einer gemeinsamen Struktur für eine größere Menge von Ressourcen. Dazu kann ich noch einmal das Beispiel des Formulars bemühen: Darin sind Felder vorgesehen, die man gern in jedem Fall mit Inhalten gefüllt haben würde („Pflichtfelder"). So wie RDF bisher beschrieben wurde, kann man aber nicht sicherstellen, dass bestimmte Tripel in Abhängigkeit von anderen immer vorhanden sein müssen („Jeder Angestellte muss eine Personalnummer haben!")[6]. Außerdem hat das Formular bei allen Angestellten die gleiche Struktur, es wiederholt sich. Auch das kann man nicht sicherstellen. Man muss im Moment noch bei jedem Angestellten die ganze Struktur wieder anlegen und darf dabei nichts vergessen[7]. Dass das dann für die Gesamtmenge aller Angestellten auch wirklich zusammenpasst, muss man sehr mühsam selbst kontrollieren.

Mit einigen zusätzlichen definitorischen Tripeln kann das aber auch das verarbeitende System übernehmen. Genau dafür wurde *RDF Schema* entwickelt. Es umfasst Ressourcen mit einer auf diese Situation zugeschnittenen Bedeutung. Auch hier ziehe ich wieder ein Beispiel heran, um das zu erläutern[8]:

```
<ex:SemanticWeb> <rdf:type> <ex:Lehrbuch> .
<ex:Lehrbuch> <rdf:type> <rdfs:Class> .
<ex:Buch> <rdf:type> <rdfs:Class> .
<ex:Lehrbuch> <rdfs:subClassOf> <ex:Buch> .
```

6 In Datenbanken geht das: Man deklariert ein Feld als NOT NULL, also nicht leer zu lassen.

7 Natürlich kann es auch Felder geben, die in einem speziellen Fall einfach nicht ausfüllen kann, z.B. „Geburtsname". Dann darf man das zugehörige Tripel durchaus weglassen. Nur bei Pflichtfeldern darf man das nicht.

8 Wenn in einem Bezeichner einer Ressource ein Doppelpunkt auftaucht, dann ist das entweder – wie oben erläutert – ein Präfix oder ein sog. *Namensraum*. Was man damit alles machen kann, ist in der einschlägigen Literatur nachzulesen, z.B. in Hitzler et al., a.a.O.

Das Prädikat „rdf:type“ sagt aus, dass das Subjekt vom dem Typ ist, der als Objekt genannt wird. Damit kann man ausdrücken, dass Herr Müller ein Angestellter ist[9]. Damit ist aber noch nicht viel gewonnen. Erst das zweite Tripel sorgt für die gewünschte Einheitlichkeit: Damit wird ausgesagt, dass es sich bei „Lehrbuch“ um eine Klasse handelt, und das bedeutet, dass dazu die Prädikate definiert sind, die ein „Lehrbuch“ haben kann bzw. muss, also z.B. „ist der Autor von“, „ist der Titel von“, „ist der Verlag von“ usw. Im dritten Tripel wird „Buch“ als eine zweite Klasse eingeführt, weil man offenbar nicht nur etwas über Lehrbücher aussagen möchte, sondern auch über Bücher im Allgemeinen. Diese Klasse verfügt ebenfalls über eine Menge von Prädikaten. Und weil „Buch“ und „Lehrbuch“ etwas miteinander zu tun haben – Lehrbücher sind ein Spezialfall von Büchern –, kann man auch das im vierten Tripel noch definieren. Was genau „rdfs:subClassOf“ bedeutet und welche Konsequenzen es hat, kann hier aus Platzgründen nicht weiter erläutert werden[10].

Weitere Elemente von RDF Schema sind die Klassen:

```
Class, Resource, Property, Literal
```

Sie können als Objekt von „rdf:type“ verwendet werden und ein Subjekt charakterisieren. Analog die Eigenschaften:

```
subClassOf, subPropertyOf, domain, range
```

Sie werden als Prädikate verwendet und verknüpfen Klassen oder Eigenschaften.

Mit diesen Mitteln kann man schon einfache Ontologien, also Begriffssysteme definieren. Man beschreibt damit, was unter einem „Lehrbuch“ oder einem „Angestellten“ zu verstehen ist. Und man kann mögliche richtige Aussagen von ganz sicher falschen trennen:

```
<ex:HansMeier> <ex:hatAlter> „-127“ .
```

Das wiederspricht einer hoffentlich an anderer Stelle mit der Eigenschaft „rdfs:range“ gemachten Aussage, dass das Alter einer Person im Bereich von 0 bis 150 liegen muss. Auch werden erste Schlussfolgerungen möglich. Aber es ist noch nicht umfassend genug, und deshalb sollen diese Dinge erst im folgenden Abschnitt ausgeführt werden.

9 Er ist dann „vom Typ“ Angestellter oder „hat den Typ“ Angestellter.

10 Wer sich mit objektorientierter Programmierung auskennt, weiß, dass es sich um eine Vererbungsbeziehung handelt.

OWL

Mit Hilfe der „Web Ontology Language“ (OWL)[11] ist dann die umfassende Definition von Taxonomien, also Begriffshierarchien möglich. Sie hat eine streng formale Semantik und erlaubt nun wirklich umfassende logische Schlussfolgerungen, speziell zur Entdekkung von Widersprüchen in einer großen Menge von Tripeln. Da die Syntax etwas umständlich ist, wird sie hier nicht gezeigt. Stattdessen werden umgangssprachliche Beispiele mit gleicher Aussage verwendet, die lesbarer sind.

Zunächst einmal gibt Klassen und Instanzen, wie es schon bei RDF Schema zu sehen war:

```
Julius „ist ein“ Mensch.
```

Und es können auch Subklassen definiert werden:

```
Alle Menschen „sind auch“ Lebewesen.
```

Den Klassen werden Eigenschaften (Properties) zugeordnet, die bei allen (!) ihren Instanzen vorhanden sein müssen:

```
Ein Mensch „hat“ einen Namen.
```

Ein Beispiel für die Verwendung von OWL ist CIDOC CRM[12], das einen Bezug zur Editionswissenschaft hat. Es definiert die Begriffe, die man benötigt, um (Museums-) Objekte eindeutig zu beschreiben. Das geschieht Ereignis-orientiert, d.h. die Beschreibung orientiert sich an Ereignissen wie der Herstellung eines Objekts (durch einen Künstler, einen Handwerker usw.), dem Fund, dem Kauf, der Aufstellung in einer Sammlung usw. Ein Beispiel:

```
<E12 Production> <P108 has-produced>
       <E24 Physical Man-Made Thing> .
```

11 Dean Allemang und Jim Hendler: *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. 2. Aufl. Morgan Kaufmann : Amsterdam a.o., 2011. – ISBN 978-0-12-385965-5. Zur Geschichte des Buchstabendrehers in der Abkürzung siehe auch http://lists.w3.org/Archives/Public/www-webont-wg/2001Dec/0169.html .

12 Das «Conceptual Reference Model» des Comité International pour la Documentation, heute: ICOM's International Committee for Documentation; siehe http://www.cidoc-crm.org/ . Das CRM definiert Klassen und Eigenschaften noch verbal, mit dem Erlangen CRM (http://erlangen-crm.org/) ist es in OWL ausformuliert worden.

Zum Ereignis „E12 Production“ ist noch eine ganze Reihe weiterer Eigenschaften definiert, ebenso wie für das Objekt „E24 Physical Man-Made Thing“. Nun ist das noch sehr abstrakt, weil es für alle möglichen (Museums-) Objekte verwendbar sein soll. Seine ganze Wirkung entfaltet es erst, wenn man die Zuordnung eigener Datenobjekte zu den Bestandteilen dieser Ontologie vornimmt:

```
“Mona Lisa” <rdf:type> <E24 Physical Man-Made Thing> .
„Mittelalter“ <rdf:type> <E2 Temporal Entity> .
```

Das Ziel ist, dass nach und nach alle Museen zu ihren Sammlungen eine Beschreibung vornehmen, die diesem vorgegebenen Muster folgt. Bisher denken sie sich eher jeweils eigene Beschreibungen aus, was zu einem geradezu babylonischen Sprach-Wirrwarr geführt hat. Mit CIDOC CRM und OWL kann man hier zu deutlich mehr Einheitlichkeit, Vergleichbarkeit, Austauschbarkeit und Zusammenführung gelangen.

Linked (Open) Data

Mit den bis hierher eingeführten Methoden lässt sich die einheitliche Beschreibung der Struktur und sogar der Bedeutung von Daten – insbesondere auch von Web-Seiten – erreichen. Und danach kann man suchen, indem man SPARQL benutzt, siehe oben. Was nun noch fehlt ist die Nutzung der Infrastruktur des WWW zur Bereitstellung von (freien) strukturierten Daten[13]. RDF kann auch das leisten. Eine Tabelle mit Einwohnerzahlen[14]:

Stadt	*Einwohnerzahl*
München	1.456.039
Hamburg	1.834.823
Berlin	3.613.495
Köln	1.080.394

wird in RDF zum Beispiel dargestellt mit:

```
<München> <hat-Einwohner> 1456039 .
<Hamburg> <hat-Einwohner> 1834823 .
<Berlin>  <hat-Einwohner> 3613495 .
<Köln>    <hat-Einwohner> 1080394 .
```

13 Christian Bizer, Tom Heath und Tim Berners-Lee: Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.* 5(3), 2009, S. 1–22.

14 Alle Zahlen stammen aus Wikipedia (Abruf am 18. März 2019).

Das man da nun noch viel ergänzen kann und sollte, ist nach dem oben Gesagten klar: So sollte ausgedrückt werden, dass alle vier Subjekte eine „Stadt" sind, also Instanzen der Klasse „Stadt". Weiterhin sollte gesagt werden, dass alle „Städte" eine Einwohnerzahl haben und das die über die Eigenschaft „hat-Einwohner" zugeordnet ist. Das ist mit den oben nur skizzierten Möglichkeiten von RDF Schema und OWL problemlos machbar.

Die Bereitstellung solcher Daten ist Teil des Semantic Web; nur sind es sind dann eben keine Web-Seiten – obwohl man sich die ganzen Tripel durchaus auch ansehen könnte, aber da erkennt man nur wenig. Ziel ist es ja, diese Daten insgesamt oder in (durch SPARQL definierten) Ausschnitten herunterzuladen und in eigene Auswertungen einzubeziehen, sei es nun in einer Datenbank, in Excel, in R oder in Python – und was es an Auswertungswerkzeugen sonst noch so gibt.

Anbieter solcher Daten sind typischerweise Regierungen, Zeitungen, Rundfunk- und Fernsehanstalten, Museen und manchmal auch Wirtschaftsunternehmen. Es gibt inzwischen schon eine sehr große Menge solcher Daten-Angebote; die Abb. 1 kann dazu nur einen groben Überblick geben. Die verschiedenen Farben kennzeichnen die Gruppen von Anbietern, wie sie eben genannt wurden. Als Einstieg in diese Welt ist http://linkeddata.org zu empfehlen, oder auch https://lod-cloud.net/ . Hier kann man in den verschiedensten Datenbeständen herumstöbern und auch den Zugriff über SPARQL ausprobieren.

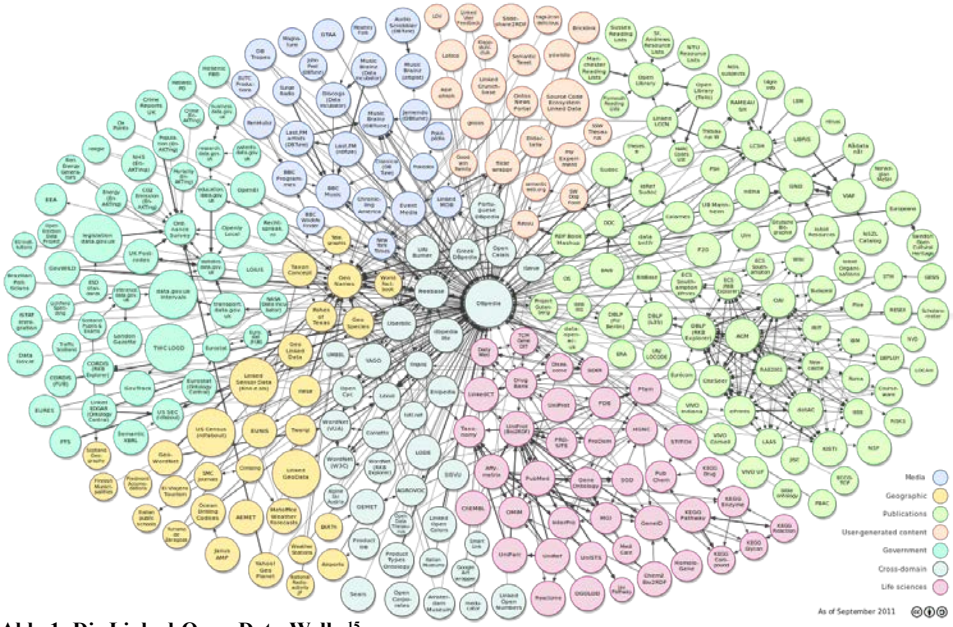

Abb. 1: Die Linked-Open-Data-Wolke[15]

15 Von Richard Cyganiak Anja Jentzsch - http://richard.cyganiak.de/2007/10/lod/, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=24597456.

Zum Schluss

Das Semantic Web ist nun schon etwa zwanzig Jahre alt. Es wird intensiv genutzt, wie gerade die zuletzt genannten Web-Seiten mit langen Listen von Einstiegspunkten demonstrieren. Dennoch ist das Potenzial bei weitem noch nicht ausgeschöpft. Was man noch ein bisschen vermisst ist eine bequeme Art, aus schon vorhandenen Daten die ganzen Tripel zu erzeugen, die man in RDF für ihre Darstellung benötigt. Das will niemand von Hand machen! Bei Datenbanken gibt es solche Werkzeuge schon, aber nur ein kleiner Teil der Daten ist in Datenbanken gespeichert. Viel mehr liegt in Dateien (Excel …) oder in Textform vor. Es gibt erste, noch unzureichende Versuche, Text in RDF zu transformieren[16], aber da ist noch viel zu tun. Dennoch bleibt es für mich eine lohnende Herausforderung, dieses Ziel weiter zu verfolgen. Man stelle sich nur vor, was es bedeutete, wenn man die Bücher aus Bibliotheken in eine dann sicher riesige Menge von Tripeln transformieren könnte, die sich mit Schlussfolgerungen auswerten ließen.

16 Zum Beispiel: Brahim Batouche, Claire Gardent und Anne Monceaux: Parsing Text into RDF Graphs. In: Proc. SEPLN 2015, Actas del XXXI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural, Alicante, 16-18 Sept 2015.