## Semantic Web

In diesem Kapitel werden zentrale theoretische Grundlagen zur Repräsentation und Organisation von Wissen in digitalen Systemen vorgestellt. Anschließend werden die grundlegenden Konzepte des Semantik Web erläutert und zentralen Begriffe aufgegriffen und erläutert. Ein besonderer Fokus liegt dabei auf der graphbasierten Darstellung und Strukturierung von Wissen.

### 2.1 Das Web

Kaum eine andere Technologie veränderte unseren alltäglichen Umgang mit Informationen in so kurze Zeit wie das World Wide Web (WWW). Die rasante Entwicklung des Internets trug entscheidend dazu bei, dass sich die Industriegesellschaft nach und nach in eine Informationsgesellschaft verwandelte. Dank seiner standardisierten Struktur und darauf abgestimmten Infrastruktur bietet das Web zahlreiche Vorteile, insbesondere die hohe Aktualität sowie permanente Verfügbarkeit von Informationen. Diese Eigenschaften führen zu einer vereinfachten schnelleren und kostengünstigeren Informationsbeschaffung. Zudem sind Informationen in zunehmendem Maßen einer breiten Öffentlichkeit zugänglich (vgl. Hitzler et al. 2007: 9–10). Dies entsprach den Vorstellungen von Tim Berners-Lee, der in seiner Zusammenfassung [*WorldWideWeb: Summary*](http://info.cern.ch/hypertext/WWW/Summary.html)das Web als ein leistungsfähiges und zugängliches System beschrieb, in dem zwei zentrale Aspekte miteinander verbundenen seien: Zum einen ist die effiziente Suche nach Informationen, zum anderen die Verknüpfung von Inhalten durch Hypertext. Im Mittelpunkt stand die Überzeugung, dass insbesondere wissenschaftlichen Wissen frei verfügbar sein sollte. Dabei lag der Fokus auf der Erleichterung des Austausches von Informationen in international verstreuten Teams und der Forderung der gemeinsamen Nutzung von Wissen. Nach rund 20 Jahren bewährte sich die Grundprinzipien des Web (vgl. Domingue/Fensel/Hendler 2011: 7).

Doch trotz der zentralen Vorteile und der ursprünglich idealistischen Zielsetzung zeigt sich, dass das web auch problematische Seiten aufweist. Das Web enthält riesige Mengen an Information (vgl. ebd.: 9), die aber hauptsächlich für Menschen gemacht wurde, nicht für Maschinen. Menschen können Texte auf Webseiten verstehen, interpretieren, umformulieren und mit anderen Informationen verknüpfen. Die Maschinen hingegen können das nicht oder sehr eingeschränkt (vgl. Hitzler et al. 2007: 10). Die Suchmaschine durchsuchen Webseiten in der Regel lediglich auf Basis von Schlüsselwörtern, ohne die Bedeutung hinter den Begriffen zu erkennen. Besonders problematisch wird dies bei mehrdeutigen Begriffen, deren Interpretation vom Kontext abhängt. Ein daraus resultierendes Problem besteht darin, dass in vielen Fällen die Antwort auf eine Suchanfrage zwar im Web vorhanden ist, jedoch über mehrere Quelldokumente verteilt. Eine vollständige Antwort erfordert daher die Inhaltliche Verknüpfung und Integration verschiedene Dokumente. (vgl. Domingue/Fensel/Hendler 2011: 9–10). Das Web ist dezentral organisiert, was bedeutet, dass jede Person oder Organisation Inhalte nach eigenen Vorstellungen veröffentlichen kann. Dadurch entsteht eine breite Palette von den Datenformate und Kodierungen, verwendeter natürlichen Sprache oder der Webseitenstrukturen (vgl. Hitzler et al. 2007: 10). Viele Webseiten werden dynamisch aus Datenbanken generiert, doch die zugrunde liegenden Daten bleiben hinter dem HTML verborgen. Dieses sog. *Dark Web* erschwert die Nutzung und Wiederverwendung der Informationen erheblich (vgl. Domingue/Fensel/Hendler 2011: 9–10). Ein weiteres grundlegendes Problem bei der Inforationssuche im Web ist, dass gesuchte Information nicht immer explizit auf Webseite vorhanden ist, ist aber aus mehreren bekannten Fakten logisch ableitbar. Ohne das Verständnis der Zusammenhänge und Fähigkeit zum Schlussfolgern kann Computer keine Schlüsse ziehen (vgl. Hitzler et al. 2007: 11). „Hier handelt sich um die Problematik des impliziten Wissens“ (vgl. ebd.: 11).

### 2.2 Wissensmodellierung im Semantik Web

Das menschliche Gedächtnis setzt sich aus zwei zentralen Komponenten zusammen: dem semantischen Teil für allgemeines, kulturell geteiltes Wissen und dem episodischen Teil für persönlich Erlebtes. Diese Kombination ermöglicht es uns, Bedeutung zu erkennen und Erinnerung sinnvoll einzuordnen. Wörter und Phrasen können verschiedene Bedeutungen haben und ohne Kontext bleibt die Bedeutung vieler Ausdrücke unklar. Selbst strukturierte Informationen wie Zeitangaben entfalten ihre Bedeutung nur, wenn man die dahinterliegende Syntax kennt und weiß, wie diese Information zu lesen ist. Syntax gibt die Regeln vor, während Semantik ihre inhaltliche Bedeutung bezeichnet. Pragmatik geht über die Bedeutungszuweisung hinaus und fragt nach dem Zweck und der Absicht sprachlicher Äußerungen. Sie ist entscheidend für eine vollständige Interpretation von Sprache im Handlungszusammenhang. Während Menschen über kognitive Fähigkeiten verfügen, um sprachliche Bedeutung intuitiv zu erfassen, muss diese Fähigkeit bei Maschinen künstlich nachgebildet oder kompensiert werden (vgl. Dengel 2012: 10–11, 13).

Vor dem Hintergrund der rasanten technologischen Entwicklungen der letzten Jahrzehnte durch das WWW wird deutlich, dass wir heute Zugang zu weit mehr Informationen haben, als wir tatsächlich verstehen oder effektiv verarbeiten können. Deshalb ist es notwendig, Maschinen mit Mechanismen auszustatten, die nicht nur syntaktische Strukturen, sondern auch semantische Bedeutungen erkennen können. Dieses Ziel verfolgt die Idee des Semantic Web, das gegen Ende der 1990er Jahre von Tim Berners-Lee eingefügt wurde und in den 2000er Jahren intensiv weiterentwickeln wurde (vgl. Davies/Studer/Warren 2006: 1–2).

Das Semantic Web erweitert das bestehende Web, indem es dessen globale Systeme zur Indexierung und Benennung nutzt. Es ist sowohl mit herkömmlichen Browsern als auch mit semantisch-fähiger Software zugänglich (vgl. Domingue/ Fensel/ Hendler 2011: 5). Dabei soll jeder Begriff eine eindeutige Kennung erhalten, um Identitäten klar aufzulösen (vgl. Domingue/Fensel/ Hendler 2011: 5). So müssen bspw. Unterschiedliche Bedeutungen eines Begriffs jeweils durch eigene Uniform Resource Identifier (URI) eindeutig gekennzeichnet werden ([vgl. Berners-Lee/ Hendler/ Lassila 2001](https://www.lassila.org/publications/2001/SciAm.pdf)). Dadurch können Beziehungen zwischen beliebigen digitalen Objekten herzustellen – von Fotos über Finanztransaktionen bis hin zu abstrakten Konzepten ([vgl. Feigenbaum ebd. 2007](http://thefigtrees.net/lee/sw/sciam/semantic-web-in-action#page2)).

Ein zentrales Element ist dabei die Repräsentation von Zusatzinformationen (Metadaten) in einer Form, die von Maschinen interpretiert werden kann (vgl. Grütter 2008: 75). Semantische Metainformationen dienen dabei als Verknüpfung der Daten, die idealerweise öffentlich zugänglich sind. Dies bildet die Grundlage für ein maschinenlesbares Wissensnetzwerk (vgl. Dörpinghaus 2022). Solche semantischen Netze organisieren Wissen durch strukturierte, graphische Darstellung, meist in Form von Knoten und Kanten: Graphen (vgl. Dengel 2012: 34).

Frühere Systeme waren häufig zentral organisiert und setzten einheitliche Begriffsdefinitionen voraus. Dies schränkte ihre Skalierbarkeit und Flexibilität stark ein. Komplexe oder offene Fragestellungen ließen sich in solchen Systemen oft nur begrenzt abbilden. Das Semantic Web verfolgte hingegen einen offenen Ansatz, der auch Ambiguitäten zulässt, um eine höhere Ausdrucksstärke und Anpassungsfähigkeit zu ermöglichen. Ziel war die Entwicklung einer Sprache, die sowohl Datenstrukturen als auch logische Schlussregeln beschrieb und die Integration bestehender Wissenssysteme ins Web erlaubte ([vgl. Berners-Lee/ Hendler/ Lassila 2001](https://www.lassila.org/publications/2001/SciAm.pdf)).

In unterschiedlichen wissenschaftlichen Disziplinen wird zunehmend anerkannt, dass Objekte viele Bedeutungen und Verbindungen aufweisen (vgl. Werner 2020: 247). Dies macht deutlich, dass Daten ohne ihren Kontext kaum aussagekräftig sind (vgl. Dengel 2012: 5). Genau hier setzen moderne informationstechnische Verfahren an. Sie versuchen, **Informationsintegration** zu ermöglichen, um ein tieferes Verständnis der Daten und ihrer Zusammenhänge zu erreichen (vgl. Davies/ Studer/ Warren 2006: 1).

Durch technologische Entwicklungen ist es möglich, Metadaten systematisch mit externem Wissen zu vernetzen (vgl. Werner 2020: 247). Dabei entsteht ein semantischer Raum, in dem Daten nicht isoliert, sondern relational verstanden und genutzt werden (vgl. Dengel 2012: 5). Damit Computer Informationen erkennen und verarbeiten können, muss Wissen in strukturierter Form repräsentiert werden. Es reicht nicht aus, Wissen als eine Ansammlung isolierter Fakten zu betrachten, vielmehr als ein dynamisches Netz miteinander verbundener Informationen zu verstehen (vgl. Werner 2020: 247). Dies ist eine zentrale Aufgabe semantischer Technologien wie sie bspw. im Rahmen des World Wide Web Consortium (W3C) entwickelt und standardisiert wurden.

Das W3C entwickelte die Grundprinzipien des Semantic Web und stellte standardisierte Sprachen bereit (vgl. Davies/ Studer/ Warren 2006: 2). Mit den Technologien des Semantic Web[[1]](#footnote-1) können Nutzer:innen Datenspeicher im Web aufbauen, Vokabulare entwickeln und Regeln zur Datenverarbeitung festlegen (vgl. Kamzelak 2018: 424). Auf diese Grundlage entstehen innovative Dienste, die den Austausch und die gemeinsame Nutzung von Wissen vereinfachen (vgl. Dengel 2012: 9). Technologien wie Resource Description Framework (RDF), Web Ontology Language (OWL) oder Protocol and RDF Query Language (SPARQL) ermöglichen es, komplexe Sachverhalte durch semantisch angereichte Beziehungen zwischen Daten dazustellen. Aus diesen Daten bildet sich ein Informationsnetz, der durch seine offene Struktur eine vielseitige Verknüpfung erlaubt. Jede Information kann über eine URI eindeutig angesprochen werden, wodurch sie mit anderen externen, offenen und öffentlich zugänglichen Datenquellen im Web verbunden werden kann (vgl. Andraschke/Wagner 2020: 118–119). Die semantisch angereichten Daten werden in sog. RDF-Tripeln modelliert. In der Praxis werden solche strukturierte Daten als Linked Open Data (LOD) bezeichnet (vgl. Andraschke/Wagner 2020: 119).

### Ontologie

Der Begriff Ontologie stammt ursprünglich aus der Philosophie und wurde später in die Künstliche Intelligenz übernommen. In der Philosophie bezieht sich Ontologie auf die Natur des Seines und auf die Frage, welche Entitäten zur Beschreibung der Welt notwendig sind. In der Informatik Kontext bezeichnet eine Ontologie hingegen eine strukturierte Form der Wissensrepräsentation; dabei wird die philosophische Frage nach dem Sein technisch interpretiert (vgl. Kleb 2015: 31–32). Ziel ist es, Wissen über Begriffe und deren Zusammenhänge in formaler Weise für Computer beschreiben, um effektive Kommunikation und automatisierte Handlungen zu ermöglichen (vgl. Dengel 2012: 64).

Mit dem Aufstieg des WWWs wurden Ontologien zunehmend zentral (vgl. Lanquillon/ Schacht 2023: 185), da sie dem Semantic Web zusätzliche Stärke verleihen: Sie ermöglichen es, Beziehungen zu klassifizieren, abzufragen und daraus Schlussfolgerungen zu ziehen ([vgl. Feigenbaum ebd. 2007](http://thefigtrees.net/lee/sw/sciam/semantic-web-in-action#page2)). Ein zentrales Problem besteht darin, dass Datenbanken häufig unterschiedliche Begriffe für dasselbe Konzept verwenden. Ontologien tragen dazu bei, solche semantischen Unterschiede zu überbrücken, bspw. durch Konvertierungsdienste ([vgl. Berners-Lee/ Hendler/ Lassila 2001](https://www.lassila.org/publications/2001/SciAm.pdf)), mit dem Ziel, ein gemeinsames Vokabular zu schaffen (vgl. Rölke/ Weichselbraun 2022: 257). Darüber hinaus tragen sie zur Reduktion der Komplexität unstrukturierter Informationen bei (vgl. Lanquillon/ Schacht 2023: 186).

Eine Ontologie ist ein Modell, mit dem ein bestimmter Bereich der Welt formal und genau beschrieben wird. Zu diesem Zweck werden die zentralen Konzepte und die Beziehungen zwischen diesen Konzepten festgelegt (vgl. Dengel 2012: 65). Ontologien werden in strukturierter, maschinenlesbarer Form vorgelegt (vgl. Rölke/ Weichselbraun 2022: 257).

Sie bestehen aus vier grundlegenden Komponenten: Konzepten (oder Klassen), Relationen (oder Eigenschaften), Instanzen und Axiomen (oder Regeln) sog. 4-Tupel <C, R, I, A> (vgl. Davies/ Studer/ Warren 2006: 4–5). Klassen bilden Begriffskategorie und werden meistens in Form einer Taxonomie hierarchisch organisiert. Relationen beschreiben Zusammenhänge zwischen den Klassen. Eine besonders häufige Beziehung innerhalb einer Taxonomie ist *ist\_ein(e)*; andere Relationen, wie z.B. *ist\_verbunden\_mit,* müssen expliziert definiert werden. Relationen können auch Eigenschaften zum Ausdruck bringen, z.B. *hat\_Farbe*. Regeln legen fest, was in einem bestimmten Bereich immer gelten soll, wie z.B. *Alle Menschen haben ein Geburtstag*. Eine Instanz repräsentiert ein konkretes, existiertes Element des beschriebenen Bereichs (vgl. Dengel 2012: 65).

Auf dieser Grundlage schaffen Ontologien ein einheitliches Verständnis der Informationsstruktur und ermöglichen Wiederverwendung von Wissen über verschiedene Anwendungen hinweg. Durch ihre Modularität lassen sich spezifische Ontologien leicht in größere Wissenssysteme integrieren. Dies fördert Effizienz, Standardisierung und die automatisierte Auswertung von Wissensbasen. Ontologien machen auch implizites Wissen zugänglich, z. B. durch Hierarchien, und strukturieren Informationen so, dass sie maschinenlesbar, anwendungsübergreifend nutzbar und für Menschen nachvollziehbar sind (vgl. Lanquillon/ Schacht 2023: 186–187).

### Wissensgraphen

Graphen dienen als mathematische Modelle zur formalen Beschreibung netzartiger Strukturen. Solche Systeme zeichnen sich typischerweise durch eine Zusammensetzung von zwei strukturell unterschiedlichen Objektklassen. In der Terminologie der Graphentheorie werden diese beiden Elementen als Knoten (Objekte) und Kanten (Verbindungen bzw. Beziehungen) bezeichnet (vgl. Tittmann 2019: 11), wobei die Knoten Entitäten eines bestimmten Bereichs wie bspw. eine Person oder ein Produkt repräsentieren. Die Kanten verknüpfen diese Entitäten miteinander und geben die Art der Verbindung zwischen ihnen an (vgl. Barrasa/ Webber 2023: [Introducing Knowlege Graphs](https://learning.oreilly.com/library/view/building-knowledge-graphs/9781098127091/ch01.html#idm45681067535008)). Ziel ist der Aufbau einer Wissensbasis, die in weiteren Anwendungen genutzt werden kann (vgl. Lanquillon/ Schacht 2023: 190).

In der Graphentheorie (Euler 1736) werden zunächst die rein topologischen Eigenschaften der Netzstruktur untersucht. Ein Graph G = (V, E) wird formal durch die Knotenmenge (V) und Kantenmenge (E) definiert. Zwei Knoten u, v ∈ V gelten als benachbart oder als Nachbarn, wenn es eine Kante {u, v} ∈ E gibt, die sie miteinander verbindet. Man unterscheidet grundsätzlich zwischen *gerichteten* (Kanten haben eine Richtung haben (u, v) ≠ (v, u)) und *ungerichteten* (Kanten haben keine Richtung (u, v) ∈ E ⇔ (v, u) ∈ E) Graphen (vgl. Tittmann 2019: 11; vgl. Dörpinghaus 2022). Gerichtet Kanten sorgen für eindeutige Interpretation (vgl. Lanquillon/ Schacht 2023: 192).

Ein Wissensgraph dient dazu, Wissen inklusive seiner Beziehungen formalisiert und strukturiert abzulegen. Es basiert auf dem Modell von Tripeln: Subjekt – Prädikat – Objekt. Ziel ist nicht nur die Speicherung, sondern auch die automatisierte Datenbeschaffung, Abfrage und Weiterverwendung (vgl. ebd.: 192) Dadurch wird Wissen nicht isoliert, sondern über Verknüpfungen zugänglich gemacht (vgl. Dengel 2012: 34). Der Satz *Lena fährt Fahrrad* lässt sich durch ein einfaches Semantisches Netz darstellen (Abbildung 2).

fährt

Abb. 2.1 Semantisches Netz für den Satz *Lena fährt Fahrrad*

Wissen wird aber nicht nur in einfache Form eines Tripels dargestellt, kann auch komplexe Konstruktionen, wie Sätze mit mehreren Beteiligten oder Rollen abgebildet werden (vgl. Dengel 2012: 34).

kommt\_aus

fährt\_zu

fährt\_mit

Abb. 2.2 Semantisches Netz für den Satz *Lena fährt mit dem Fahrrad zu einem Freund, der gerade aus Berlin zurück ist.*

Die Verbindungen haben immer eine definierte Richtung, die von einem Knoten zu einem anderen verläuft und können ebenfalls keine oder mehrere Eigenschaften besitzen. Dabei hat jede Beziehung immer einen Start- und einen Endknoten (vgl. Barrasa/ Webber 2023: [Introducing Knowlege Graphs](https://learning.oreilly.com/library/view/building-knowledge-graphs/9781098127091/ch01.html#idm45681067535008)). Die Beziehungen sind oft bidirektional, werden aber häufig in einer bestimmten Richtung benötigt (vgl. Dengel 2012: 34).

Durch ihre flexible Struktur können Graphen verschiedenartige Datenquellen verknüpfen und unterschiedliche Beziehungstypen und Anwendungsbereiche abzudecken (Heist et al. 2021: 128). Unterschiedliche Visualisierungen desselben Graphen können dabei verschiedene strukturelle Merkmale sichtbar machen oder in den Vordergrund rücken. Bei der Darstellung gerichteter Graphen wird die Richtung der Relationen zwischen den Knoten üblicherweise durch Pfeile gekennzeichnet (vgl. Dörpinghaus 2022). Diese Struktur dient dazu, Objekte und deren Eigenschaften mit verschiedenen Formen von Relationen zu verbinden, um sie besser zu verstehen (vgl. Dengel 2012: 35). Darüber hinaus können Wissensgraphen durch hierarchische Ontologie ergänzt werden, um indirekte Relationen abzuleiten. Dadurch können implizite Informationen bei Abfragen berücksichtig werden. Ein weiterer Vorteil von Wissensgraphen ist, dass die Mehrdeutigkeit eines Begriffs aufgelöst wird, indem Begriffe durch semantische Beziehungen eindeutig zugeordnet werden. So wird der Begriff nicht isoliert gespeichert, sondern mit Bedeutung versehen (vgl. Lanquillon/ Schacht 2023: 193).

### Zusammenfassung

## W3C Standards

Die Abbildung 3.1 zeigt die Sprachstandards, die dem Semantik Web zugrunde liegen. Sie stellt die Struktur der verschiedener Ebenen dar, auf denen das Semantic Web aufbauet. Im Unterschied zu früheren Darstellungen des sog. Layer Cake von Berners-Lee, liegt der Fokus auf W3C-Empfelungen.

Ontologie:

OWL

Regeln:

RIF

Query: SPARQL

Datenaustausch:

RDF

XML

Protocol: HTTP/ URI/ IRI

Abb. 3.1 Semantic Web protocol and language standards (Harth/ Janik/ Staab 2011: 50)

IRIs und URIs verwenden typischerweise HTTP-basierte Bezeichner und ermöglichen die Nutzung internationaler Zeichensätze in Web-Adressen. Dadurch können die damit verknüpften Ressourcen eindeutig identifiziert und abgerufen werden. XML wird zur Kodierung und Serialisierung von Dokumenten eingesetzt. Es dient dazu, unstrukturierte Inhalte zu strukturieren. Darüber hinaus ermöglicht XML den systemübergreifenden Austausch strukturierter Daten im Semantic Web. RDF bietet ein graphenbasiertes Datenmodell zur Integration und Beschreibung von Informationen aus unterschiedlichen Quellen. RDFs ergänzt RDF, indem es das Beschreiben von Klassen und Eigenschaften ermöglicht. OWL ist eine Ontologiesprache, mit der sich Wissensdomänen genau beschreiben lassen. Sie ermöglicht automatische Schlussfolgerungen aus den vorhandenen Daten. SPARQL erlaubt gezielte Abfragen auf RDF-Daten (vgl. Harth/ Janik/ Staab 2011: 50–51).

In diesem Kapitel liegt Fokus darauf, die einzelnen W3C-Standards und ihren Zusammenhang im Kontext des Semantic Web zu erläutern. Ziel ist es, aufzuzeigen, wie diese Standards miteinander verknüpft sind und welche Rolle sie beim Aufbau des spielen.

### 3.1 URI

Der Zugriff auf die Daten spielt eine wichtige Rolle bei dem Aufbau des Semantic Web. Dafür benötigt man ein Link sowohl Netzwerkinformationen (Hostname, IP, Port), als auch Protokoll- und Pfandangaben. Der URI ist das übergeordnete Adressierungskonzept für alle Internetressourcen, der vom W3C eingefügt wurde (vgl. Kuhlen et al. 2022: 782–783). Damit können Nutzer:innen problemlos auf verschiedene Server zugreifen und Daten dezentral verknüpfen. Die Technologie URI/ IRI und HTTP bilden zentrale Grundlagen für die dezentrale Adressierung und Datenübertragung im (Semantic) Web (vgl. Harth/ Janik/ Staab 2011: 54).

Ein wesentliches Prinzip ist dabei die Einheitlichkeit der Adressierung: Verschiedene Arten von Ressourcen, wie z. B. Webseite, Dienste, Bilder und nicht-digitale Objekte, wie Bücher oder Personen, können mit einheitlichen URIs adressiert werden, unabhängig von konkreten Zugriffsmechanismus oder verwendeten Anwendung (vgl. Berners-Lee/Fielding/Masinter 2005: 4).

Organisationen können **Daten miteinander verknüpfen, ohne sich vorher abzustimmen oder technische Schnittstellen zu vereinbaren**. Dies ermöglicht das offene und flexible Verknüpfen von Wissen im Sinne des Semantic Web (vgl. Harth/ Janik/ Staab 2011: 54). Wird ein URI verwendet, um ein Vokabular oder Schema zu kennzeichnen (also eine Menge von Begriffen), spricht man von einem Namespace. Namespaces identifizieren die Schemata, in deren Ressourcentypen und Beziehungsarten für RDF-Graphen definiert sind (vgl. Gandon et al. 2011:124). Wenn Veröffentlichung von Daten den **Linked Data-Prinzipien** folgt, entsteht ein **einheitliches System** für **Zugriff und Verknüpfung** von Daten im Semantic Web (vgl. Harth/ Janik/ Staab 2011: 55).

### 3.2 XML/ TEI

Eine Grundvoraussetzung für die Erstellung des Semantic Web sind strukturierte und maschinenlesbare Daten. Um diesen Anforderungen gerecht zu werden, wurde XML als Teil der Standardized Generalized Markup Language (SGML) entwickelt (vgl. Jeckle 2002: 11). XML ist eine Struktursprache, mit der man Inhalte logisch organisieren kann (vgl. Dengel 2012: 114). Es ermöglicht, eine individuelle Auszeichnungssprache zu erstellen und Elemente sowie Attribute nach Bedarf zu definieren (vgl. Ray 2001: 157). Diese Flexibilität erlaubt es, Daten in einem einheitlichen, maschinenlesbaren Format zu erstellen (vgl. Jeckle 2002: 11).

XML wird durch Regeln zur Unterscheidung zwischen Kodierung und Text charakterisiert, indem es Verwendung von Markup-Elementen vorschreibt, die durch spitze Klammern gekennzeichnet sind. Dabei handelt es sich um Elemente (*tags)*, die korrespondierende Paare aus einem Anfangs- (<text>) und einem Endtag (</text>) bilden. Durch die Verwendung von *tags* können Textbereiche markiert und die Bedeutung zugewiesen werden. XML-Elemente können andere Elemente beinhalten und müssen vollständig in einem anderen Element zugewiesen sein. Diese Regel ist die wichtigste Eigenschaft von XML, da sie eine strikt hierarchische Struktur der Kodierung ermöglicht (vgl. Vogeler & Sahle 2017: 128–130). XML dient dazu, bestimmte Teile eines Textdokuments mit zusätzlicher Information sog. Metadaten zu versehen. Jedes XML-Dokument besitzt genau ein Wurzelelement, das alle anderen Elemente einschließt. Elemente in XML können Attribute enthalten, die denselben syntaktischen Einschränkungen unterlegen, wie andere Bestandteile des Dokuments (vgl. Hitzler et al. 2007: 17, 20). Abhängig von der Fragestellung und -komplexität der Forschung, die mit der XML-Auszeichnung zusammenhängt, ändert sich Informationsdarstellung dieser Struktur für Nutzer:innen (vgl. Baillot & Schnöpf 2015: 7–9). XML ist besonders geeignet für die strukturierte Darstellung von Wissen, da es maschinen- und menschenlesbar ist. Darüber hinaus lassen sich heterogene Datenquellen durch Verlinkung und Strukturierung mithilfe XML vereinheitlichen und strukturieren (vgl. Dengel 2012: 114).

XML ermöglicht es, eine individuelle Auszeichnungssprache zu erstellen und Elemente sowie Attribute nach Bedarf zu definieren. Allerdings soll das Vokabular von Elementen und Attributen beschränkt werden, um die Grammatik der Elemente zu kontrollieren. Dieser Prozess wird als Dokumentenmodellierung bezeichnet. Es existiert zwei Methoden zur Modellierung eines Dokuments: Dokumenttyp-Definitionen (DTDs), in denen die Struktur eines Dokuments durch Regeln beschrieben wird, und das XML-Schema, das die Struktur des Dokuments anhand eines Musters mit Elementvorlagen (Templates) beschreibt. Die Modellierung von Dokumenten beinhalten die Definition eines Dokumentenmodells*.* Es beantwortet die Frage, ob ein Element einen Titel haben kann oder ob ein bestimmtes Attribut zwingend erforderlich ist. Es ermöglicht die Überprüfung, ob ein XML-Dokument den Anforderungen des definierten Dokumententyps entspricht (vgl. Ray 2001: 157). In der Regel wird die Modellierung angewendet, wenn es viele Dokumente mit ähnlichen Strukturen gibt. (vgl. Eckstein/ Eckstein 2004: 26). Wenn Datenstrukturen nur einmal verwendet werden, ist ein Model nicht notwendig (vgl. Vonhoegen 2009: 77).

Ein Dokumenttyp oder -Schema dient dazu, die Strukturen einzuschränken, um effizient mit den Dokumenten zu arbeiten und Fehler zu vermeiden (vgl. Eckstein/ Eckstein 2004: 26). Wenn auf ein Datenmodell zur Gültigkeitskontrolle von XML-Dokumenten verzichtet wird, akzeptiert ein normaler XML-Prozessor jeden Elementnamen, der die XML-Namensregeln einhält und keine ungültigen Zeichen enthält. In diesem Fall gibt es keine grammatikalischen Einschränkungen, und ein Element kann beliebigen Inhalt haben, einschließlich Kindelemente, Text, Mischungen aus Elementen und Text oder gar nichts (vgl. Vonhoegen 2009: 77).

Auf Grundlage von XML wurden zahlreche weitere Sprachen entwickelt, die entweder auf XML basieren oder XML als Metagrammatik oder Syntax einsetzen. Ein Beispiel dafür ist die TEI (Text Encoding Initiative), eine weit verbreitete Auszeichnungssprache zur Kodierung literarischer oder wissenschaftlicher Texte (vgl. Vogeler & Sahle 2017: 138). Die TEI stellt eine Richtlinie und keine Norm dar. Sie ist keine statische oder endgültige Zusammenstellung von Auszeichnungen, sondern vielmehr eine dynamische Entität, die erweiterbar und modifizierbar ist. Damit können neue Texte jederzeit in das System integriert werden, ohne die Integrität des Gesamtsystems und der Interoperabilität zu beeinträchtigen (vgl. Bader 1999: 13)

Obwohl XML als universelle Meta-Sprache gilt und erlaubt die Strukturierenu von Informationen, gibt aber keine Bedeutung vor ([vgl. Berners-Lee/ Hendler/ Lassila 2001](https://www.lassila.org/publications/2001/SciAm.pdf)). Tags in XML können für Menschen verständlich sein, doch ihre Bedeutung ist für Maschinen nicht automatisch erkennbar. Damit XML-Daten jedoch zwischen verschiedenen Anwendungen korrekt ausgetauscht werden können, muss die Bedeutung aller verwendeten Tags eindeutig definiert und dokumentiert werden. Dies führt oft zu umfangreicher und schwer nachvollzierbarer Dokumentation. Das Semantic Web verfolgt das Ziel, solche Dokumentationen überflüssig zu machen, indem die Bedeutung von Tags maschinenlesbar und formal eindeutig kodiert wird. Daher dient XML aus Sicht des Semantic Web als technische Grundlage, auf der weiterführende Sprachen wie RDF und OWL aufbauen (vgl. Hitzler et al. 2007: 30).

### 3.3 RDF

RDF ist eine domänenunabhängige Sprache zur semantischen Beschreibung von Datenmodellen (vgl. Baumeister/ Seipel 2005: 40). Damit werden sowohl Datenobjekte als auch deren Relationen auf eine Weise, die Maschinen verstehen und verarbeiten können ([vgl. Feigenbaum ebd. 2007](http://thefigtrees.net/lee/sw/sciam/semantic-web-in-action#page2)). Im Gegensatz zu XML, das vor allem auf korrekte Darstellung von Dokumenten ausgerichtet ist, steht RDF die Kombination und Weiterverarbeitung der enthaltenen Informationen im Vordergrund (Hitzler et al. 2007, S. 34).

RDF besteht aus zwei Komponenten: dem RDF-Modell (einer Menge von Aussagen) und der RDF-Syntax zur Serialisierung und damit zum Austausch (vgl. Dengel 2012: 118). RDF nutzt Triple-Strukturen zur Repräsentation von Bedeutung (Subjekt-Prädikat-Objekt). Jede Aussage wird in Form eines Tripels modelliert, wobei Subjekt und Objekt durch URIs eindeutig identifiziert werden ([vgl. Berners-Lee/ Hendler/ Lassila 2001](https://web.archive.org/web/20070713230811/http:/www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21)) (Abbildung 3.2). Auch wenn Ressourcen nicht online verfügbar sind, werden URIs verwendet. Dabei spielt es keine Rolle, ob der URI tatsächlich existiert (vgl. Hitzler et al. 2007: 38). Zusätzlich erlaub RDF, abstrakte Ressourcen mithilfe von Literale (Zeichenfolgen) als Objekts eines Tripels zu beschreiben (vgl. Domingue/ Fensel/ Hendler 2011: 121–123, vgl. Hitzler et al. 2007: 38). URIs dienen dabei als Referenz auf den gemeinten Dingen und Literale hingegen sind reservierte Bezeichner für RDF-Ressourcen eines bestimmten Datentyps (vgl. Hitzler et al. 2007: 39).

https://www.wikidata.org/wiki/Property:P19

Abb. 3.2 Ein einfacher RDF-Graph zur Beschreibung einer Beziehung zwischen einer Person (Friedrich Wilhelm Kölbing) und einem Geburtsort (Neuwied)

Diese Tripel bilden zusammen einen gerichteten Graphen aus Knoten und Kanten (vgl. Dengel 2012: 110, vgl. Hitzler et al. 2007: 36). Sowohl als Knoten als auch die Kanten besitzen eindeutige Bezeichner. Dadurch wird eine präzise Identifikation von Ressourcen ermöglicht (vgl. Hitzler et al. 2007: 36–37) und so werden bspw. Datenfelder aus unterschiedlichen Datenbanken eindeutig identifiziert und miteinander verknüpft ([vgl. Berners-Lee/ Hendler/ Lassila 2001](https://www.lassila.org/publications/2001/SciAm.pdf)).

Mehrere Tripel werden in der Syntax Turtle dargestellt (vgl. Kuhlen et al. 2022: 260). Mithilfe der *@prefix-Direktiven* können Namespace-Bindungen eingeführt werden, wodurch sich URIs effektiv abkürzen lassen. Zusätzlich ermöglich die Komma-Notation eine kompakte Darstellung von mehreren Anweisungen mit identischem Subjekt-Prädikat-Paar und unterschiedlichen Objekten. Diese Abkürzungen machen Turtle zu dem einfachsten und kompaktesten RDF-Serialisierungsformat (vgl. Gandon et al. 2011: 128–129). Turtle kann sowohl von Menschen als auch von Maschinen gut gelesen und geschrieben werden (vgl. Hitzler et al. 2007: 42). Es ist ein kompaktes, textbasiertes RDF-Format, das abkürzende Schreibweisen erlaubt, sodass Wiederholungen von Subjekten oder der Kombination aus Subjekt und Prädikat nicht erneut angegeben werden müssen. Darüber hinaus erlaubt Turtle auch verschachtelte Strukturen mit Klammern, so bleiben die erzeugten Graphen gut lesbar und leicht verständlich (vgl. Kuhlen et al. 2022: 260–261).

Bei der Modellierung komplexer Beziehungen hat RDF einige Einschränkungen. Es kann bspw. nicht festgelegt, dass für einen Eigenschaft wie *hat\_Kind* die Beziehung je nach Entitätstyp unterschiedlich ist, also dass bei einer Person das Kind ebenfalls eine Person und bei Elefanten ein Elefant ist. Zudem fehlen in RDF die Möglichkeit, Anzahlbegrenzungen festzulegen, wie z.B. *jede Person hat zwei Elternteil* oder *jede Person muss eine Mutter haben*. Darüber hinaus kann RDF keine logischen Eigenschaften wie Transitivität, Symmetrie oder Inversen definieren (vgl. Dengel 2012: 127).

Wenn komplexeres Wissen über Daten auszudrücken werden soll, kommt RDF-Schema (RDFS) zum Einsatz (vgl. ebd.: 124). Das anwendungsspezifische Vokabular eines RDF-Datenmodells wird mithilfe von RDFS sowie ergänzend durch Ontologie in OWL definiert. Im Unterschied zu XML-Schema oder DTD, die die Struktur von XML-Dokumenten durch Einschränkungen definieren, beschreibt das RDFS die Eigenschaften von Objekten sowie deren Beziehungen zueinander und zu ihren Wertebereichen (vgl. Baumeister/ Seipel 2005: 40–41). RDFS stellt dabei ein Vokabular zur Verfügung, mit dem sich Klassen (rdfs:Class, rdfs:subClassOf), Eigenschaften (rdf:Property, rdfs:subPropertyOf) und die Beziehungen zwischen Klassen und Datentypen (rdfs:domain, rdfs:range) definieren lassen. Außerdem ermöglicht RDFS die Angabe von Metadaten zu der Ontologie (vgl. Kuhlen et al. 2022: 262). RDF-Schema erweitert RDF durch die Einführung einer extern spezifizierten Semantik. Während RDF Eigenschaften von und Beziehungen zwischen Ressourcen mithilfe benannter Properties und Werte definiert, stellt RDFS genau diese Mechanismen zur Verfügung (vgl. Dengel 2012: 124).

### OWL

Wie schon gesagt wurde, weist RDF gewisse Defizite auf. Aus diesem Grund wurde OWL als eine Erweiterung der bestehenden Web-Standards wie XML, RDF und RDFS entwickelt (vgl. Dengel 2012: 127). Bei OWL besteht ein Gleichgewicht zwischen der Ausdruckstärke der Sprache und ihrer Skalierbarkeit (vgl. Hitzler et al. 2007: 125). OWL basiert auf Beschreibungslogiken[[2]](#footnote-2) (Prädikatenlogik der ersten Stufe), mit der sich Wissen klar und eindeutig dargestellt lässt. Genauso wie in der traditionellen klassischen Logik muss jeder Satz in OWL entweder wahr oder falsch sein, unabhängig davon, ob es ein behaupteter Sachverhalt, Domänenwissen oder ein Argumentationsergebnis sei. OWL bietet Möglichkeit, Dinge genau zu beschreiben (vgl. Zhongli Ding/Yun Peng 2004: 1–3). Die Sprache liegt in drei Variante vor: OWL Lite, OWL Description Logic (OWL DL) und OWL Full (vgl. Dengel 2012: 127), die für unterschiedliche Anforderungen konzipiert wurden (vgl. [OWL: Overview](https://www.w3.org/TR/owl-features/?utm_source=chatgpt.com)).

Im Unterschied zur klassischen Prädikatenlogik der ersten Stufe erlaub OWL nur einfache Aussagen mit ein- oder zwei Elementen (sog. ein- und zweistellige Prädikaten). Dadurch ist OWL weniger mächtig, aber deutlich leichter zu verwenden. Die Varianten OWL Lite und OWL DL machen den praktischen Umgang zusätzlich einfacher. Außerdem gibt OWL den Daten eine klare Objektstruktur, die in der klassischen Prädikatenlogik so nicht vorgesehen ist (vgl. Dengel 2012: 127). Dadurch werden Arten von Aussagen genau beschrieben. (vgl. Zhongli Ding/Yun Peng 2004: 2).

OWL Lite wurde für eine einfache Implementierung entwickelt und bietet eine funktionale Teilmenge der Sprache, die den Einstieg in die OWL-Nutzer:innen erleichtert. OWL DL wurde entwickelt, um den Anforderungen im Bereich der Beschreibungslogiken gerecht zu werden und eine Sprachteilmenge für logische Systeme bereitzustellen. OWL Full ist die umfassende Variante der OWL-Sprache. Sie hebt einige Einschränkungen von OWL DL auf, ist aber dadurch nicht mehr vollständig mit der Beschreibungslogik kompatibel. Aufgrund ihrer Flexibilität wird sie von vielen Datenbank- und Wissensrepräsentationssysteme unterstützt (vgl. [W3C: Reference](https://www.w3.org/TR/owl-ref/)).

Eine OWL-Ontologie besteht aus Klassen, Eigenschaften (Properties bzw., Rollen) und Individuen, die als RDF-Instanzen von Klassen deklariert werden. Diese werden mithilfe logikbasierter Konstruktionen beschrieben (vgl. Hitzler et al. 2007: 126). Jede Klasse muss durch eine eindeutige URI definiert und als Instanz von *owl:Class* deklariert werden. Gleiches gilt für Eigenschaften, die entsprechend als *owl:ObjectProperty* oder *owl:DatatypeProperty* typisiert werden müssen. Zusätzlich muss jedes Individuum mindestens einer Klasse zugeordnet sein, auch wenn es nur *owl:Thing* ist.

Eine typische Klassendeklaration sieht beispielsweise so aus: <owl:Class rdf:ID="Animal"/>. Damit wird eine Klasse namens *Animal* erstellt, die zu *owl:Class* gehört. Auch Eigenschaften lassen sich hierarchisch organisieren. Mit *rdfs:subPropertyOf* kann etwa festgelegt werden, dass eine Eigenschaft eine Untereigenschaft einer anderen ist (vgl. Zhongli Ding/Yun Peng 2004: 2–3, vgl. [W3C: Reference](https://www.w3.org/TR/owl-ref/)).

OWL verlangt außerdem, dass URIs für Klassen, Eigenschaften und Individuen eindeutig und voneinander abgegrenzt sind. Diese formalen Anforderungen müssen erfüllt sein, damit ein RDF-Dokument als gültiges OWL Lite- oder OWL DL-Dokument gilt (vgl. [OWL: Overview](https://www.w3.org/TR/owl-features/?utm_source=chatgpt.com)).

### SPARQL

Um gezielt Aussagen aus Ontologien und Wissensgraphen zu extrahieren, kommen Abfragesprachen zum Einsatz. Voraussetzung dafür, dass die Graphen in einem Triplestore gespeichert sind und über standardisierte Abfrageschnittstellen zugänglich gemacht wurden. Für solche Abfragen etabliert sich SPARQL (vgl. Kuhlen et al. 2022: 265). SPARQL-Abfragen bestehen in der Regel aus Dreifachmuster, sog. Graph-Muster. Im Gegensatz zu festen RDF-Tripeln können dabei Subjekt, Prädikat und Objekt durch Variablen ersetzt werden (vgl. [SPARQL](https://www.w3.org/TR/sparql11-query/)). Zu Veranschaulichung dient das folgende Beispiel einer SPARQL-Anfrage:

SELECT ?name

WHERE {

?person foaf:name ?name.

}

Die Syntax von SPARQ orientiert sich an SQL und verwendet ähnliche Schlüsselwörter wie *SELECT*, *WHERE*. (vgl. Dengel 2012: 162). Das Schlüsselwort *SELECT* gibt an, dass die Abfrage den Wert der Variable *?name* zurückgeben soll. Durch das Schlüsselwort *WHERE* wird grundlegende Graph-Muster definiert. In diesem Fall besteht er aus Triple-Muster *?person foaf:name ?name* und beschreibt eine Beziehung zwischen Subjekt *(?person*) und einem Objekt *(?name*), verbunden über das Prädikat (*foaf:name*).

Der Musterabgleich umfasst Funktionen wie optionale Teile für unvollständige Daten, Mustervereinigung zur Kombination mehrerer Muster, Verschachtelung und das Filtern von Übereinstimmungen. Zusätzlich kann die die Datenquelle ausgewählt werden. SPARQL-Abfragen können je nach Ziel unterschiedliche Ergebnisse liefern, wie Ja/Nein-Antworten, die Rückgabe von Variablenwerten, die Genirierung neuer RDF-Daten oder Beschreibung von Ressourcen (vgl. Pérez/ Arenas/ Gutierrez 2009: 2).

SPARQL ist auf RDF-Graphen und deren einfache Semantik ausgelegt und stößt daher an Grenzen, wenn es um die Verarbeitung komplexerer semantischer Strukturen wie in RDFS oder insbesondere OWL DL geht. Ein Grund dafür liegt in der Modellierung von OWL-Ontologien, die häufig aus einer Vielzahl möglicher Modelle bestehen und somit die Abfrage erschweren. In OWL DL werden Eigenschaften als binäre Relationen zwischen Elementen dargestellt, die sich als Graphen modellieren lassen (vgl. Hitzler 2007: 233–234). Um die Abfrage solcher OWL-DL-Wissensbasen dennoch zu ermöglichen, wurde vom OWL-Reasoner ***Pellet*** eine angepasste Teilmenge von SPARQL entwickelt. Diese modifizierte Anfragesprache wurde speziell darauf ausgelegt, mit den logikbasierten Strukturen von OWL DL kompatibel zu sein (vgl. Dengel 2012: 169).

### Zusammenfassung

1. Eine ausführliche Darstellung der zugrunde liegenden Technologien des Semantic Web erfolgt in Kapitel 3 dieser Arbeit. [↑](#footnote-ref-1)
2. Beschreibungslogiken sind spezielle **formale Sprachen**, mit denen **Wissen in strukturierter Form dargestellt werden kann**. Sie helfen dabei, Begriffe, Eigenschaften und Zusammenhänge aus einem bestimmten Themenbereich **klar und logisch** zu beschreiben. Dabei werden einfache Bausteine wie Begriffe (Konzepte) und Beziehungen (Rollen) verwendet, die mithilfe logischer Konstruktionen zu Ausdrücken kombiniert werden. Diese Ausdrücke bestehen aus atomaren Konzepten (unären Prädikaten) und atomaren Rollen (binären Prädikaten) (vgl. Baader/ Horrocks/ Sattler 2008: 2). [↑](#footnote-ref-2)