

2 Wissensrepräsentation

Übersicht

2.1	Prinzipien der maschinellen Wissensverarbeitung	23
2.2	Fakten und Regeln	23
2.3	Logik und Inferenz	27
2.4	Prädikatenlogik	30
2.5	Semantische Netze	33
2.6	Frames	39
2.7	Taxonomien	48
2.8	Conceptual Dependencies	52
2.9	Taxonomien mit formaler Semantik: Die KL-ONE-Familie	58
2.10	Ontologien	64
2.11	Fazit	71
2.12	Weiterführende Literatur	72

In diesem Kapitel möchten wir näher erläutern, wie sich Wissen in seinen unterschiedlichsten Formen und Ausprägungen auf dem Computer repräsentieren lässt und wie es formal verarbeitet und damit interpretiert werden kann.

Wissen ist grundlegend für unsere Anpassungsfähigkeit und bestimmt die Qualität, mit der wir alltäglich Probleme lösen und Entscheidungen treffen. Es umfasst u. a. Erfahrungen und Kenntnisse über die Welt, die Dinge und Lebewesen, die sie beherbergt, sowie ihre Eigenschaften und Beziehungen zueinander, aber auch Spezialwissen über bestimmte Gebiete, das wir uns im Laufe unseres Lebens aneignen. Wissen ermöglicht uns, Objekte zu unterscheiden, zu abstrahieren, zu assoziieren, schöpferisch kreativ zu sein oder Visionen und Ziele zu entwickeln. Im Umgang mit alltäglichen Sinneseindrücken aus unserer Umwelt und deren Interpretation befähigt uns Wissen zu zielgerichtetem, adaptivem Handeln, also zur Durchführung von Aktionen.

Seit Erfindung der Schrift ist das explizite Niederschreiben ein erprobtes Vehikel, Wissen zu explizieren und es in Form von Information über Ort und Zeit hinweg anderen Menschen verfügbar zu machen. Die Leser eines Buches können – vorausgesetzt, sie sind in der Lage, den Inhalt zu verstehen – aus dem Gelesenen ihre Schlüsse ziehen, ihr Wissen erweitern und entsprechend agieren. Dieses seit Jahrtausenden bewährte Prinzip erhält heute dank der allgegenwärtigen Verfügbarkeit scheinbar unendlicher

Informationsmengen im Internet ganz neue Möglichkeiten – und stößt gleichzeitig offensichtlich an die Grenzen menschlicher Informationsverarbeitung.

Können wir Computer befähigen, an unserer Stelle auf Informationen oder Umweltreize zu reagieren und in einer dem Menschen vergleichbaren Weise „intelligent“ zu handeln? Ist es insbesondere möglich, Wissen in einer Form zu repräsentieren, dass Computer auf dessen Grundlage in nützlicher Weise handeln können?

Solche Fragen stehen im Fokus der KI, der Künstlichen Intelligenz (Dengel 1999). Diese Wissenschaft befasst sich mit den Grundlagen und der Erstellung von Computersystemen, die ein Verhalten realisieren und aufweisen, welches beim Menschen als intelligent angesehen wird. Konsequenterweise betrachten wir repräsentiertes Wissen als eine Grundlage und Voraussetzung des vom Computer simulierten menschlichen Verhaltens.

Dies kann beispielsweise bedeuten, die kognitive Fähigkeit der Kategorisierung auf den Computer zu übertragen. Der Computer soll also in der Lage sein, selbstständig zu entscheiden, ob ein gegebenes Objekt zu einer Kategorie gehört oder nicht, z. B. ob Gesine Mustermann ein Mensch ist oder ob Konzepte wie Rudi und Gesine zur gleichen Kategorie gehören.

Solche Kategorien sind Vorstellungen, wie sie in unseren mentalen Modellen entstanden sind. Viele Theorien (z. B. Collins und Quillian 1969) gehen davon aus, dass solche Kategorien über das Vorhandensein bestimmter Merkmale oder Attribute bestimmt werden, was bedeutet, dass die mentalen Kategorien in einzelne Teilbedeutungen zerlegt werden können, also dekomponierbar sind. Semantische Technologien sind eine technische Basis, solche wissensbasierten Prinzipien auch in offenen und komplexen Informationswelten – wie etwa im Kontext von Unternehmen, Lieferketten oder im Internet – zu realisieren und so aus einer großen Informationsmenge einen reichhaltigen Wissensschatz zu erzeugen, der vielfältige nützliche Computerdienste ermöglicht. Entsprechende Aspekte werden wir in den späteren Kapiteln noch aufarbeiten und eingehender beleuchten.

In den folgenden Abschnitten wollen wir uns eingehend mit der Repräsentation von Wissen und der Modellierung von Domänen beschäftigen. Wir werden zunächst einige grundlegende Prinzipien (Abschnitt 2.1) erläutern, bevor wir uns der Nutzung von Fakten, Regeln (Abschnitt 2.2) und der Prädikatenlogik (Abschnitt 2.4) als Wissensrepräsentationsformalismus widmen. Danach werden wir uns den Semantischen Netzen (Abschnitt 2.5) zuwenden, Schemata in ihren wesentlichen Ausprägungen kennen lernen (Abschnitt 2.6) und uns mit den Prinzipien der abstrahierenden Konzeptualisierung von Wissen mit Hilfe von Taxonomien auseinandersetzen (Abschnitt 2.7). Als weiteres wichtiges Element der Wissensrepräsentation geht das Kapitel auf die Repräsentation von Ereignisfolgen in Form von Skripts ein (Abschnitt 2.8). Als Vorstufe von ontologischem Wissen werden schließlich formale Methoden der taxonomischen Beschreibung im Rahmen der KL-ONE Sprachen aufgegriffen (Abschnitt 2.9). Zum Abschluss des Kapitels werden wir Ontologien behandeln (Abschnitt 2.10), welche für Semantische Technologien in den letzten Jahren eine zentrale Bedeutung erlangt haben. Abschließend

ziehen wir ein Fazit (Abschnitt 2.11) und geben Hinweise auf weiterführende Literatur zu den angesprochenen Themen (Abschnitt 2.12).

2.1 Prinzipien der maschinellen Wissensverarbeitung

Das Wissen von Menschen mit Hilfe des Computers zu modellieren ist keine leichte Aufgabe. Viele geistige Leistungen des Menschen gehen auf eine intuitive Bewertung zurück und sind oft weder berechenbar noch nachvollziehbar. Die meisten logischen Umsetzungen kreativer Ideen sind emotional getrieben, also von komplexen Strukturen subjektiver Empfindungen tief in unserem Inneren. Emotionales Wissen schärft unseren Verstand und erlaubt es uns, den über unsere Sinnesorgane aufgenommenen Informationsfluss zu bewerten und Entscheidungen für Aktionen zu treffen. Der einfache Umgang und das Rechnen mit Zahlen reicht nicht aus, um eine gefühlsmäßige Vorstellung von Dimensionen, Zeit und Ausdehnung zu bekommen, was aber oft unumgänglich ist, um die konkreten situativen Gegebenheiten im Kontext richtig einzuschätzen. Wissen ist komplex, vielschichtig und daher nur bedingt begreifbar.

Um Menschen und Maschinen miteinander kommunizieren zu lassen, muss die ausgetauschte Mitteilung die gewünschte Information repräsentieren. Da Sprache, sei sie geschrieben oder gesprochen, eindimensional ist, unsere Umgebung in ihrem räumlichen und zeitlichen Kontext eines Ereignisses oder einer Situation jedoch mehrdimensional und vielschichtig, muss man sich Formalismen zu Nutze machen, welche die gegebenen Umstände und Rahmenbedingungen eines Ereignisses oder einer Situation in Zusammenhang bringen, welche also ein maschinelles Verstehen erlauben.

Wesentliche Teile des menschlichen Wissens lassen sich begrifflich und präzise so beschreiben, dass sie auf einem Computer formal darstellbar und verarbeitbar sind. Dazu hat die Künstliche Intelligenz in den letzten Jahrzehnten umfassende Technologien entwickelt, welche es einem Computer erlauben, mit einer methodischen Vorschrift und formalem Wissen ausgestattet, über aktuell vorliegende Daten Schlussfolgerungen zu ziehen. Die Art und Weise, wie Wissen formalisiert wird, nennt man *Repräsentation* (engl. knowledge representation), die Vorschrift zum Schlussfolgern heißt *Inferenzmechanismus*, *Inferenzregel* oder kurz *Inferenz*.

2.2 Fakten und Regeln

2.2.1 Formalisierung von Aussagen

Um das Prinzip der Repräsentation und Verarbeitung von Wissen zu veranschaulichen, möchten wir auf die zu Beginn dieses Buches eingeführte Beispielwelt von Rudi Baispilog zurückgreifen und um einige Aspekte erweitern. Dazu sollen aus unserem Anwendungsbereich folgende umgangssprachliche Aussagen betrachtet werden:

„Rudi Baispilov ist 38 Jahre alt. Vor elf Jahren hat er Anke geheiratet. Sie haben zwei noch minderjährige Kinder, Cindy und Bert, und leben in Ottobrunn bei München. Rudi arbeitet bei der CarFS AG, einem großen Financial Service Unternehmen, der zu einem großen deutschen Automobilhersteller gehört und seinen Sitz in München, in der Schrickelstraße 11-13, hat. Durch die Nähe zu seinem Wohnort braucht Rudi täglich nur etwa 20 Minuten mit der S-Bahn.“

Aufgrund der in dem Text enthaltenen Aussagen fällt es leicht, uns einen ersten Eindruck von Rudis Verhältnissen zu machen. Wir können darüber hinaus versuchen, den Text aufzuspalten und die Aussagen des Textes aufzuschreiben. Also etwa:

- (1) Rudi „hat-Alter“ 38.
- (2) Rudi „hat-geheiratet“ Anke.
- (3) Cindy „ist-Kind-von“ Rudi.
- (4) Bert „ist-Kind-von“ Rudi.
- (5) Rudi „hat-Adresse“ Ottobrunn.
- (6) Ottobrunn „liegt-bei“ München.
- (7) Rudi „arbeitet-für“ CarFS.
- (8) CarFS „ist-ein“ Finanzdienstleister.
- (9) Finanzdienstleister „ist-ein“ Unternehmen.
- (10) CarFS „hat-Adresse“ München.

2.2.2 Fakten

Jede der aufgeführten Aussagen ist als ein Tripel dargestellt, das einen elementaren Satz der Form Subjekt-Prädikat-Objekt wiedergibt, einer Darstellungsform, mit der wir uns in Kapitel 4 noch eingehender beschäftigen werden. Subjekt und Objekt sind dabei Begriffe, welche die Informationselemente unserer Umgebung oder eines Diskursbereiches darstellen. Das Prädikat bezeichnet eine wahre Beziehung, die zwischen den jeweiligen Informationselementen besteht. Aussagen folgen somit einer Syntax und damit einer formalen Vorschrift. Weiter können wir erkennen, dass jede Aussage für sich steht und damit nicht mehr abhängig von der gerade beschriebenen Kurzgeschichte ist. Jede Aussage (1) bis (10) stellt ein Faktum dar und kann in jedem anderen Zusammenhang (wieder) verwendet werden, z. B. um Fragen zu beantworten.

Für uns Menschen ist es einfach herauszufinden, wie groß der Haushalt von Rudi ist, bzw. aus den Aussagen des Textes weiter zu schließen, dass mit der täglichen Fahrt wohl die Anfahrt zu seinem Arbeitsplatz gemeint ist, ohne dass beide Sachverhalte explizit im Text stehen. Aufgrund unserer Sprachkompetenz erschließt sich uns unmittelbar, dass „Rudi“, „Anke“ und „Cindy“ als Namen von Menschen zu verstehen sind, während „München“ eine Stadt bezeichnet, zumindest aber einen Ort bezeichnen muss, selbst wenn wir diesen nicht kennen würden.

Auch die in unserer Tripel-Notation gewählten Prädikate „hat-Alter“ oder „ist-ein“ sind von ihrer Bedeutung her für uns Menschen unmittelbar einsichtig. Für einen Computer sind alle diese Wörter jedoch zunächst Zeichenketten ohne weitere intrinsische Bedeutung. Damit die Semantik der Aussagen für den Computer erfassbar wird – damit also passende Aktionen daraus abgeleitet werden können –, muss dem Computer eine Brücke zwischen der syntaktischen Darstellung und der daraus resultierenden Aktion gebaut werden.

2.2.3 Inferenz mit Regeln

Betrachten wir die Beziehung „ist-ein“, die in den Aussagen (8) und (9) enthalten ist. Es ist sicher nützlich, wenn der Computer – wie der Mensch – diese Relation als transitiv (vgl. Abschnitt 2.5.2) verstünde und so aus den beiden Aussagen automatisch erkennen könnte, dass auch die Aussage

(11) CarFS „ist-ein“ Unternehmen

gilt. Diese Form der Wissensherleitung nennt man Schlussfolgern oder auch *Reasoning*. Allgemein versteht man darunter die Ableitung von bisher Unbekanntem aus Bekanntem (Shapiro 1992). Unter Inferenz *Inferenz* versteht man dabei elementares und formales Schlussfolgern, was prinzipiell auch mit Hilfe eines Computers stattfinden kann.

Die für uns offensichtliche Transitivität in den Aussagen (8) und (9) ist jedoch für einen Computer nicht selbstverständlich zu erkennen. Wir müssen sie ihm erst beibringen, d. h. wir müssen Wissen bereit stellen, so dass der Inferenzmechanismus diese Relation entsprechend verarbeiten kann. Es gibt verschiedene Möglichkeiten, solches Wissen als formalen Zusammenhang zu repräsentieren. Eine Option besteht in der Verwendung von Regeln, mit deren Hilfe ein Computer dann Sachverhalte aus anderen Aussagen inferiert. Die Transitivität der Beziehung „ist-ein“ müsste etwa formuliert werden als

(12) **wenn** X „ist-ein“ Y **und** Y „ist-ein“ Z, **dann** X „ist-ein“ Z

wobei X, Y und Z als Variable für beliebige Objekte stehen. Ein Computerprogramm – der Inferenzmechanismus –, das die Variablen mit den verschiedenen Fakten unseres Beispiels verbinden kann und auch die Bedingungen und Konsequenzen der Regel syntaktisch korrekt verarbeitet, wird auf dieser Grundlage die gewünschte Schlussfolgerung ziehen können.

Natürlich könnten wir die korrekte Verarbeitung der Relation „ist-ein“ auch in einem speziellen Computerprogramm in der Programmiersprache unserer Wahl realisieren, ohne die Regel (12) explizit zu formulieren. Jedoch wäre eine solche Lösung im Einzelfall in ihrer Anwendbarkeit begrenzt und für den Betrachter nicht ohne weiteres sichtbar – im Gegensatz zu der eingängig formulierten Regel. Der für die Regelverarbeitung realisierte Inferenzmechanismus hingegen kann problemlos eine Vielzahl von neuen

Zusammenhängen verarbeiten. Nehmen wir an, wir wissen beispielsweise weiter, dass gilt:

(13) Rudi „ist-ein“ Mann

(14) Anke „ist-eine“ Frau

dann können wir Regeln aufstellen, die beschreiben, wann die Eigenschaften *Mutter* und *Vater* gelten, also:

(15) **wenn** Y „ist-Kind-von“ X **und** X „ist-ein“ Mann, **dann** X „ist-Vater-von“ Y

(16) **wenn** Y „ist-Kind-von“ X **und** X „ist-eine“ Frau, **dann** X „ist-Mutter-von“ Y

Die Verwendung der Variablen X und Y entspricht einer Verallgemeinerung, so dass die beiden Regeln folgende Aussagen machen:

- Alle Männer, die Kinder haben, sind Vater bzw.
- Alle Frauen, die Kinder haben, sind Mutter

Weitere Regeln lassen sich für andere Zusammenhänge aufstellen, wie etwa für Adressen oder die Angestellten-Beziehung:

(17) **wenn** X „hat-Adresse“ Y **dann** Y „ist-ein“ Ort

(18) **wenn** X „arbeitet-für“ Y **und** Y „ist-ein“ Unternehmen,
dann X „ist-Angestellter-von“ Y

Aus diesen Darstellungen und Möglichkeiten wird bereits ein wesentliches Prinzip und Modellierungsziel jeder Wissensrepräsentation deutlich: Die Gegebenheiten der zu modellierenden Realität sollen *Wissen!deklaratives* dargestellt werden. Das bedeutet, dass wir Ausdrucksweisen verwenden, die das Wissen in Form von als wahr geltenden Aussagen niederschreiben (und nicht etwa versteckt als Ergebnis eines Programmblaufs). Wir gewinnen durch die deklarative Darstellung i. d. R. an Verständlichkeit, Klarheit (und damit auch Präzision) und Flexibilität der Verarbeitung. Will man weitere Schlussfolgerungen ermöglichen, so reicht es aus, die Wissensbasis um neue Fakten und/oder neue Regeln zu erweitern.

Hier werden die Stärken einer solchen formalen Darstellung deutlich. Fakten und Regeln bilden jeweils eigenständige Aussagen, welche jede für sich in bestimmten Zusammenhängen verwendet werden können. Die Hinzunahme von neuem Wissen bedeutet daher i. d. R., dass das bestehende Wissen nicht angetastet werden muss. Da das neue Wissen der gleichen formalen Syntax folgt, können die Inferenzregeln ohne Veränderung auch das neue Wissen verarbeiten, Schlussfolgerungen ziehen und damit neue Probleme lösen oder Fragen des Benutzers beantworten.

2.2.4 Anfragen

Für das bisher bekannte Wissen in unserem Beispiel können etwa Anfragen über personelle Zusammenhänge im Unternehmen oder über den Status von Mitarbeitern,

wie „Welche Orte gibt es?“ oder „Wessen Vater ist Rudi?“ mit Hilfe von Variablen – dargestellt durch die Begriffe *was* und *wem* – formuliert werden:

(19) *Was* „ist-ein“ Ort?

(20) Rudi „ist-Vater-von“ *wem*?

Anfragen wie diese folgen der gleichen Subjekt-Prädikat-Objekt-Syntax, mal nach dem Subjekt fragend, mal nach dem Objekt. Der Inferenzmechanismus findet dann in den vorhandenen Fakten und Regeln die entsprechenden Antworten (von denen es im Beispiel jeweils mehrere gibt), indem er gültige Fakten sucht und Werte für die Variablen findet.

Wir können also festhalten, dass Wissensrepräsentation in deklarativer Weise Aussagen über die zu modellierende Realität darstellen kann. Dazu werden Fakten (in Form von Objekten und Beziehungen) und Regeln formuliert. Hieraus können Inferenzmechanismen weitere Fakten durch Reasoning (Inferenz) ableiten.

Die Semantik der Wissensrepräsentation manifestiert sich in den Operationen, die ein Computer auf Basis des repräsentierten Wissens ausführen kann. Die Ausdrucksmächtigkeit einer Wissensrepräsentation ist also untrennbar mit der Leistungsfähigkeit der jeweiligen Inferenzregel verknüpft. Letztlich lassen sich alle in Frage kommenden Ansätze auf die mathematisch präzise Ausdrucksweise der Logik und der verschiedenen Kalküle (also Inferenzvorschriften) abbilden, weshalb wir diesen Aspekt in den nächsten Abschnitten näher beleuchten.

2.3 Logik und Inferenz

Die Fähigkeit, Wissen abzuleiten oder zu inferieren, ist zentraler Bestandteil menschlicher Problemlösungsfähigkeit. Wir schließen aufgrund bestehender Sachverhalte, dass bestimmte plausible Ereignisse schon stattgefunden haben oder stellen Vermutungen an, indem wir Zusammenhänge von Ereignissen deuten.

Es ist eines der Hauptanliegen der KI solche Inferenztechniken zu untersuchen und formal zu beschreiben. Dazu bedient sie sich der Logik, die sich unter Umständen als Repräsentationsformalismus für Wissen eignet. Die Mächtigkeit der mathematischen Logik erlaubt es, das menschliche Denken zu abstrahieren und Wissensbasen zu konstruieren, auf deren Inhalten Schlussfolgerungen gezogen oder neue Sachverhalte hergeleitet werden können. Wir wollen uns zunächst einige wichtige Grundlagen der Aussagenlogik ansehen, bevor wir zur Prädikatenlogik kommen, welche Grundlage vieler Systeme der Künstlichen Intelligenz ist.

2.3.1 Logische Operatoren

Nehmen wir an, A sei die Aussage „Rudi wohnt in Ottobrunn“ und B die Aussage „Ottobrunn liegt bei München“, dann entspricht der Ausdruck $A \wedge B$ der Aussage „Rudi wohnt in Ottobrunn und Ottobrunn liegt bei München“. Auf gleiche Weise lassen sich Aussagen mit einem „oder“ verbinden oder negieren. Die beiden Symbole A und B sind Aussagenvariablen und können folglich beliebige Aussagen repräsentieren. Je zwei Aussagenvariablen können mit Hilfe von logischen Operatoren oder Junktoren in Formeln verknüpft werden (eine Ausnahme stellt die Negation dar, die sich ja nur auf eine Aussagevariable bezieht).

Man unterscheidet dabei:

\neg (nicht), \wedge (und), \vee (oder), \rightarrow (Implikation), \leftrightarrow (Äquivalenz)

Die letzten beiden Operatoren sind auch identisch mit den verständlichen Umschreibungen „wenn-dann“ und „genau-dann-wenn“. Mit Hilfe der Junktoren lassen sich Ausdrücke bilden, die gemäß der zweiwertigen Logik entweder „wahr“ oder „falsch“ sind, wobei der Wahrheitswert eines Ausdrucks davon abhängt, ob die enthaltenen Aussagenvariablen erfüllt sind oder nicht. Ein Ausdruck $A \wedge B$ ist dann „wahr“, falls A „wahr“ ist und B „wahr“ ist. Den Zusammenhang von Wahrheitswerten in der Verknüpfung mit Junktoren veranschaulicht die folgende Tabelle 2.1, wobei die Werte „wahr“ und „falsch“ mit den Ziffern 1 und 0 dargestellt sind:

Tab. 2.1 Wahrheitswerte für die Junktoren der Aussagenlogik.

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

2.3.2 Inferenzmechanismen

Auf Grundlage von Ausdrücken der Aussagenlogik können Inferenzregeln angewandt werden, die aus bereits bekannten wahren Aussagen (Prämissen) neue wahre Aussagen (Konklusionen) ableiten. Eine der wichtigsten dieser aussagenlogischen Inferenzmechanismen ist der *Modus Ponens*, eine bereits in der Antike bekannte Schlussfolgerungsform, dem die Formel $(A \wedge (A \rightarrow B)) \rightarrow B$ entspricht. Die Anwendung dieser Formel lässt uns aus bekannten Fakten und Zusammenhängen neue Fakten herleiten; dieses so

genannte *deduktive Schließen* geschieht nach folgender Struktur:

$$\begin{array}{ll} A & \text{(Prämisse 1)} \\ A \rightarrow B & \text{(Prämisse 2)} \\ \hline B & \text{(Konklusion)} \end{array}$$

Der *Modus Ponens* erlaubt es, aus gültigen Aussagen (hier: A) und definierten Zusammenhängen oder Regeln $A \rightarrow B$ (was so viel heißt, wie „wenn A gilt, dann gilt auch B“ oder kurz „aus A folgt B“) neue wahre Aussagen herzuleiten.

Diese Form der Inferenz haben wir indirekt bereits kennen gelernt, denn um beispielsweise die Anfrage unter (19) im letzten Abschnitt zu beantworten, müssen wir eine Belegung, sprich einen gültigen Wert, für die Variable *was* finden, welcher der Anwendung von Regel (17) genügt. Gemäß *Modus Ponens* kann die Struktur dieser Herleitung wie folgt dargestellt werden:

$$\begin{array}{ll} \text{Rudi „hat-Adresse“ Ottobrunn} & \\ x \text{ „hat-Adresse“ } y \rightarrow y \text{ „ist-ein“ Ort} & \\ \hline y \text{ „ist-ein“ Ort} & \end{array}$$

Wenn man die noch freie Variable *y* an den Wert „Ottobrunn“ bindet, erhält man die entsprechende Antwort. Das Ergebnis stellt wiederum eine wahre Aussage dar.

Mit einer Inferenzregel ausgestattet müssen bei einem aussagenlogischen System demnach nicht alle Fakten in der Wissensbasis gespeichert werden. Es reicht aus, allgemeingültige Zusammenhänge anzugeben, damit das System gewisse Fakten eigenständig herleiten kann. Diese Besonderheit deutet an, wie vorteilhaft die Aussagenlogik für die Wissensrepräsentation und -verarbeitung ist.

Der *Modus Ponens* ist ein klassisches Beispiel für das sogenannte deduktive Schließen, das grundlegend für alle formalen Systeme ist. Deduktion ist eine logisch korrekte, d. h. wahrheitserhaltende Inferenz, mit deren Hilfe spezielle Tatsachen aus allgemeinen Zusammenhängen gefolgert bzw. hergeleitet werden. Wie in der Mathematik werden so Ergebnisse aus Axiomsystemen gefolgert. Während also deduktive Prinzipien das Vorhandensein von gültigen Prämissen grundsätzlich verlangen, gibt es auch andere Formen der Inferenz, welche nicht diesen Grad der Bestätigung verlangen.

Dazu wollen wir uns wieder ein Beispiel ansehen. Nehmen wir auf unser Szenario zurückgreifend an, wir würden in einem Gespräch erfahren, dass Gesine an einem Ort mit Namen Kaiserslautern arbeitet, dann würden wir intuitiv schließen, dass Gesine damit auch eine Adresse in Kaiserslautern hat.

Diese Form der Inferenz, bei der wir aus einem bekannten Zusammenhang und einem beobachteten Resultat annehmen, dass die Voraussetzung gilt, nennt man *abduktives Schließen* oder kurz *Abduktion*. Die entsprechende Formel lautet $((A \rightarrow B) \wedge B) \rightarrow A$ und entspricht der folgenden logischen Struktur:

Kaiserslautern „ist-ein“ Ort

x „hat-Adresse“ $y \rightarrow y$ „ist-ein“ Ort

x „hat-Adresse“ Kaiserslautern

Wie das Beispiel zeigt, sind Abduktionen im Gegensatz zu Deduktionen nicht notwendigerweise korrekt, sondern eher heuristischer Natur, denn Gesine könnte auch eine ganz andere Adresse, z. B. in einem der schönen Kaiserslauterer Vororte haben. Abduktionen bezeichnen Rückschlüsse auf plausible Ursachen bei Betrachtung der auftretenden Effekte und sind beispielsweise bei der Suche nach möglichen Erklärungen von großem Nutzen. Die Korrektheit des gefundenen Ergebnisses bleibt aber zu überprüfen.

Eine weitere Möglichkeit der Inferenz stellt die sogenannte *Induktion* dar. Während beim deduktiven Schließen aus gültigen, eher allgemeinen Aussagen auf konkrete Fälle geschlossen wird, folgt induktives Schließen eher einem umgekehrten Ansatz. Induktion ist auch keine wahrheitserhaltende Inferenz, denn es werden eher allgemeine Zusammenhänge aus speziellen Tatsachen abgeleitet, etwa wird auf Grundlage von zwei oder mehreren Fällen oder Fakten auf einen allgemein gültigen Satz geschlossen; die gemachten einzelnen Beobachtungen werden also zu einer Regel abstrahiert.

Der Mangel dieses Inferenzverfahrens wird am folgenden Beispiel deutlich:

CarFS „hat-Adresse“ München

CarFS „ist-ein“ Unternehmen

x „ist-ein“ Unternehmen $\rightarrow x$ „hat-Adresse“ München

Das Ergebnis der Anwendung in diesem Fall würde soviel bedeuten wie „Alle Unternehmen haben eine Adresse in München“ und damit eine falsche Aussage darstellen. Die Prämissen beim induktiven Schließen bestehen in diesem Beispiel darin, dass auf einem einzigen Fall – es könnten auch mehrere sein – begründet, angenommen wird, dass eine Klasse in einer anderen enthalten ist, was natürlich nicht zutrifft.

Zusammenfassend lässt sich festhalten, dass Deduktion im Gegensatz zu Induktion und Abduktion folgerichtige und wahrheitserhaltende Inferenzregeln darstellen. Induktion oder Abduktion verwendet man also zur Herleitung eines Schlusses nur dann, wenn Deduktion nicht möglich, zu aufwändig ist oder nicht zum Ziel führt. Obwohl abduktives und induktives Schlussfolgern also logisch nicht korrekt sind, d. h. die damit erzeugten Schlüsse sind nicht notwendigerweise richtig, bilden sie wesentliche Vorgehensweisen des Menschen ab. Abduktion erlaubt, mutmaßliche Erklärungen zu beobachteten Ereignissen zu finden. Induktion hilft uns, Beobachtungen zu abstrahieren und so zu neuen Vermutungen über allgemeine Gesetzmäßigkeiten zu kommen. Damit sind diese Schlussweisen für die KI und die semantischen Technologien von elementarer Bedeutung; die Korrektheit der damit gefundenen Ergebnisse muss dann im Einzelfall überprüft werden. Die Aussagenlogik ist aufgrund ihrer begrenzten Ausdrucksmittel nur eingeschränkt zur Wissensdarstellung geeignet. Wir wollen uns daher mit der umfassenderen Prädikatenlogik beschäftigen, die es uns erlaubt, nicht nur Aussagen als Ganzes, sondern die enthaltenen Objekte selbst zu beschreiben.

2.4 Prädikatenlogik

Die Prädikatenlogik ist eine grundlegende Form der Wissensdarstellung, die automatische Inferenzen universell und direkt unterstützt. Ausdrücke in der Prädikatenlogik sind denen der Aussagenlogik sehr ähnlich, erweitern diese jedoch. In der Aussagenlogik gibt es keinerlei Möglichkeiten, sich auf Objekte in Aussagen zu beziehen. Die Aussage „Rudis Kind Bert ist minderjährig“ ist in der Aussagenlogik zwar gültig, der direkte Bezug auf die Objekte „Rudi“ und „Bert“ ist jedoch nicht möglich.

2.4.1 Symbolformen

In der Prädikatenlogik kann man vielmehr einen betrachteten Weltausschnitt, bestehend aus Objekten unterschiedlicher Objekttypen, etwa Individuen, Eigennamen oder Elementen, beschreiben. Als Grundlage zur Beschreibung von Objekten dienen *Konstantensymbole*, wie z. B.:

Rudi, 38, CarFS, München, Ottobunn, *wahr*, *falsch*.

Unter Verwendung von *Prädikatssymbolen* lassen sich Merkmal, Eigenschaften und Beziehungen der Konstantensymbole qualitativ beschreiben. Angenommen männlich steht für ein Prädikatssymbol und Rudi für ein Konstantensymbol, dann führt die Aussage männlich(Rudi) in diesem Fall zu einer wahren Aussage. Beispielsweise wird die Wortfolge x „hat-geheiratet“ Anke zu einer wahren Aussage, wenn man für die Variable x den Wert Rudi einsetzt. Es gibt also ein- und mehrstellige Prädikate, die folgender Syntax folgen:

männlich(Rudi), hat-geheiratet(Rudi, Anke), fährt-mit(Rudi, S-Bahn, 20)

Das Symbol „hat-geheiratet“ steht hier für ein Prädikatensymbol, die Zeichenketten „Rudi“, „Anke“ und „S-Bahn“ oder die Zahl 20 stehen für Konstantensymbole. Als weitere Symbolmenge dienen die sogenannten *Funktionssymbole*, um Abbildungen von Elementen eines Definitionsbereichs auf Elemente derselben Menge vorzunehmen. Prädikate können mit Hilfe der bereits bekannten logischen Junktoren verknüpft werden. *Variablensymbole* schließlich repräsentieren beliebige Elemente und erlauben allgemeingültige Aussagen für Teilmengen von Elementen eines Definitionsbereichs.

2.4.2 Quantoren

Zusätzlich zu den Junktoren werden als Ausdrucksmittel sogenannte *Quantoren* verwendet, die Aussagen darüber erlauben, für wie viele Elemente eines Definitionsbereichs ein Prädikat zutrifft, also sich einschränkend bzw. verallgemeinernd auf die Gültigkeit von

Aussagen auswirken. Der *Existenzquantor* \exists legt fest, dass ein Prädikat für mindestens ein Element gilt, also $\exists x, y$ bedeutet „Es gibt ein x und ein y , für die gilt...“. Der *All-* oder *Universalquantor* \forall sagt dahingegen aus, dass Prädikate für alle Elemente eines Definitionsbereichs zutreffen. Also schreibt man beispielsweise für die Aussage „Es gibt jemanden, der mit Anke verheiratet ist“ folgende Formel:

$$\exists x \text{ ist-verheiratet}(x, \text{Anke})$$

Dabei steht x für ein Variablensymbol, das in der Art seiner Nutzung aussagt, dass mindestens ein Element des Definitionsbereichs die beschriebene Aussage wahr macht.

Quantoren sind mächtige Ausdrucksmittel. Mit ihnen lassen sich unterschiedlichste Aussagen formulieren, beispielsweise:

$\forall x \exists y \text{ hat-geheiratet}(x, y)$	heißt	„Jeder hat mindestens Einen geheiratet“
$\exists x \forall y \text{ hat-geheiratet}(x, y)$	heißt	„Es gibt Jemanden, der Alle geheiratet hat“
$\exists x \forall y \text{ hat-geheiratet}(y, x)$	heißt	„Es gibt Jemanden, der von Allen geheiratet wurde“
$\exists x, y \text{ hat-geheiratet}(x, y)$	heißt	„Mindestens Einer hat mindestens Einen geheiratet“
$\forall x, y \text{ hat-geheiratet}(x, y)$	heißt	„Jeder hat Jeden geheiratet“

Die Kombinationen kann man erschöpfend fortführen, unabhängig davon, ob die Aussagen tatsächlich sinnvoll sind oder nicht. Wichtig ist, dass alle mit wahr oder falsch beantwortet werden können.

Durch Einbeziehung der Negation kann man den einen Quantor mit Hilfe des anderen ausdrücken und damit den gleichen Sachverhalt unterschiedlich ausgedrückt in eine Äquivalenzrelation bringen:

$$\exists x \neg \text{hat-geheiratet}(x, \text{Anke}) \leftrightarrow \neg \forall x \text{ hat-geheiratet}(x, \text{Anke}),$$

was wahlweise bedeutet „Es gibt mindestens Einen, der Anke nicht geheiratet hat“ bzw. „Nicht alle haben Anke geheiratet“ und von der Aussage her äquivalent ist. Man spricht in diesem Fall auch von Dualität der Operatoren.

Mit der Prädikatenlogik lassen sich unter Verwendung von Operatoren und Quantoren komplexe Sachverhalte ausdrücken, z. B. die Definition der Geschwisterbeziehung:

$$\forall x, y \text{ geschwister}(x, y) \leftrightarrow [\neg(x = y) \wedge \exists v, w \neg(v = w) \wedge \text{elternnteil}(v, x) \wedge \text{elternnteil}(w, x) \wedge \text{elternnteil}(v, y) \wedge \text{elternnteil}(w, y)]$$

Zusammenfassend gibt es folgende Regeln, wie in der Prädikatenlogik syntaktisch gültige Formeln erstellt werden dürfen:

- Eine Konstante oder eine Variable ist ein Term.
- Eine auf n Terme angewandte n -stellige Funktion ist wieder ein Term.
- Ein auf n Terme angewandtes n -stelliges Prädikat entspricht einer korrekten Formel.
- Eine logische Kombination von korrekten Formeln entspricht wiederum einer korrekten Formel, wobei alle logischen Junktoren der Aussagenlogik erlaubt sind.
- Eine korrekte Formel kann in eine neue Formel überführt werden, indem man einen Quantor und eine Variable voranstellt. Falls erforderlich, sollten Klammerungen verwendet werden, um den Geltungsbereich eines Quantors zu beschreiben.

Die Prädikatenlogik liefert ausreichend Sprachmittel zur Darstellung von Faktenwissen und Inferenzregeln, die Ableitungen erlauben. Trotzdem sollte an den aufgeführten Beispielen klar werden, dass es dieser Form der Wissensrepräsentation daran mangelt, das repräsentierte Wissen zu organisieren, was sich insbesondere bei großen Wissenssammlungen nachteilig auswirkt. Wir wollen uns daher im Folgenden einigen Formen der Wissensdarstellung widmen, die es uns erlauben, Wissen im kontextuellen oder sachlichen Zusammenhang darzustellen.

2.5 Semantische Netze

Semantische Netze gehen auf (Quillian 1966) zurück und definieren Wortbedeutungen. Die Repräsentationsform geht davon aus, dass die Bedeutung eines Objekts (Begriffs) in erster Linie durch seine assoziativen Beziehungen zu anderen Objekten (Begriffen) beschreibbar ist. Da solche Assoziationen nicht immer hierarchischen oder anderen, z. B. alphabetischen, Ordnungskriterien folgen, lassen sich diese manchmal besser mit Hilfe eines Netzwerkes darstellen. Dies wird an einem kleinen Experiment deutlich: Wenn wir anderen Menschen bestimmte Wörter vorgeben und diese dann bitten, uns spontan einen Begriff zu nennen, der ihnen dazu einfällt, dann führt dies oft zu assoziativen Antworten, die subjektiven Aspekten genügen. Beispielsweise führt das Wort „schwarz“ oft zu einer Antwort „weiß“, oder der Begriff „Nacht“ löst in vielen Fällen den Gedanken an den Begriff „Tag“ aus. Assoziationen können demnach ein Verhältnis von zwei völlig selbstständigen Objekten ausdrücken, die nichts miteinander zu tun haben müssen, aber unter bestimmten Rahmenbedingungen in eine lose Beziehung gebracht werden können.

2.5.1 Wissensdarstellung mit Graphen

Diesem Gedanken folgend abstrahieren Semantische Netze vorhandenes Wissen mit Hilfe eines markierten Graphen. Die Knoten repräsentieren Objekte, wie Themen, Ereignisse, Situationen, Individuen oder deren Eigenschaften. Die Kanten drücken assoziative Beziehungen zwischen Objekten (und deren Eigenschaften) aus. Assoziationen sind oft bidirektional, werden aber häufig nur in einer bestimmten Richtung (des Zugriffs) benötigt und implementiert. Falls der Zugriff nur auf ein bestimmtes der verbundenen Objekte erfolgen darf, spricht man von einer gerichteten Beziehung, entsprechend sind die Kanten im Graphen auch gerichtet. Somit wird jegliche Information, die notwendig ist, um ein Objekt zu repräsentieren, direkt über seine Beziehungen zugreifbar, d. h.: Wissensherleitung erfolgt durch Suche in dem Netzwerk von Beziehungen entlang von gerichteten Kanten.

Semantische Netze wurden ursprünglich eingeführt, um Sätze der natürlichen Sprache in eine formale Darstellung zu überführen. Um das Prinzip zu erläutern, wollen wir uns zunächst mit einfachen Sätzen beschäftigen, die aus dem Tripel Subjekt-Prädikat(Verb)-Objekt bestehen. Subjekt und Objekt stehen dann für Knoten und Prädikate bzw. Verben für Kantenbezeichnungen. Der Satz „Rudi mag Zimtgebäck“ lässt sich durch ein einfaches Semantisches Netz darstellen, das in Abbildung 2.1 zu sehen ist.

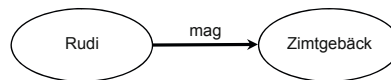


Abb. 2.1 Semantisches Netz für den Satz „Rudi mag Zimtgebäck“.

Aussagen sind jedoch nicht allein auf eine einfache Form eingeschränkt. Auch Sätze wie „Anke backt Zimtgebäck für Rudi“ lassen sich als Semantische Netze beschreiben, wie Abbildung 2.2 zeigt. In diesem Fall gehen wir anders vor und beschreiben ein Semantisches Netz für die stattfindende Aktion mit dem beteiligten Akteur und dem Empfänger des Objektes. Das Verb selbst wird in diesem Beispiel zum zentralen Knoten und beschreibt die verschiedenen Rollen mittels der Relationen.

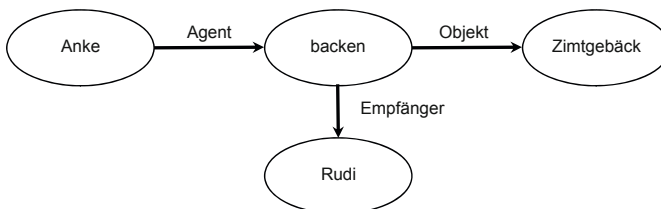


Abb. 2.2 Semantisches Netz für den Satz „Anke backt Zimtgebäck für Rudi“.

Wir werden uns in Kapitel 3 noch intensiver mit verschiedenen Formen von Semantischen Netzen beschäftigen, denn die Ausdrucksmöglichkeiten Graphen-basierter Wissensrepräsentation sind mannigfaltig. Im Folgenden werden wir uns auf die wichtigsten Eigenschaften Semantischer Netze konzentrieren und uns im nächsten Schritt mit

den (meist binären) Assoziationsbeziehungen, die zwischen Knoten bestehen können, auseinandersetzen.

2.5.2 Relationen und ihre Eigenschaften

In Semantischen Netzen steckt das Wissen in den Begriffen der Knoten und den Attributen der Kanten. Im Allgemeinen wird dabei versucht, Objekte und Eigenschaften mit Relationen in Zusammenhang zu bringen, welche den verschiedenen Formen einer Assoziation genügen. Bevor wir uns näher damit beschäftigen, möchten wir zunächst drei wesentliche Eigenschaften ansprechen, mit deren Hilfe man die verschiedenen Relationstypen auf einer mathematischen Grundlage charakterisieren kann. Diese Eigenschaften sind *Reflexivität*, *Symmetrie* und *Transitivität* und entsprechen folgenden Definitionen:

- Eine zweistellige Relation R heißt *reflexiv* für eine Menge von Objekten, wenn jedes Element der Menge in Relation zu sich selbst steht, also wenn $x R x$ für alle Elemente x der Menge gilt.
- Eine zweistellige Relation R heißt *symmetrisch* für eine Menge von Objekten, wenn aus $x R y$ stets $y R x$ folgt.
- Eine zweistellige Relation R heißt *transitiv* für eine Menge von Objekten, wenn aus $x R y$ und $y R z$ stets $x R z$ folgt.

Eine der wichtigsten Formen lexikalisch-semantischer Relationen ist die Vererbungsrelation, welche die Beziehung zwischen einem Unter- (*Hyponym*) und seinem Oberbegriff (*Hyperonym*) beschreibt. Diese transitive und asymmetrische Beziehung kann man am besten mit dem Ausdruck „ist-ein“ umschreiben. Wir werden uns später noch intensiver mit dieser Relation beschäftigen.

Eine andere Art von dekomponierenden Wissensabstraktionskonzepten ist die Aggregation. Sie drückt ein starkes semantisches Verhältnis von zwei an sich selbstständigen Objekten aus, von denen eines Teil des anderen ist, sehr typisch für Domänen, die physikalische Objekte oder Mengen (z. B. Bauteile oder Länder) betrachten. Daher spricht man hier auch von einer „ist-Teil-von“-Beziehung. Oft wird diese Wissensabstraktionsform auch als *Partonomie* oder *Meronymie* bezeichnet. Die dazu konverse Relation nennt man *Holonymie*, was einer „enthält“- oder „hat-Teil“-Beziehung gleichkommt. Während die Semantik der „ist-ein(e)“-Beziehung gut untersucht und umfangreiche Inferenzsysteme für deren Nutzung zur Verfügung stehen, ist die „ist-Teil-von“-Verknüpfung schwieriger zu handhaben und in ihrer Semantik deutlich domänenabhängiger.

Eine weitere Form der Verknüpfung betrifft die Bedeutungsgleichheit, die wir zwischen Begriffen in Form von Synonymen und Akronymen wiederfinden. Mit Hilfe dieser Relation kann man etwa ausdrücken, dass gilt:

DFKI „ist-gleich-zu“ Deutsches Forschungszentrum für Künstliche Intelligenz, oder
Handynummer „ist-gleich-zu“ Mobiltelefonnummer

Beide Beispiele machen deutlich, dass diese *Synonymie* eine Äquivalenzbeziehung darstellt, also reflexiv, symmetrisch und transitiv ist.

Auch die Fähigkeit des Ausdrückens von Gegensätzen, sogenannter Antonyme, ist eine wichtige Form der Assoziation. Ein bereits genanntes Beispiel ist:

schwarz „ist-gegensätzlich-zu“ weiß.

Die *Antonymie* ist zwar symmetrisch, jedoch genügt sie nicht den Eigenschaften der Reflexivität und Transitivität.

Über diese beiden Vergleichsrelationen hinaus wird oft noch eine kausale Abhängigkeit zwischen Objekten mit Hilfe von Assoziationen ausgedrückt, etwa Ereignisse, die gewisse Zustände oder andere Ereignisse nach sich ziehen. Man nennt eine solche Relation auch *Kausation*. Kausale Relationen werden oft mit der Beziehung „folgt-aus“ dargestellt. Beziehungen zwischen Ursache und Wirkung sind offensichtlich transitiv, aber nicht symmetrisch, da eine feste zeitliche Abhängigkeit existiert.

Kausale Verbindungen gehören zu den sogenannten *Common Sense*-Verknüpfungen, zu denen auch temporale oder lokale Relationen gehören. Diese werden in speziellen wissensbasierten Anwendungen eingesetzt. So werden temporale Relationen oft in Planungssystemen, lokale Relationen in Konfigurationsaufgaben und kausale Relationen bei modellbasierten Diagnosesystemen angewendet. Darüber hinaus werden sie manchmal noch zur Darstellung von Dokumentinhalten beim Information Retrieval herangezogen.

Wie bereits erwähnt, werden wir uns in Kapitel 3 ausgiebig mit Semantischen Netzen auseinandersetzen und verschiedene Variationen und Weiterentwicklungen davon vorstellen und diskutieren. In Abschnitt 3.1.2 werden in Verbindung mit der Klassifikation Semantischer Netze auch nochmals die verschiedenen Relationsarten betrachtet und kurz diskutiert. Eine systematische Aufarbeitung der verschiedenen Verbindungstypen für solche Frame-Systeme findet sich beispielsweise in der *frame ontology*, die als Erweiterung im Knowledge Interchange Format (KIF) (Genesereth und Fikes 1992) implementiert ist und in der 18 Typen von Relationen definiert sind.

2.5.3 Eigenschaften Semantischer Netze

Zurück zu unserem Beispiel: Nehmen wir an, dass Anke, Rudis Frau, in der Vorweihnachtszeit Zimtgebäck gebacken hat. Dieses Zimtgebäck besteht aus gewissen Zutaten, die wiederum in Form bestimmter Mengen zum Gelingen beigetragen haben. Alle Aspekte, welche dieses spezielle Zimtgebäck ausmachen, kann man mit Hilfe eines Semantischen Netzes repräsentieren. Abbildung 2.3 zeigt die graphische Darstellung dieses Netzes, das unterschiedlichste Typen von Verbindungen nutzt um Ankes Zimtgebäck zu beschreiben.

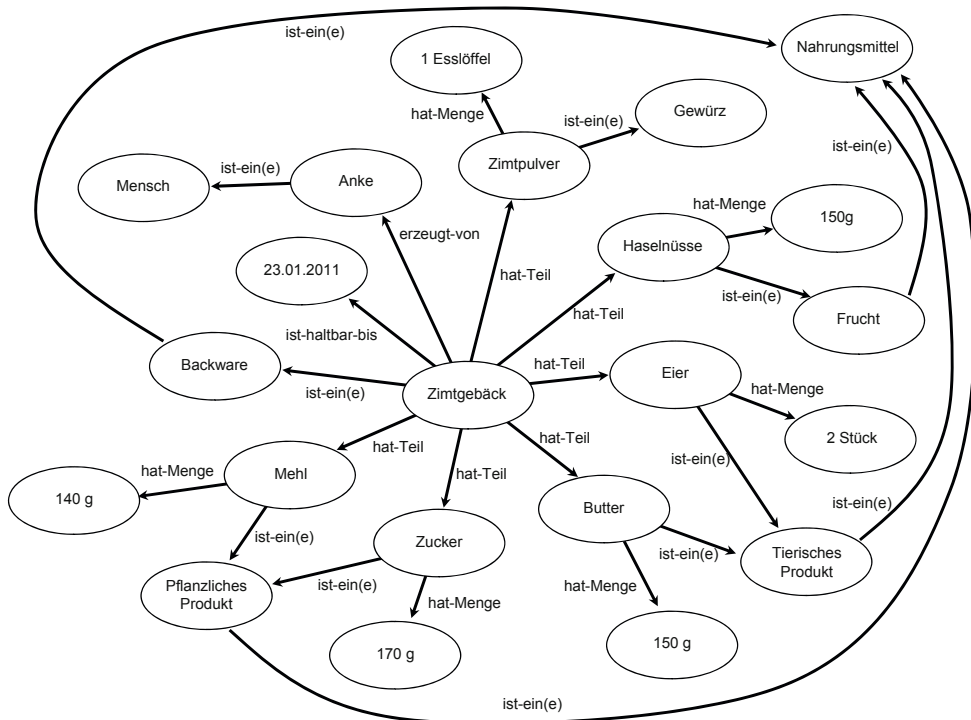


Abb. 2.3 Semantisches Netz zur Beschreibung von Zimtgebäck.

Vergleicht man die Darstellung in Abbildung 2.3 mit der logischen Form der Wissensdarstellung, dann scheint eine graphische Repräsentation intuitiver, da sie sich an der natürlichen Sprache orientiert. Dies ist gut nachvollziehbar, wenn man vom Knoten mit dem Begriff „Zimtgebäck“ aus den Relationen entlang der verschiedenen Verästelungen folgt. Auch scheint es „wirtschaftlich“ zu sein, Semantische Netze zu verwenden, denn ein Objekt oder eine Eigenschaft müssen jeweils nur bei ihrem ersten Auftreten einmal als Knoten benannt werden, wodurch Speicherplatz gespart wird. Weitere Aussagen zu Begriffen werden einfach über neue Relationen im Graphen ergänzt.

Man kann an dem Beispiel in Abbildung 2.3 aber auch sehr gut erkennen, wie schnell Semantische Netze wachsen. Obwohl das illustrierte Beispiel eigentlich keinen komplizierten Sachverhalt darstellt, benötigen wir bereits viele Knoten und Relationen und haben längst nicht alle Details modelliert. Die semantische Repräsentation richtig komplexer Sachverhalte kann also bei der Modellierung einer Alltagsdomäne aufwendig und unübersichtlich werden, da die verschiedenen Aspekte des Wissens unter Umständen sehr zersplittert im Semantischen Netz vorliegen. Man kann sich demnach gut vorstellen, dass die Herleitung von Wissen einen aufwändigen Suchprozess auslösen kann. Die Suche in semantischen Netzen verwendet einen sogenannten Spreading-Activation-Mechanismus in Form einer *Breitensuche* (siehe <http://de.wikipedia.org/wiki/Prädikatenlogik>). Ob das Semantische Netz in sich konsistent ist, lässt sich dabei nicht prüfen.

2.5.4 Kardinalität von Relationen

Betrachten wir nochmals das Beispiel in Abbildung 2.3. Wie das Semantische Netz veranschaulicht, kann ein Objekt durch die gleiche Relation mit mehreren anderen benachbarten Objekten in Beziehung stehen. So sind die verschiedenen Zutaten von Zimtgebäck alle mit der gleichen „hat-Teil“-Relation beschrieben. Die Nutzbarkeit einer Beziehung für ein Objekt eines bestimmten Objekttyps wird durch deren *Kardinalität* definiert. Die *Kardinalität* eines Beziehungstyps in einem Entity-Relationship-Diagramm (ERD) legt je nach Notationsform entweder fest, an wie vielen Beziehungen eine Entität teilnehmen kann (Min-Max-Notation) oder wie viele Entitäten mit einer gegebenen Entität oder Entitätskombination höchstens in Beziehung stehen können. In Semantischen Netzen legt die Kardinalität fest, wie oft ein gegebenes Objekt eines Objekttyps (Objekttypen können von ihrer Bedeutung her etwa mit den bisher behandelten Klassen gleich gesetzt werden.) unter Verwendung eines bestimmten Relationstyps mit Objekten eines anderen Objekttyps verbunden sein darf. In Bezug auf einen bestimmten Relationstyp unterscheidet man dabei folgende Fälle:

- Ein Objekt eines Objekttyps steht mit **höchstens einem** Objekt eines anderen Objekttyps in Beziehung (Kardinalität 1:1).
- Ein Objekt eines Objekttyps steht mit **beliebig vielen** Objekten eines anderen Objekttyps in Beziehung. In der Gegenrichtung steht ein Objekt des einen Objekttyps mit **höchstens einem** Objekt des anderen Objekttyps in Beziehung (Kardinalität 1:n).
- Ein Objekt eines Objekttyps steht mit **beliebig vielen** Objekten eines anderen Objekttyps in Beziehung (Kardinalität n:m).

Es gibt also Relationen, die ein Objekt mit vielen anderen Objekten verbinden, wie z. B. die „hat-Teil“-Relation, oder aber nur einmalig pro Objekt verwendet werden kann, wie dies bei einer Mengenangabe der Fall ist. Die Verwendung solcher Relationen beschreibt die Semantik eines Objektes, wobei die Interpretation, was Objekt und was Eigenschaft ist, nur durch nähere Betrachtung der verwendeten Begriffe möglich ist.

Abbildung 2.4 (a) zeigt die abstrakte Darstellung eines Semantischen Netzes, dessen Knoten bewusst keine Bezeichner enthalten, denn Semantische Netze machen keinen Unterschied, ob der Begriff im Knoten ein Objekt oder seine Eigenschaft repräsentiert. Um das Problem deutlicher zu machen, haben wir Objekte und deren Eigenschaften im Beispiel mit unterschiedlich umschreibenden Linien versehen. Wäre diese graphische Unterscheidbarkeit nicht gegeben, so könnte der Betrachter nicht differenzieren, welche Semantik ein Knoten hat, also ob es sich um ein Objekt oder um eine Eigenschaft handelt.

Aber nicht nur der Zugriff auf die Semantik eines Knotens wäre wünschenswert, denn bei Betrachtung des Beispiels liegt nahe, Objekte mit all ihren zugehörigen beschreibenden Eigenschaften als semantische Einheiten zusammenzufassen, wie dies in Abbildung 2.4 (b) gemacht wurde. Darüber hinaus könnten auch gemeinsame Beschreibungselemente (grau hinterlegte Bereiche) festgestellt werden, um daraus allgemeiner geltende

Eigenschaften zu generieren, die verallgemeinernde Objekte besitzen. Wir werden uns mit diesem Gedanken intensiver im nächsten Abschnitt befassen, denn er ist besonders reizvoll.

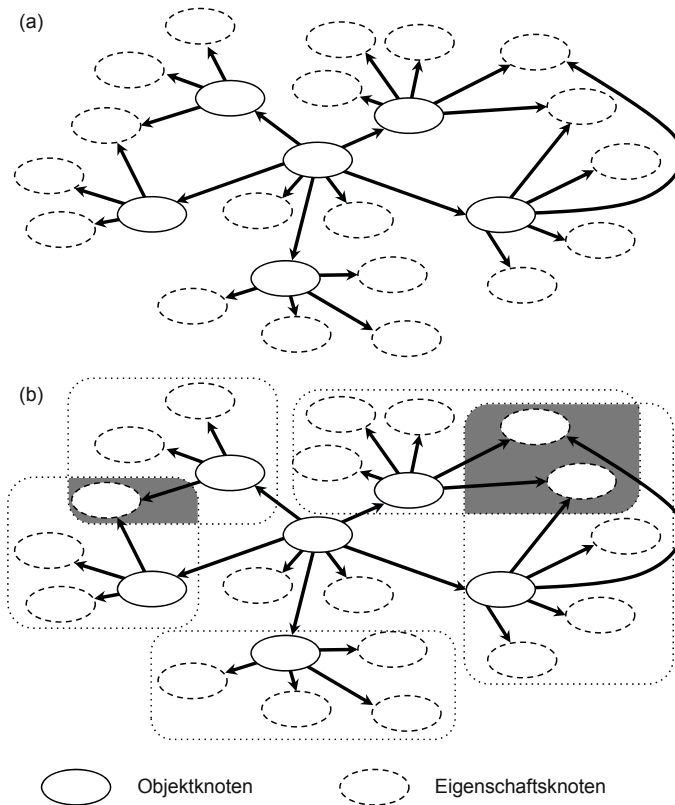


Abb. 2.4 Zusammenfassung von Objekten und Eigenschaften in Semantischen Netzen. (a) Ohne Bildung semantischer Einheiten und (b) mit Bildung semantischer Einheiten durch Zusammenfassung von Objekten und Eigenschaften.

2.6 Frames

2.6.1 Der Schemabegriff

Wie wir in Abschnitt 2.4 bereits aufgezeigt haben, sind Prädikate wesentliche Elemente zur Beschreibung von Eigenschaften. Mit ihrer Hilfe lassen sich zeitliche und räumliche Daten quantitativ sowie ideelle Werte, äußere Merkmale oder wesentliche mentale Eindrücke qualitativ ausdrücken. Prädikate stehen dabei immer im Zusammenhang mit sprachlich-logischen Feststellungen über Entitäten der realen Welt, die wir über unsere Sinnesorgane wahrnehmen. Prädikate stehen für das, was über ein Objekt zu sagen ist,

also seine Eigenschaft. Das damit in Verbindung stehende Wissen lässt sich in einem sogenannten Schema zusammenfassen. Der Begriff *Schema* wurde ursprünglich von (Bartlett 1932) eingeführt und bezeichnet das Wissen von Menschen, etwa über ein Individuum, ein Objekt, ein Ereignis oder eine Handlung.

Ein Schema stellt somit eine kognitive Struktur dar, welche in sich selbstständig ist (z. B. (Rumelhart 1980, Schank und Abelson 1977)), und damit die Verarbeitung seiner Informationen bestimmt. Das heißt, dass ein Schema in seiner Beschreibung abgrenzbar ist und als Einheit funktioniert, was wiederum bedeutet, dass seine es definierenden Bestandteile zur gleichen Zeit aktiviert werden. Schemata lassen sich durch Reize von außen oder über Beziehungen zu anderen Schemata von innen aktivieren.

Mandel und Spada (1988) unterscheiden sechs charakteristische Merkmale für Schemata:

- 1) Schemata sind kognitive Strukturen, in denen allgemeines Wissen im Gedächtnis repräsentiert ist. Das Wissen über typische Zusammenhänge in einem Realitätsbereich ist in Schemata organisiert.
- 2) Schemata weisen Variablen auf, die unterschiedliche Werte annehmen können.
- 3) Schemata können ineinander eingebettet sein.
- 4) Schemata enthalten sowohl episodisches als auch generisches Wissen.
- 5) Schemata haben sowohl eine Struktur-, als auch eine Prozesskomponente.
- 6) Schemata repräsentieren Wissen unterschiedlichster Inhaltsbereiche.

Mit einem Schema werden die mentalen Vorstellungen, wie wir sie im ersten Kapitel im aristotelischen Sinne diskutiert haben, mit einer Realdefinition einer symbolischen Beschreibung verknüpft. Ein bestimmter Gegenstand lässt sich damit einer Kategorie oder Gattung zuweisen, welche diese definiert.

Wenn wir uns mit einem Freund oder einer Kollegin über etwas unterhalten, dann neigen wir dazu, uns das ausgesprochene Etwas, sei es ein Gegenstand, ein Individuum oder eine bestimmte Situation, vorzustellen. Wenn wir diese Imaginationen beschreiben wollen, so greifen wir meist auf qualitative und quantitative Prädikate zurück, welche die wesentlichen Imaginationen sprachlich-logisch wiedergeben und die in einer Schemabeschreibung zusammengefasst werden können. Betrachten wir uns dazu folgendes Szenario:

„Als Rudi den Schlüssel in die Haustür steckt, wird er bereits stürmisch von Cindy und Bert empfangen. Nachdem er seinen Mantel abgelegt hat, zerren ihn die Kinder ungeduldig ins Wohnzimmer. Dort brennen bereits die Kerzen, Weihnachtslieder erfüllen dem Raum und es duftet nach Zimtgebäck.“

Beim Durchlesen dieser Kurzgeschichte fällt es uns sicherlich nicht schwer, uns die Situation vor Augen zu führen. Die spärlichen Aussagen im Text wirken dabei wie

ein Schlüssel, der eine Tür zu unseren mentalen Modellen aufschließt und uns das beschriebene Szenario noch viel umfangreicher, gegenständlicher und detailreicher vergegenwärtigt.

Wir stellen uns wahrscheinlich vor, dass die Szene am Abend spielt, möglicherweise an Heiligabend. Wir können darüber hinaus vermuten, dass Rudi von seinen Kindern empfangen wird und die Kinder aufgrund ihres Verhaltens ein bestimmtes Alter haben müssen. Es ist weiter anzunehmen, dass die Kinder nicht allein zu Hause sind, sondern ihre Mutter, Rudis Frau Anke, die Musik eingeschaltet und die Kerzen angezündet haben muss. Wir haben womöglich eine facettenreiche detaillierte Vorstellung von der gesamten Szene im Kopf.

2.6.2 Frames als Denkmodell

Dem Leitgedanken von Aristoteles „Was ist da und was kann ich über das sagen, was da ist“ folgend, führte Marvin Minsky 1974 in seinem einflussreichen Aufsatz „A framework for representing knowledge“ den Begriff *Frame* als Grundlage für ein intuitives Denkmodell in die Künstliche Intelligenz ein. Frames sind nichts anderes als Schemata, die in ihrem Zusammenspiel ein komplexes Netzwerk mentaler Konzepte repräsentieren. Sie simulieren – wie in Abbildung 2.5 nachgestellt – somit Denkprozesse. Die Knoten des Netzwerks repräsentieren die Schemata und die Kanten die Wechselwirkungen zwischen diesen.



Abb. 2.5 Ein Framesystem als intuitives Denkmodell.

Jedes Frame bezeichnet ein mentales Konzept und beschreibt es durch Einbeziehung visueller oder auditiver Sinneseindrücke, die es von anderen Konzepten differenzieren. So könnte unser Beispiel in ähnlicher Weise bereits von uns selbst erlebt worden oder Teil einer Szene sein, die wir vielleicht schon einmal in einem Film gesehen haben. Dies kann dazu führen, dass wir uns Dinge gedanklich ausmalen können, die gar nicht in den Aussagen unserer Kurzgeschichte enthalten sind, etwa die Vorstellung des Raumes, der

Garderobe, an die Rudi seinen Mantel hängt, das heimelige Licht des Wohnzimmers, das Zimtgebäck in einer Schale auf einem Tisch usw.

Frames sind uniforme Objekte oder Konzeptrepräsentationen, welche die verschiedenen Ausprägungen ihrer begrifflichen Bedeutung beschreiben. Sie wirken dabei wie Filter, die wesentliche Sinneseindrücke mit dem mentalen Kontext, in dem Objekte stehen, verbinden. So könnte man sich vorstellen, dass man Rudi auf Grundlage von beispielsweise Vorname, Nachname, Geburtsdatum, Geschlecht, Größe und Gewicht sowie Haarfarbe beschreibt. Das entsprechende Objekt in Frame-Darstellung ist in Abbildung 2.6 dargestellt.

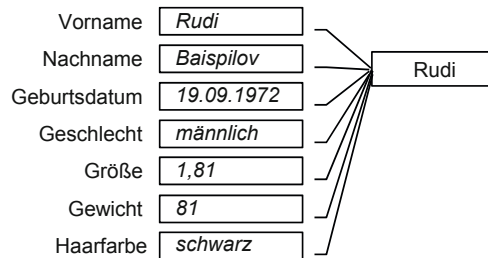


Abb. 2.6 Beispielhafte Beschreibung des Objektes „Rudi“ als schematisiertes Frame.

Jedes Frame hat zunächst einen Bezeichner, der das dahinterstehende Konzept beschreibt. In vielen Fällen verwendet man dazu einen eindeutigen Namen, wie einen Identifikator (ID) oder einen regulären Ausdruck. Bei Semantischen Technologien, die im Internet ihre Anwendung finden, werden dazu gerne URIs verwendet. Die sogenannten Uniform Resource Identifier (URI) folgen einer bestimmten, weltweit genutzten Standard-Syntax und sind eindeutig, was den Vorteil hat, dass damit jede betrachtete Ressource auch eindeutig ansprechbar ist. Wir kennen diese URI von unserem Computer, wo wir Dateien auf diese Weise benennen (z. B. file://Dokumente/Buchmanuskript/Kapitel2) oder von der Adresse, über die wir eine Website aufrufen (z. B. <http://de.wikipedia.org/Frames>).

In unserem Fall halten wir uns der Einfachheit wegen an die gewählten Objektbezeichnungen unseres Diskursbereiches. Die Eigenschaften sind mit einem Prädikat oder Attribut in Fächern beschrieben und können einen Wert besitzen.

Als Minsky Frame-Systeme einführte, war er war überzeugt, dass das vorgeschlagene Modell dem menschlichen Denken (wie wir es aus der Kognition kennen) bereits sehr nahe kommt. Er sah dies insbesondere darin begründet, weil Frames es erlauben, Objekte eines Problemereichs ähnlich der realen Welt auf Grundlage von Eigenschaften oder Prädikaten zu gliedern oder zu abstrahieren sowie ihre Wechselwirkungen zu repräsentieren.

Frames können aber nicht nur Objekte systematisch beschreiben. Auch zeitliche Aspekte lassen sich strukturell repräsentieren, wie der Ablauf einer Weihnachtsfeier oder die Reihenfolge der Aktionen beim Backen von Zimtgebäck oder der Besuch eines Restaurants (vgl. Abschnitt 2.8).

Frames sind also nichts anderes als modulare Beschreibungen deklarativer und prozeduraler Aspekte unserer Wahrnehmung, die sich im kognitiven Zusammenhang abrufen lassen. Es ist offensichtlich, dass mit dem Ansatz von Minsky zwei wesentliche Aspekte der Wissensrepräsentation abgedeckt werden:

- Frames sind abstrakte Informationseinheiten, mit deren Hilfe sich stereotype Wissen strukturell beschreiben lässt.
- Durch Vernetzung mit anderen Frames können Zusammenhänge und Wechselwirkungen und damit komplexe Sinneseindrücke dargestellt werden.

Wir haben Rudi am Anfang dieses Kapitels kurz vorgestellt, wir können ein Zimtgebäck beschreiben, wir wissen, dass Heiligabend ein Ereignis ist usw., aber alle diese Aspekte erlangen ihre individuelle Bedeutungsfülle erst im inhaltlichen Zusammenhang der beschriebenen Szene.

Auffallend ist, dass es scheinbar Konzepte mit unterschiedlichem Abstraktionsgrad gibt. Dies wird deutlich, wenn wir uns bewusst werden, dass nicht nur Rudi der Beschreibung in Abbildung 2.6 folgt, sondern Anke nach dem gleichen Muster beschrieben werden kann. Gleiches gilt für andere Personen in unserer Beispieldomäne, wie Gesine, Cindy oder Bert.

2.6.3 Klassifikation

Folglich wäre es sinnvoll, alle Personen in einem Frame zusammenzufassen, welches die gemeinsamen Eigenschaften der beschriebenen Konzepte als Schablone zur Verfügung stellt. Solche Musterframes nennt man *Klassen*. Eine Klasse ist also ein „wertloses“ Frame, das für eine Menge individueller Objekte angibt, welche Eigenschaften diese teilen. Mit dem Konzept „Mensch“ können entsprechend alle Personen, egal ob männlich oder weiblich, ob jung oder alt, zusammengefasst werden. Sollte zwischen männlichen oder weiblichen Personen weiter unterschieden werden (müssen), kann man, statt eines einzelnen Faches über das Geschlecht, die Konzepte „Mann“ und „Frau“ einführen. Die Zusammenfassung von gleichartigen Frames zu einer Klasse nennt man *Klassifikation*.

Abbildung 2.7 beschreibt die Klasse Mensch mit den bereits für das Objekt „Rudi“ verwendeten Eigenschaften, die in Fächern angeordnet sind. Alle individuellen Ausprägungen dieser Klasse, also in unserer Domäne die bisher bekannten Menschen, werden so klassifiziert und besitzen ebenfalls diese Eigenschaften. Die Fächer bei den einzelnen Personen sind selbstverständlich – wie bei Rudi – entsprechend alle mit konkreten Werten gefüllt.

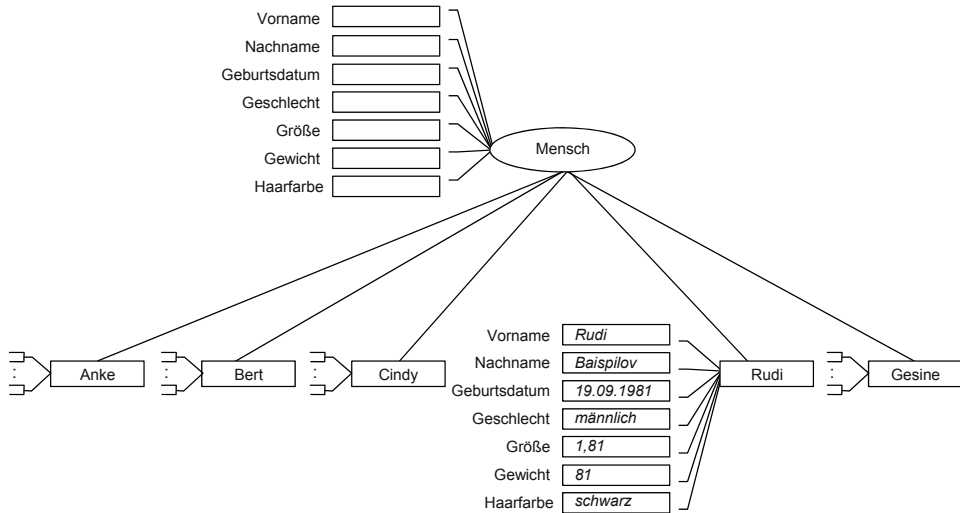


Abb. 2.7 Klassifikation von Frames.

2.6.4 Generalisierung und Spezialisierung

Klassen in Framesystemen wirken wie Erfahrungswissen, wie wir es bei der Interpretation der Kurzgeschichte anwenden, um Eigenschaften abzuleiten und Zusammenhänge zu erkennen. Dies gilt nicht nur für die beteiligten Personen. Dazu wollen wir unser Beispiel um eine weitere Beschreibung erweitern:

„Zimtgebäck bezeichnet ein Objekt, das ein Nahrungsmittel darstellt. Genauer gesagt gehört es zu Backwaren, die von Menschen hergestellt und begrenzt haltbar sind. Wann immer es dem Prozess des Essens unterzogen wird, existiert es nicht mehr. Das Zimtgebäck, das an Heiligabend bei Rudi im Wohnzimmer steht, hat Anke tags zuvor gebacken. Sie verwendete dabei Zutaten wie ...“

Beispielsweise wissen wir grundsätzlich, dass Zimtgebäck etwas zum Essen ist und wie alle anderen Formen von Gebäck zunächst gebacken werden muss, um als Gebäck zu gelten. Dieser Kenntnis zugrunde liegen Erfahrungen, die wir in der Vergangenheit gemacht haben und die es uns erlauben, zu abstrahieren bzw. zu generalisieren und das gelernte, verallgemeinerte Wissen auf vorliegende Daten anzuwenden. Dabei entstehen Verkettungen von Unter- und Oberbegriffen, wie:

Der Begriff „Nahrungsmittel“ ist eine Verallgemeinerung für „Backwaren“ was wiederum eine umfassendere Bezeichnung für „Zimtgebäck“ darstellt.

Der Bezeichner „Nahrungsmittel“ ist also ein Oberbegriff (*Hyperonym*) von „Backwaren“ und „Zimtgebäck“ ist ein Unterbegriff (*Hyponym*) davon. Die schrittweise Abstraktion der Begriffsbeschreibung führt zu einer Hierarchie von Klassen.

In Framesystemen gibt es verschiedene Vorgehensweisen, solche Klassenhierarchien aufzubauen. Man kann einerseits vom Speziellen zum Allgemeinen gehen. Den Prozess nennt man Generalisierung. Objekte werden verschieden klassifiziert, die resultierenden Klassen werden sukzessive abstrahiert und während des Verallgemeinerungsprozesses in einer baumartigen Hierarchie zusammengefasst. Klassen besitzen demnach eine Oberklasse, wenn eine Verallgemeinerung für sie existiert. Mit einer Generalisierung verbunden ist die Extraktion von gemeinsamen Eigenschaften und deren Verlagerung in die Oberklasse.

Den umgekehrten Prozess nennt man Spezialisierung. Er beinhaltet die Verfeinerung der Klassen durch Verwendung von Unterbegriffen und Ergänzung der resultierenden Unterklasse mit weiteren, unterscheidenden Eigenschaften. Eine Klasse hat demnach eine Unterklasse, wenn es eine Spezialisierung davon gibt.

Generalisierung findet oft dann statt, wenn eine begrenzte Menge realer Objekte vorliegt, beispielsweise bei Klassifikationsproblemen. Die Objekte werden zunächst untersucht und aufgrund differenzierender Merkmale in Klassen getrennt. Die entstehenden Klassen werden entsprechend genauso behandelt, so dass nach und nach – wie in Abbildung 2.8 (a) – ein Klassenbaum entsteht. Spezialisierung ist ein typisches Instrument zur Top-Down-Beschreibung von Domänen, bei der Klassen bereits initial feststehen und sich systematisieren lassen, um die Objekte der Domäne aufgrund gemeinsamer Merkmale zu charakterisieren (vgl. Abbildung 2.8 (b)).

Falls zu einem späteren Zeitpunkt neue Objekte hinzu genommen werden müssen, so kann dies die bestehende Klassenhierarchie beeinflussen. Wie in Abbildung 2.8 (c) dargestellt, müssen eventuell neue Klassen eingeführt und in die Hierarchie eingebunden werden. Das Einbinden kann unter Umständen auch dazu führen, dass Veränderungen in der Fächerstruktur der bestehenden Klassen vorgenommen, d. h. neue Fächer eingeführt oder bestehende Fächer verfeinert oder überschrieben werden müssen (siehe auch Überdeckungsbeziehung, Abschnitt 2.7.1).

Frames sind also schematische Wissensstrukturen mit unterschiedlichem Abstraktionsgrad. Man kann davon ausgehen, dass je allgemeiner ein Objekt ist, das beschrieben wird, desto unspezifizierter bleiben dabei seine Eigenschaften. Je spezieller man jedoch Objekte betrachtet, umso mehr werden sie mit Daten angereichert.

2.6.5 Vererbung

Manche Eigenschaften, beispielsweise dass Backwaren und Zimtgebäck nach einer gewissen Zeit ungenießbar werden, müssen – wie bereits angedeutet – daher nicht in jeder betreffenden Klasse explizit definiert werden. Es genügt zu wissen, dass alle Nahrungsmittel ein Haltbarkeitsdatum haben. Andere Eigenschaften, wie etwa den „Bäcker“ des köstlichen Zimtgebäcks auf Rudis Wohnzimmertisch können erst dann beschrieben werden, wenn das entsprechende Objekt erzeugt wurde.

An diesem Beispiel wird deutlich, dass sich gewisse Objekte auf Grundlage bestimmter Begriffe zu Klassen, Konzepten oder Schemata zusammenfassen lassen, die allgemeine

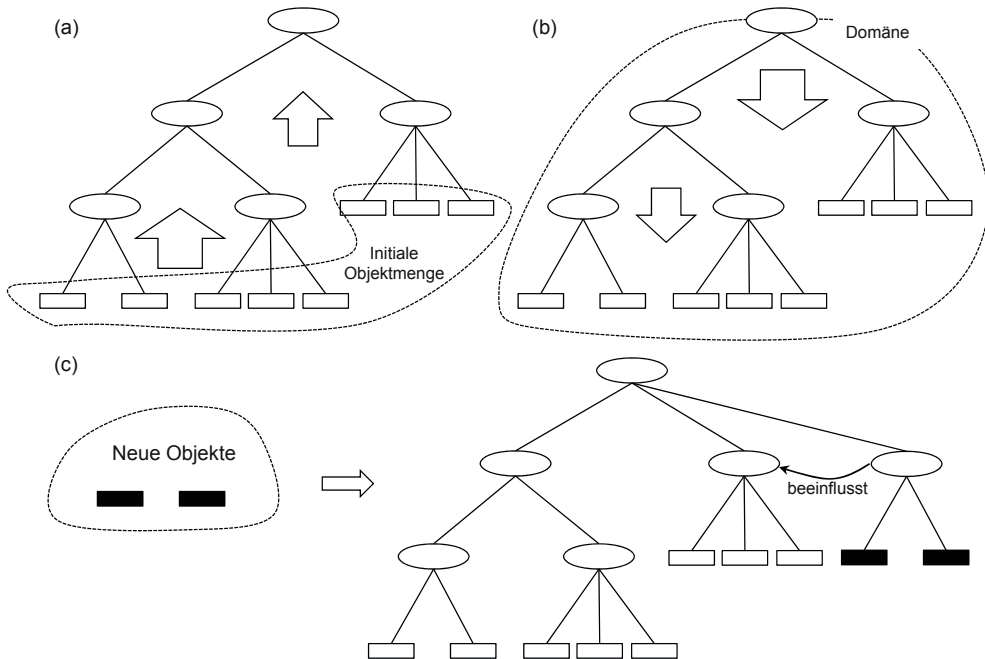


Abb. 2.8 Aufbau und Veränderung von Klassenhierarchien, (a) Generalisierung, (b) Spezialisierung und (c) nachträgliche Aufnahme neuer Objekte.

Beschreibungen oder Muster ihrer Exemplare oder Ausprägungen darstellen. Es reicht dabei aus, alle dieser Klasse, dieses Konzeptes oder diesem Schema innewohnenden Eigenschaften nur einmal, möglichst an der Stelle zu beschreiben, wo sie als allgemeingültig für alle daraus abgeleiteten Spezialisierungen gelten; in unserem Beispiel also etwa die Eigenschaft „Haltbarkeitsdatum“ bei der Klasse „Nahrungsmittel“.

Da es verschiedenste Nahrungsmittel gibt, lassen sich wiederum speziellere Klassen daraus ableiten. Solche Klassen, wie „Fleisch“, „tierische Produkte“, „pflanzliche Produkte“, „Gewürze“, „Getreide“, „Obst“, „Gemüse“ oder aber „Backwaren“ erben diese Eigenschaft sozusagen von ihrer jeweiligen Oberklasse, ohne weiteres Zutun. Eine Vererbung ist damit eine Strukturbeziehung zwischen Klassen, die konstitutiv für alle abgeleiteten Exemplare ist.

In den entstehenden neuen Klassen können dann weitere Fächer mit spezielleren Eigenschaften, wie z. B. „Zutaten“ und „Hersteller“ bei „Backwaren“ enthalten sein, welche wiederum an die Unterklasse „Zimtgebäck“ vererbt werden. Eine Unterklasse ist also vollständig konsistent mit ihrer Oberklasse, erweitert diese jedoch um zusätzliche Eigenschaften. Da eine Unterklasse wiederum Oberklasse von weiteren Unterklassen sein kann, sind mehrere Stufen der Vererbung möglich.

Demnach besitzt die Klasse „Zimtgebäck“ Eigenschaften von „Backwaren“ aber auch solche, die allen Nahrungsmitteln gemein sind. Zwischen Ober- und Unterklassen

besteht eine Hyperonym-Hyponym-Beziehung, die – wie bereits bei den Semantischen Netzen eingeführt – zu den beiden zusammengefassten Aussagen

Zimtgebäck „ist-ein(e)“ Backware „ist-ein(e)“ Nahrungsmittel

führt, nur mit dem Unterschied, dass die Begriffe Zimtgebäck, Backware und Nahrungsmittel auch strukturierten Konzepten oder Objekten mit Eigenschaften entsprechen. Man nennt solche Klassenhierarchien auch Taxonomien.

Der Begriff Taxonomie leitet sich von den griechischen Wörtern *táxis* und *nomos* ab und bedeutet so viel wie Ordnungsgesetz. Es ordnet Klassen, zwischen denen die Beziehung „ist-ein(e)“ eine Oberbegriff–Unterbegriff–Relation definiert. Taxonomien wurden originär in der Biologie verwendet, bei der die Einteilung traditionell in einen bestimmten Rang, wie *Art*, *Gattung* oder *Familie* erfolgt. Der Begriff hat sich aber in den letzten Jahrzehnten auch zunehmend in den Bibliotheks- und Sprachwissenschaften sowie bei den semantischen Technologien etabliert, in denen er zur Begriffsklassifikationen verwendet wird. Auch andere Fachbereiche verwenden den Begriff der Taxonomie allgemein für ein *Klassifikationssystem*, eine *Systematik* oder den Vorgang des *Klassifizierens*. Taxonomien sind ein intuitiv nachvollziehbares Abstraktionskonzept und aus unserer täglichen Arbeit bestens bekannt, z. B. findet man das gleiche Prinzip auch bei der Ordnermetapher im Dateisystem moderner Betriebssysteme wieder. Wir werden uns im nächsten Abschnitt noch intensiver mit Taxonomien beschäftigen.

2.6.6 Instanziierung

Natürlich ist das Zimtgebäck von Rudis Frau Anke etwas ganz Besonderes, nicht nur für Rudi, auch in der Betrachtung von Framesystemen. Zunächst ist es natürlich auch ein Zimtgebäck und könnte demnach der gleichen Beziehung „ist-ein(e)“ zu der es beschreibenden Klasse genügen. Es gibt jedoch einen wesentlichen Unterschied zu den bisher betrachteten Klassen, die eher den Charakter eines allgemeinen Musters oder einer strukturierten Schablone entsprechen. Ankes Zimtgebäck existiert und ist damit real. Es entsteht und ist da vom Zeitpunkt an, wo es aus dem Backofen genommen wird und existiert damit, hat also eine Lebensdauer, im Allgemeinen, bis es aufgegessen wurde. Solange gilt es als spezielle Ausprägung von Zimtgebäck. In der Fachsprache spricht man auch oft von einer Instanz einer Klasse. Daher gilt auch die Beziehung:

Ankes Zimtgebäck „ist-Instanz-von“ Zimtgebäck

Diese Aussage ist für die gesamte Lebensdauer von Ankes Zimtgebäck gültig oder wahr. Zudem hat das Objekt „Ankes Zimtgebäck“ mit dem Moment seiner Fertigstellung oder Erzeugung einen individuellen Hersteller (Anke), ein Haltbarkeitsdatum (23. Januar) und gewisse Zutaten (Butter, Eier, Haselnüsse, Mehl, Zucker, Zimtpulver, ...), die wir wie in Abbildung 2.6, wo wir Rudi beschreiben, in die jeweiligen Fächer eintragen.

Wie man gut erkennen kann, sind Teile einer Instanz *existenzabhängig*. Die Herstellung von Zimtgebäck erzeugt auch seine Bestandteile. Die Löschung einer Instanz löscht auch deren Komponenten. Wenn also das Zimtgebäck aufgegessen wurde, dann auch seine Zutaten. Man spricht in solchen Fällen auch von echter Aggregation oder Komposition.

Man kann an diesem Beispiel sehr schön erkennen, wie ein solches Netzwerk entsteht, da nicht nur in diesem Fall die deklarativen Einträge in den Fächern, also Butter, Eier, Haselnüsse, Mehl, Zucker, Zimtpulver usw., selbst wieder Beschreibungen für Klassen oder Instanzen sein können. So könnten beispielsweise weitere Beziehungsketten wie die folgenden gelten:

Eier „ist-ein(e)“ tierisches Produkt „ist-ein(e)“ Nahrungsmittel,
 Zimt „ist-ein(e)“ Gewürz „ist-ein(e)“ Nahrungsmittel, oder
 Anke „ist-Instanz-von“ Mensch „ist-ein(e)“ Lebewesen.

Wenn Instanzen erzeugt werden, lassen sich ihre Eigenschaftsfächer nicht immer ohne Weiteres mit Werten belegen. In unserem Beispiel ergeben sich die Werte für Hersteller und Zutaten von „Ankes Zimtgebäck“ entweder aus der Beobachtung, aus Fragen an Anke, welche Zutaten sie verwendet hat, oder indem wir die Angaben einfach aus Ankes Rezeptsammlung entnehmen. Für die Bestimmung der Haltbarkeitsdauer ist die Situation aber nicht ganz so einfach. Diese lässt sich, wenn überhaupt, nur aufgrund von Erfahrungswerten berechnen. Im Beispiel enthält das Fach „Haltbarkeitsdatum“ im einfachsten Fall also eine Operation, die z. B. zum Fertigstellungsdatum vier Wochen dazu addiert und das berechnete Datum dann in das entsprechende Fach der Instanz „Ankes Zimtgebäck“ einträgt.

Natürlich können solche Operationen beliebig komplex sein. So ist ein Haltbarkeitsdatum von vielen Aspekten gleichzeitig abhängig, etwa den Zutaten und deren Haltbarkeitseigenschaften oder von Umwelteinflüssen, wie Temperatur, Feuchtigkeit und Lichteinflüssen. Man kann sich also sehr komplizierte Operationen vorstellen, deren Daten vom Menschen erst eingegeben werden müssen, aus Datenbanken, Prozessbeschreibungen oder von Sensoren stammen, wobei bei letzteren vielleicht die Werte in den Fächern der Instanz nicht nur einmal, sondern periodisch aktualisiert werden müssen.

2.7 Taxonomien

Viele Semantische Technologien setzen auf dem Schema-Gedanken zur Wissensrepräsentation auf. Am Anfang steht dabei oft eine Taxonomie, auf Grundlage derer sich die relevanten Objekte in einer Ober- und Unterklassen-Beziehung beschreiben lassen. Bei der Interpretation einer Taxonomie wird ein geschlossenes und eindeutiges (oder auch kontrolliertes) Vokabular vorausgesetzt, dieses setzt sich aus einer Menge von Termen zusammen, welche die Konzepte eines Diskursbereichs beschreiben.

2.7.1 Multi-perspektivische Schemahierarchien

Taxonomien reichen in der Realität jedoch noch nicht aus, um natürliche Denkprozesse im Sinne mentaler Gedankengänge nachzubilden. Wenn wir uns unsere Beispielwelt mit Rudi, seinem Arbeitgeber und seiner Familie vergegenwärtigen, so werden wir feststellen, dass diese Welt, in Abhängigkeit von Blickwinkel und Ausschnitt, ganz anders erscheinen kann. Assoziationen und Imaginationen sind multi-perspektivisch und erlauben subjektive Anschauungen und Sichtweisen auf bestimmte Konzepte der Welt.

Nehmen wir Rudi: Rudi ist in seiner Rolle bei der CarFS eine Instanz der Klasse Sachbearbeiter, deren Oberklasse die Klasse Angestellter ist, die eine Spezialisierung der Klasse Arbeitnehmer darstellt, welche wiederum eine spezielle Gruppe von Erwerbstätigen repräsentiert. Diese Taxonomie ist in Abbildung 2.9 dargestellt.

Für alle Klassen der Taxonomie lassen sich Fächer definieren, welche die Eigenschaften der jeweiligen Instanzen enthalten. Angestellte sind Arbeitnehmer und erben daher die Eigenschaften Vertragsstatus, Vertragslaufzeit, Urlaubstage, Arbeitszeit und Monatsabrechnung. Gleiches gilt für die benachbarten Unterklassen Arbeiter und Beamte. Arbeiter, Angestellte und Beamte haben allesamt die gleichen Arbeitnehmerattribute.

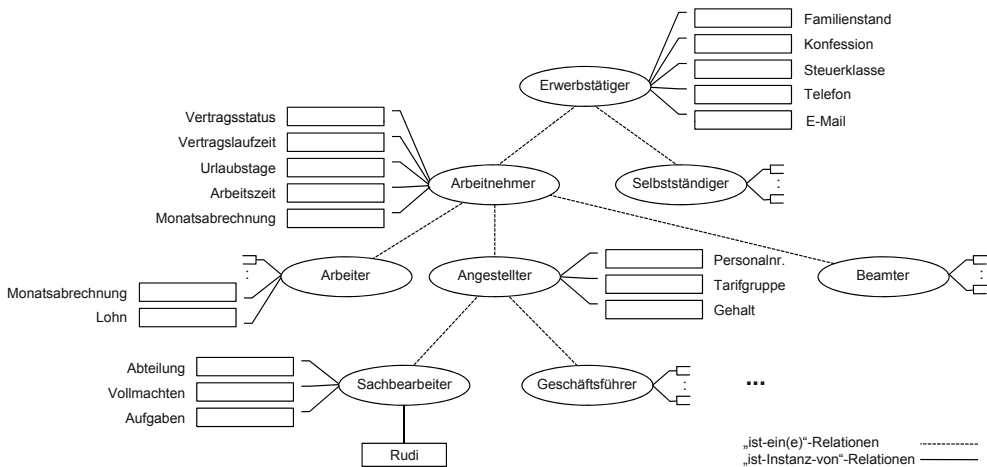


Abb. 2.9 Taxonomie zur Beschreibung von „Erwerbstätigen“.

Auffallend ist, dass in Abbildung 2.9 bei der Klasse Arbeiter nochmals das Fach Monatsabrechnung auftaucht, obwohl dieses bereits in der allgemeineren Klasse Arbeitnehmer angegeben ist. Das bedeutet, dass die beiden Spezialisierungen der Klasse Arbeitnehmer, Angestellte und Beamte, die Eigenschaft ererben, für die Unterklasse Arbeiter jedoch verändert wurde. Ein solcher Fall trifft zu, wenn die verallgemeinernde Eigenschaft der Oberklasse nicht passt und für die Unterklasse Besonderheiten gelten. Daher wird die Klasseneigenschaft neu definiert. Sie überdeckt damit die entsprechende Eigenschaft der Oberklasse. Man spricht daher auch von einer *Überdeckungsbeziehung* zwischen Unterklasse und Oberklasse.

Die Erklärung für diese Überdeckungsbeziehung im konkreten Beispiel in Abbildung 2.9 liegt darin begründet, dass es in Rudis Unternehmen, der CarFS, nicht nur Angestellte gibt. In der Posteingangsstelle sitzen auch Arbeiter, welche im Zweischichtbetrieb die vielen Zehntausend täglich eingehenden Dokumente (Anträge, Kündigungen, Adressänderungen, Schadenberichte, Gutachten, ...) bearbeiten und die prozessrelevanten Daten in Datenmasken am Bildschirm eingeben müssen. Für sie gilt eine spezielle Form der Monatsabrechnung, die sich von den Angestellten der CarFS unterscheidet. Dies hängt mit den Besonderheiten der Entgelte im Schichtbetrieb, z. B. spezielle Zuschläge, und der damit verbundenen Abrechnungsweise zusammen. Die Überdeckung führt in diesem Fall also dazu, dass die Definition der Monatsabrechnung der Klasse Arbeitnehmer in der Unterklasse Arbeiter außer Kraft gesetzt wird und andere Regeln und Modalitäten den individuellen Arbeitslohn berechnen.

Da Rudi eine Instanz der terminalen Klasse der Hierarchie entspricht, besitzt das entsprechende Objekt alle Eigenschaften von Erwerbstätigen, Arbeitnehmern, Angestellten und Sachbearbeitern gleichzeitig. In seinen Fächern sind – wie in Abbildung 2.10 veranschaulicht – dementsprechend konkrete Werte eingetragen.

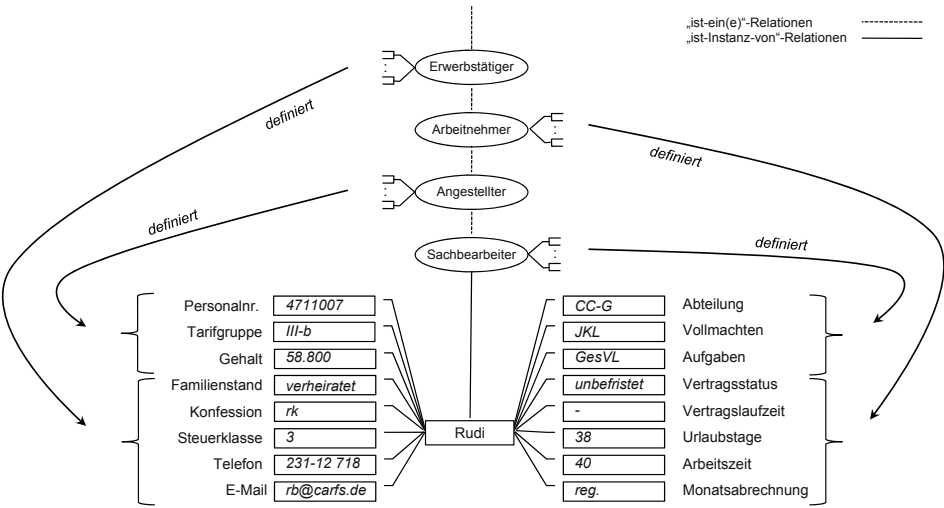


Abb. 2.10 Instanz „Rudi“ mit entsprechender Wertebelegung.

Jede der Klassen in Abbildung 2.10 definiert einen Teil der Eigenschaften, die den Erwerbstätigen Rudi beschreiben. Man kann die taxonomische Beschreibung von Rudi auch wieder in der umschreibenden Form lesen:

Rudi „ist-Instanz-von“ Sachbearbeiter „ist-ein(e)“ Angestellter
„ist-ein(e)“ Arbeitnehmer „ist-ein(e)“ Erwerbstätiger.

2.7.2 Probleme bei unterschiedlichen Rollen

Rudi hat in der Welt, in der er lebt, jedoch noch ganz andere Rollen, die teilweise oder vollständig unabhängig von seiner beruflichen Tätigkeit sind. Folglich könnte Rudi, je nach Zielsetzung, auch unter den Aspekten seiner sozialen oder biologischen Rolle betrachtet werden, die wiederum anderen Generalisierungsaspekten folgen, etwa:

Rudi „ist-Instanz-von“ Vater „ist-ein(e)“ Elternteil „ist-ein(e)“ Lebewesen, oder
Rudi „ist-Instanz-von“ Mensch „ist-ein(e)“ Säugetier „ist-ein(e)“ Wirbeltier.

Die Frage stellt sich, wie man diese neuen Aspekte in die bestehende Wissensrepräsentation einbeziehen kann. Das Problem, mit dem wir uns damit auseinandersetzen müssen, stellt keine einfache Erweiterung einer bestehenden Taxonomie mehr dar. Die Klasse Mensch könnte man noch als Verallgemeinerung eines Erwerbstätigen ansehen, jedoch müsste man dann auch eine Klasse Nichterwerbstätige einführen, was nicht besonders elegant wäre. Noch mehr Schwierigkeiten hätten wir mit der sozialen Rolle von Rudi, denn nicht jeder Vater ist erwerbstätig und auch ist nicht jeder Erwerbstätige gleichzeitig Vater. Eine einfache Erweiterung der Taxonomie in Abbildung 2.9 ist daher sehr schwierig und nicht widerspruchsfrei zu lösen.

Eine Option, die sich anbietet bestünde darin, die Taxonomie mit einer allgemeineren Klasse so zu erweitern, dass alle drei Rollenkonzepte abbildbar sind. Aber auch diese Lösung ist, wie Abbildung 2.11 darlegt, nicht befriedigend, da Rudi als Konsequenz unterschiedlichen Konzepten entsprechen und über die Vererbung entlang der jeweiligen Klassenhierarchie ganz oder teilweise andere Eigenschaften besitzen würde. Rudi würde also in Form von mehreren Instanzen existieren, obwohl es sich um ein und dasselbe Individuum handelt, was in der Wissensbeschreibung nicht abbildbar ist.

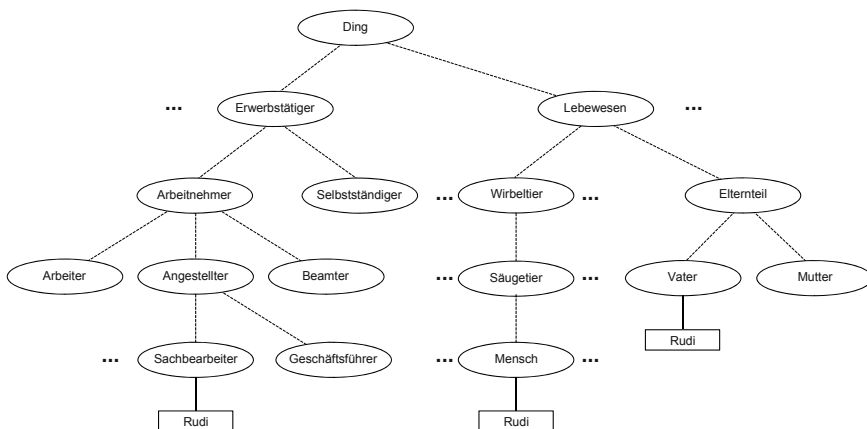


Abb. 2.11 Erweiterte Taxonomie mit Rollenbeschreibungen.

Das Problem der Taxonomie ist also die inhärente Annahme, dass die Klassen sich auf jeder Ebene gegenseitig ausschließen, in ihrer Gesamtheit aber abdeckend sind.

Das Beispiel in Abbildung 2.11 macht folgende Aussage: Ein Ding ist ENTWEDER ein Erwerbstätiger ODER ein Lebewesen, nicht aber beides, denn nur so erreicht die Taxonomie die effiziente Unterteilung des Suchraums, was in der Praxis, wie das Beispiel zeigt, eben nicht immer funktioniert.

Wir werden später sehen, dass es mit Thesauri und Topic Maps noch zwei Ansätze gibt, die im Bereich Semantischer Technologie eingesetzt werden und die es erlauben, weitere Beziehungen zwischen Begriffen einer Taxonomie einzuführen. Wir wollen diese jedoch erst später (vgl. Kapitel 3) behandeln und uns im Folgenden zunächst noch ein wenig mit dem Schemagedanken auseinandersetzen.

Bisher haben wir uns mit Schemata beschäftigt, die auf Grundlage von Frames Objekte und deren Eigenschaften beschrieben haben. Wir haben jedoch bereits erwähnt, dass viele sprachliche Akte auch Wissen über Ereignisse und deren zeitliche Abfolge enthalten.

2.8 Conceptual Dependencies

In den 1970er Jahren begann Roger C. Schank, Professor an der Yale University in Connecticut die sogenannte Conceptual Dependency Theory (CDT) zu entwickeln. Dem Schemagedanken von Bartlett folgend (vgl. Abschnitt 2.6.1), entwarf er ein System zur Darstellung der Bedeutung von sprachlichen Äußerungen. In ihrem Buch „*Scripts, plans, goals and understanding*“ schlagen Schank und Abelson (Schank und Abelson 1977) einen umfassenden Ansatz vor, bestimmte Ereignisabläufe bzw. musterhafte Standardszenen strukturiert zu repräsentieren. Shank bezeichnet diese speziellen Schemata als *Scripts*. Scripts beschreiben stereotype Verhaltensweisen und Kommunikationsformen in vertrauten situativen Kontexten, wie z. B. die Abfolge der Tätigkeiten beim Backen von Zimtgebäck, die Handlungsfolge beim Abschluss von Leasinganträgen bei der CarFS oder das Benutzen der S-Bahn zwischen Ottobrunn und München.

Solche mentalen Drehbücher liefern Anweisungen darüber, wie man sich in einer bestimmten Situation verhält und handelt. Schank hat sich dabei intensiv mit dem Aspekt des Lernens auseinandergesetzt, und Scripts zunächst einzelfallspezifisch oder kulturabhängig betrachtet, um die gelernten Muster sozusagen „zu entkulturalisieren“, d. h. sie so zu verallgemeinern, dass sie als generisches Ablauf- oder Prozesswissen repräsentiert und im Kontext abgerufen werden können. Damit simulieren Scripts menschliches Verhalten. Mit anderen Worten: So wie wir selbst in unseren mentalen Strukturen Handlungen durch mehrmaliges Wiederholen anpassen und erneut durchführen, wissen wir dann in gegebenen Situationen einfach, was von uns erwartet wird bzw. was sich gehört.

Scripts sind demnach eine Art von Erfahrungswissen, das wir nach einer gewissen Zeit internalisiert haben und anwenden können. Ein Script hat einen allgemeinen Aufbau, der folgende Informationen zusammenfasst:

- Art des Scripts
- beteiligte Entitäten, als „Dinge“, die in der (Ab)handlung vorkommen
- Rollen beteiligter Akteure
- Vorbedingungen, die gelten müssen, damit die Ereignissequenz der Handlung stattfinden kann
- Nachbedingungen, die bestehen, nachdem die Ereignissequenz abgeschlossen ist
- Ereignisse, welche die (Ab)handlung bestimmen, wobei die Ereignisse selbst rekursiv definiert sein können, also verfeinerte Schemata darstellen, in welchen weitere Scripts beschrieben sind.

2.8.1 Handlung und Zustand

Jedes Script enthält eine detaillierte Beschreibung der Ereignisse und Aktionen und der jeweils beteiligten Akteure, die mit den Rollen beschrieben sind. Die den Scripts zugrunde liegende Conceptual Dependency Theory greift dazu auf eine relativ einfache Notation zurück. Sprachliche Äußerungen entsprechen entweder Begriffen, die für eine Handlung stehen (*active conceptualization*) oder die einen aus einer Handlung resultierenden Zustand (*stative conceptualization*) adressieren. Man verwendet dazu die folgenden Schreibweisen:

Aktor Aktion Objekt Richtung [Instrument]
Objekt (ist in) Zustand [mit Wert].

Mit *active conceptualizations* lassen sich Ereignisse ausdrücken, wie etwa „Rudi fährt mit der S-Bahn von Ottobrunn nach München“. Auf solche Ereignisse und ihre Modellierung wollen wir uns in den nächsten Abschnitten beschränken, um die Möglichkeiten von Scripts zu beschreiben.

2.8.2 Handlungsoptionen

Ähnlich wie in anderen bereits diskutierten Wissensrepräsentationsformen stehen *Actor* und *Object* für Individuen und Entitäten, die mit Eigenschaften beschrieben sind. Die Aktionsmenge eines Scripts umfasst eine Menge „primitiver“ Aktionen, die ausdrucksstark genug sind, um sämtliche sprachliche Äußerungen semantisch abzubilden. Die Aktionen werden, ähnlich wie Computerbefehle, durch mnemonische Kürzel bezeichnet, die da sind:

- ATRANS:** Übertragung einer Beziehung zu einem Objekt (wie z. B. beim „geben“)
- PTRANS:** Physischer Transfer eines Objektes (wie z. B. „bewegen“ oder „gehen“)
- PROPEL:** Anwendung physischer Kraft auf ein Objekt (wie z. B. etwas „antreiben“ oder „schieben“)
- MOVE:** Bewegen eines Körperteils durch Akteur (z. B. „treten“, „zucken“, „kauen“)
- GRASP:** Aufnehmen eines Objektes durch einen Akteur (z. B. „ergreifen“)
- INGEST:** Einnahme eines Objektes durch einen Akteur (z. B. „trinken“, „essen“)
- EXPEL:** Ausstoß eines Objektes durch einen Akteur (z. B. „schwitzen“, „spucken“)
- MTRANS:** Übertragung von Information zwischen Akteuren (z. B. „sprechen“)
- MBUILD:** Mentale Erzeugung neuer Information (z. B. durch „entscheiden“, „aufschreiben“)
- CONC:** Erzeugung von Ideen (z. B. „nachdenken“)
- SPEAK:** Erzeugung von sprachlichen Lauten (z. B. „rufen“, „vortragen“)
- ATTEND:** Sensorische Fokussierung (z. B. „zuhören“, „hinsehen“)

Die Aktion **PTRANS** z. B. steht also für eine Ortsveränderung eines Objektes und deckt damit eine ganze Reihe von Verben als **PTRANS** -Handlungsbegriffe ab. „Ich gehe“ bedeutet also in der Script-Sprache „ich **PTRANS** mich irgendwohin“ oder die Handlung „Ich lege“ bedeutet übersetzt „ich **PTRANS** irgendetwas irgendwohin“. Die Aktion **ATTEND** umschreibt die Aktion des Ausrichtens eines Sinnesorgans auf einen Reiz, etwa „**ATTEND** Ohr“ ist demnach soviel wie „hören“ und „**ATTEND** Auge“ steht für „sehen“. Um die gerade gemachte Aussage „Rudi fährt mit der S-Bahn von Ottobrunn nach München“ zu beschreiben, kann man sich folgende Bindung der Ereignisvariablen vorstellen:

Aktor:	Rudi
Aktion:	PTRANS
Objekt:	Rudi
Richtung TO:	München
Richtung FROM:	Ottobrunn

Schank hat sich intensiv mit der Sprache auseinandergesetzt und dabei verblüffender Weise festgestellt, dass man also nur ein Dutzend abstrahierender Aktionen (abgesehen von wenigen Ausnahmen) braucht, um eine große Anzahl von Handlungen (durch Verben ausgedrückt) zu repräsentieren. Jede primitive Aktion steht also als Repräsentant einer ganzen Menge von Ereignissen. Scripts repräsentieren also allgemeine Situationen in einer Wissensbasis, die mittels einer Sequenz von Szenen prozessartig beschrieben werden. Die Szenen selbst werden durch eine Sequenz primitiver Aktionen dargestellt, um diese als Muster für die Interpretation von sprachlichen Abhandlungen zu verwenden.

2.8.3 Mentale Drehbücher

Betrachten wir dazu wieder eine Situation aus unserer Beispielwelt. Wir wissen allgemein gut darüber Bescheid, wie ein Restaurantbesuch normalerweise abläuft, da wir bereits einige Erfahrungen damit gesammelt haben. Daher wissen wir auch, wie wir situativ agieren und in welcher Reihenfolge dies geschieht. Betrachten wir uns dazu die folgende Abhandlung:

„Rudi hat sich nach Feierabend in der Münchner Innenstadt zum Abendessen mit seiner Familie verabredet. Er trifft sich mit Anke und den Kindern vor dem Restaurant. Nachdem sie hineingegangen sind, suchen sie einen freien Tisch. Sie finden einen Tisch am Fenster und nehmen Platz, ...“

Man kann den weiteren Ablauf des Abends, d. h. das Bestellen der Speisen, den Verzehr dergleichen, das Ordern der Rechnung und das Bezahlen usw., in der gleichen Form weiter beschreiben, kann aber stattdessen auch ein allgemeines Script für einen Restaurantbesuch erstellen, das die vollständige Ereignisfolge modelliert. Das entsprechende Script ist in Abbildung 2.12 gezeigt.

Die im Beispiel aufgeführten Ereignisse sind noch sehr allgemein gehalten, fassen aber den wesentlichen Ablauf eines Restaurantbesuchs, wie bei einem Drehbuch, in Szenen zusammen. Alle an einer Szene beteiligten Rollen werden explizit festgehalten.

Um die individuellen Szenen weiter zu detaillieren, verwenden wir die von Schank und seinen Mitarbeitern definierten primitiven Aktionen. Sie ermöglichen es uns, eine allgemeine Ereignisfolge festzulegen und als Wissen zu repräsentieren. So lässt sich die erste Szene, also für das Eintreten und Platz nehmen im Restaurant, wie in Abbildung 2.13 darstellen.

Die Ereignisse, die in der Szene beschrieben werden, korrespondieren direkt mit den Verben in der gerade beschriebenen Kurzgeschichte. Das „Eintreten“ in das Restaurant entspricht demzufolge einer **PTRANS**-Aktion und das „Platz Nehmen“ einem **MOVE**. Alle Ereignisse einer Sequenz werden in der Regel als chronologisch betrachtet, so dass implizit angenommen werden kann, dass sich mit Ausführung einer Aktion auch Zustände z. B. von Akteuren verändern können.

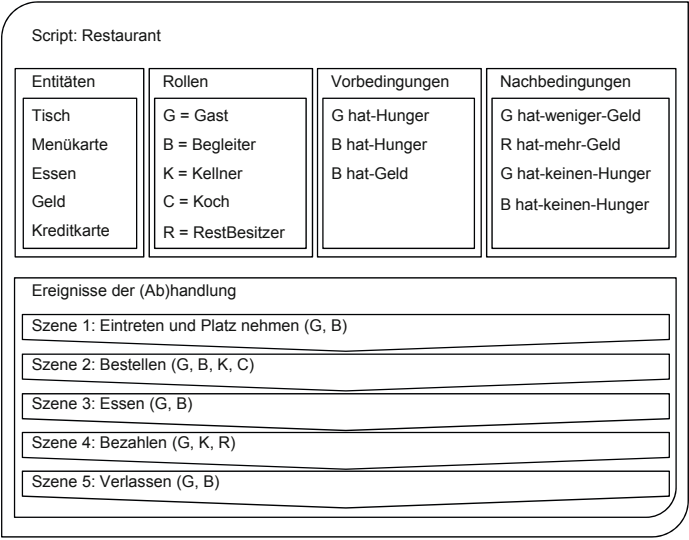


Abb. 2.12 Exemplarisches Script für den Besuch eines Restaurants.

Während wir für Szene 1 noch eine einfache Sequenz von Ereignissen festlegen können, deren Fortsetzung i. d. R. direkt im Bestellen der Speisen besteht, hängt Letzteres selbst von einigen individuellen Gepflogenheiten des besuchten Restaurants ab. So ist der weitere Fortgang von Szene 2 davon abhängig, ob beispielsweise eine Speisekarte bereits auf dem Tisch liegt, der Kellner diese von sich aus bringt oder gar der Gast den Kellner erst darum bitten muss.

Aus diesem Grund müssen wir solche Optionen als alternative Ereignisse in einer entsprechenden Szenenbeschreibung definieren. Abbildung 2.14 gibt einen Einblick, wie eine Ereignisfolge beim Bestellen von Speisen aussehen könnte. Durch die verschiedenen Ausgangspunkte jeder Option und Abfolgealternativen der Ereignisse im weiteren Fortgang der Szene wird die Scriptbeschreibung für die Szene 2 schon etwas komplexer.

Auch die weiteren Szenen lassen sich in der gleichen Art und Weise beschreiben. Hierzu verweisen wir auf (Schank und Abelson 1977).

Scriptdarstellungen und ihre Beschreibung, wie sie in den Abbildungen 2.12 bis 2.14 exemplarisch aufgegriffen wurden, dienen dazu, natürlichsprachlichen Text zu analysie-

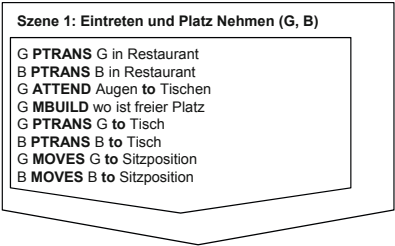


Abb. 2.13 Ereignisfolge beim Eintreten und Platz Nehmen im Restaurant.

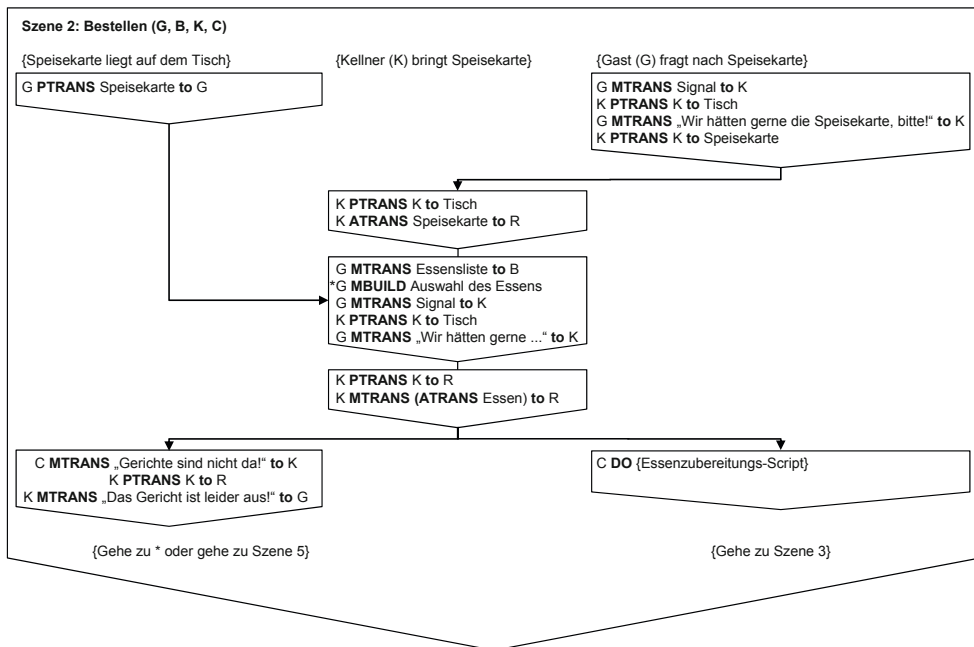


Abb. 2.14 Ereignisfolge für das Bestellen von Speisen.

ren und ihn mittels Parsing in eine Conceptual Dependency (CD)-Repräsentation zu transformieren. Gängige Natural Language Processing (NLP)-Verfahren gehen dabei i. d. R. mehrstufig vor: Zunächst werden die relevanten Wörter auf Grundlage eines kontrollierten Vokabulars gesucht, um dann daraus – gemäß den Regeln einer vorgegebenen Grammatik – einen Syntaxbaum zu konstruieren. Erst mit diesem Syntaxbaum wird die semantische und pragmatische Bedeutung gewonnen.

Im Gegensatz zu diesem klassischen Vorgehen kommt der CD-Ansatz weitgehend ohne syntaktische Analyse aus, denn wann immer ein Verb des Lexikons im Text auftaucht, wird versucht, dieses an eine Menge von primitiven Aktionen zu binden. Solche *expectations* (als Menge alternativer Bedeutungen), wie sie Schank nennt, werden dann von dem Parser verwendet, um einzelne Aspekte der bisher bestehenden begrifflichen Struktur (bereits gebundene primitive Aktionen einer Szene) zu untersuchen, um bestehende Lücken aufzufüllen oder um neue Strukturen zu binden.

Zusammenfassend kann man sagen, dass Scripts eine spezielle Form von Schemata sind, die Handlungsfolgen repräsentieren können, im Unterschied zu Plänen aber nicht unbedingt ein Ziel benötigen. Sie stellen Wissen in Form von Schablonen zur Verfügung, welche jeweils eine verallgemeinerte Folge von Ereignissen beschreibt, zu welchen eine ganze Menge unausdrückliches Alltagswissen vorausgesetzt ist. Viele Details, wie etwa, dass auch Information zum Tempus einer Aussage mittels CDs abgebildet werden kann, sind hier weggelassen. Schanks Programme berücksichtigen solche Details aber durchaus, es ist ganz erstaunlich, wie mächtig sich dieser doch sehr primitiv anmutende

Apparat erweist. Dabei arbeiten Scripts – wie Schank ausführt und zeigt – gar nach Prinzipien, die ausdrücklich von konkreten, natürlichen Sprachen unabhängig sind.

2.9 Taxonomien mit formaler Semantik: Die KL-ONE-Familie

Intuitiv sind die Ansätze der Framesysteme und der Taxonomien unmittelbar einsichtig: Die Beschreibung von Objekten der Welt mit ihren Eigenschaften und die Betonung von Abstraktion und Spezialisierung entsprechen durchaus der üblichen Begriffsbildung und sind daher offensichtlich geeignet, menschliches Denken und Wissen zu repräsentieren. Aus Sicht einer computerorientierten Wissensrepräsentation haben die bisher vorgestellten Ansätze jedoch den Nachteil, dass die Bedeutung der jeweils modellierten Elemente zu einem wesentlichen Teil nur durch die Interpretationsleistung der auf den jeweiligen Datenstrukturen arbeitenden Programme bestimmt wird. Selbst die auf den ersten Blick so eindeutig erscheinende Subsumptionsrelation in einer taxonomischen Struktur kann in konkreten Realisierungen unterschiedlich interpretiert werden: Wie geht man etwa praktisch mit dem Problem der multiplen Vererbung um?

Um wissensbasierte Systeme für den Menschen effizienter nutzen zu können, bedarf es einer wohldefinierten Semantik, die sowohl von Menschen als auch von Maschinen im weitestgehenden Sinne interpretiert werden kann. Dazu benötigen wir einen Formalismus, der einerseits über eine ausreichende Ausdrucksmächtigkeit verfügen muss, um die relevanten Konzepte in ihrer nötigen Informationstiefe darstellen zu können, aber andererseits nicht zu ausdrucksmächtig sein darf, damit die Berechnungen der impliziten Beziehungen durch vorhandene Reasoning-Tools nicht undurchführbar werden. Aus solchen Überlegungen heraus wurde eine umfangreiche Familie von Repräsentationssprachen entwickelt, die auf der aus der Prädikatenlogik hervorgegangenen Beschreibungslogik fundieren.

Durch die fehlende Semantik bei Frames und Semantischen Netzen war es nicht ersichtlich, ob die Schlussfolgerungsprozedur eines Systems korrekt, vollständig oder entscheidbar ist. Als man Mitte der 1980er Jahre schließlich erkannte, dass Frames und semantische Netze als Fragmente der Prädikatenlogik betrachtet werden können, da sie im Wesentlichen aus einstelligen und zweistelligen Prädikaten bestanden, die jeweils eine Gruppe von Individuen bzw. Beziehungen zwischen diesen bezeichneten, entstand die Beschreibungslogik aus der bereits zur Verbesserung semantischer Netze entwickelten Sprache KL-ONE (Brachman und Schmolze 1985).

Beschreibungslogiken sind eine Familie von logikbasierten Formalismen und repräsentieren das implizite und explizite Wissen über die Struktur einer Anwendungsdomäne in Form von *Konzepten* und *Rollen*.

Ein Konzept wird hierbei als Begriff verstanden, der wie die einstelligen Prädikate der Prädikatenlogik für sich selbst steht, ohne die Existenz zweier Objekte zu implizieren. Ein Konzept, genauer gesagt ein generisches Konzept, entspricht somit einem Schema,

das viele Individuen in einer Welt beschreiben kann. So wie sprachliche Begriffe unter Verwendung anderer Begriffe definiert werden können, kann auch das KL-ONE Konzept mit Hilfe anderer Konzepte gebildet werden. Ein Konzept ist also ein allgemeiner Term der durch KL-ONE definierten Sprache.

Konzepte repräsentieren die Begriffe der zu modellierenden Welt. Dabei werden in KL-ONE definierte und primitive Konzepte unterschieden:

- *Definierte Konzepte* sind von anderen Konzepten abgeleitet und werden durch ihre Definition vollständig beschrieben, die in der Begriffsdefinition angegebenen Bedingungen sind notwendig und hinreichend zur Identifizierung des Konzepts.
- *Primitive Konzepte* sind im Gegensatz dazu nicht vollständig beschrieben, d. h. es ist keine hinreichende Bedingung angegeben. Wohl aber können beliebig viele notwendige Bedingungen aufgeführt werden. Dieses Sprachelement trägt dem Wunsch Rechnung, bei der Modellierung von Begriffswelten Individuen aus der realen Welt abbilden zu können, ohne eine vollständige Beschreibung erstellen zu müssen. (Eine solche vollständige Beschreibung ist u. U. nicht möglich und i. d. R. nicht nötig).

Ein Konzept in KL-ONE hat einen *Namen*, der für die Definition des Konzepts keine Bedeutung hat, aber das Konzept identifiziert. Beschrieben wird es durch die Angabe seiner *Superkonzepte* und seiner *Rollen* (und deren Beziehungen und Strukturen).

2.9.1 Superkonzepte

Die Superkonzepte eines Konzeptes sind ihrerseits Konzepte. Ein Superkonzept subsumiert sein Subkonzept, es repräsentiert einen Oberbegriff zum Subkonzept. In der für semantische Netze üblichen Sprechweise wird die Beziehung Subkonzept → Superkonzept durch einen isa-Link dargestellt. Die Angabe mehrerer Superkonzepte bei der Definition eines Konzeptes ist als Konjunktion zu verstehen, das Subkonzept erhält die Konjunktion aller Bedingungen seiner Superkonzepte.

Dieser Vorgang entspricht in etwa der bei den Framesystemen beschriebenen Vererbung. Im Gegensatz zu der in frameartigen Systemen üblichen Verwendung des Begriffs ist die Vererbung in KL-ONE jedoch strikt, ein Konzept kann Bedingungen seiner Superkonzepte nicht überdecken, auflösen oder sonst in irgendeiner Weise ignorieren. Dieses Verhalten entspricht dem intuitiven Verständnis der Begriffsdefinition durch Angabe von Konjunktionen von Oberbegriffen (vgl. Abschnitt 2.6.5).

2.9.2 Rollen

Eine *Rolle* ist eine Relation oder Eigenschaft, die wie die zweistelligen Prädikate der Prädikatenlogik mindestens zwei Begriffe miteinander verbindet. Rollen beschreiben die Struktur eines Konzeptes also näher (wie die *Fächer* eines Frames). Eine Rolle ist als Restriktion auf die Konjunktion der Superkonzepte aufzufassen. Als "Generalisierte Attributbeschreibung" repräsentiert eine Rolle mögliche Beziehungen zwischen durch Konzepte beschriebenen Individuen.

Die durch die Rolle gegebene Beschreibung gliedert sich in die *Wertrestriktion* (Value Restriktion V/R), die das Konzept des Individuums bestimmt, das die Rolle füllen kann, und die *Restriktion/Kardinalitäts-* (Number Restriction N/R), die festlegt, wie viele Rollenfüller der angegebenen Rolle ein Individuum des beschriebenen Konzepts mindestens haben kann bzw. höchstens haben darf.

Die durch die Rollen angegebenen Bedingungen sind als notwendig zu verstehen, d. h. ein Individuum, das eine der durch die Rollen angegebenen Restriktionen nicht erfüllt, kann nicht Instanz des fraglichen Konzepts sein. Zur Illustration betrachten wir wieder die Arbeitswelt von Rudi und definieren beispielsweise die Führung einer Gruppe im Unternehmen als Konzept, das aus beliebig vielen Sachbearbeitern besteht, wobei ein Sachbearbeiter die Rolle des Gruppenleiters übernimmt. Die in KL-ONE übliche grafische Darstellung dieser Konzeptdefinition ist in Abbildung 2.15 gezeigt.

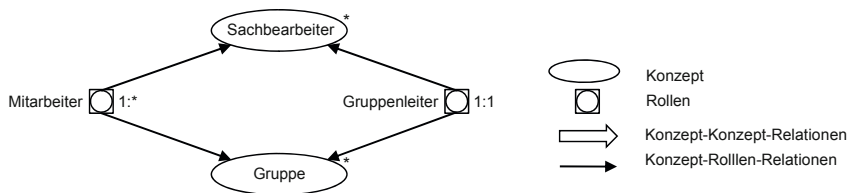


Abb. 2.15 „Gruppe“ in KL-ONE-Darstellung.

Rollen machen Aussagen über Bedingungen, die an Individuen zu stellen sind, damit diese in der angegebenen Beziehung zu einem Individuum des jeweiligen Konzepts stehen können. Somit beschreiben Rollen die potenziellen Rollenfüller. Im Beispiel könnte also niemand Mitarbeiter der Gruppe werden, der nicht gleichzeitig Sachbearbeiter ist!

2.9.3 Vererbung und Rollenrestriktionen

Rollen werden von den Superkonzepten an die Subkonzepte weitervererbt. Eine ererbte Rolle kann an einem Konzept weiter eingeschränkt werden, und zwar sowohl nach Wert als auch nach Kardinalität.

Betrachten wir eine spezielle „Taskforce“, wie in Abbildung 2.16 dargestellt und wie sie bei der CarFS in vielen Fällen gegründet wird. Aus Gründen der Effizienz soll diese

mindestens drei und maximal sechs Mitarbeiter (plus Gruppenleiter) umfassen. Durch die entsprechende Einschränkung der Mitarbeiter-Rolle ergibt sich die neue Definition.

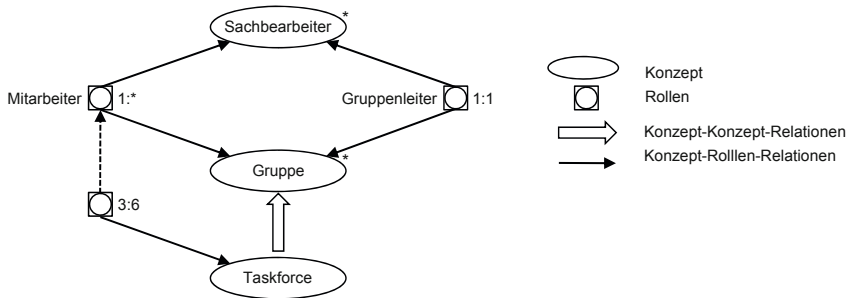


Abb. 2.16 Taskforce als Einschränkung einer Gruppe.

Wird eine Rolle von zwei Superkonzepten ererbt und hat sie bei diesen Superkonzepten unterschiedliche Wert- oder Kardinalitätsrestriktionen, so erhält das neue Konzept die Konjunktion der Restriktionen.

Eine ererbte Rolle kann an einem Konzept in mehrere Teilrollen aufgespalten werden, diese neuen Rollen stellen dann Subrelationen der durch die ererbte Rolle beschriebenen Relation dar. Dieser Vorgang wird als *Rollendifferenzierung* bezeichnet. Die differenzierenden Rollen erben von der differenzierten Rolle die Minimums-Restriktion und die Wertrestriktion. Das Maximum wird für jede differenzierende Rolle neu bestimmt, wobei aber keines der Maxima größer als das Maximum der differenzierten Rolle sein darf.

2.9.4 Wohlgeformte Terme und ihre Semantik

Damit ein aus den Sprachelementen zusammengesetztes Konzept wohlgeformt ist, muss mindestens eine der folgenden Bedingungen erfüllt sein:

- Das Konzept muss mehr als ein Superkonzept haben, ein Konzept ohne eigene lokale Restriktion ist definiert als die Konjunktion seiner Superkonzepte.
- Das Konzept muss sich von seinem Superkonzept in mindestens einer Restriktion unterscheiden.
- Das Konzept ist primitiv.

Diese Grundlagen genügen, um uns die wesentlichen Eigenschaften von KL-ONE (und ähnlichen Sprachen) nochmals zu vergegenwärtigen:

- Die *Bedeutung eines Konzepts* hängt nur von seinen lokalen Restriktionen und seinen Superkonzepten ab. Die Definition von Begriffen (dargestellt durch die Konzepte) erfolgt in KL-ONE also streng objektzentriert und lokal.
- Die *Objekte* in KL-ONE sind durch ihre Beziehungen (Subsumption und Rollenbeziehung) zu anderen Objekten definiert. Die resultierenden Verbindungen legen nahe, eine Menge von KL-ONE-Konzeptdefinitionen als Spezialfall eines semantischen Netzes aufzufassen. Die Konzepte entsprechen den Knoten, die Kanten werden durch Subsumptionsbeziehungen und Rollen dargestellt.
- Die *Restriktionen eines Konzepts* gelten für alle seine Subkonzepte, es herrscht eine strenge Vererbung; Überdecken oder Auflösen ererbter Restriktionen ist nicht möglich.

Diese Eigenschaften erlauben die Definition einer strengen *Mengensemantik* für KL-ONE. Ein Konzept wird dabei als Beschreibung einer Menge von Individuen einer Welt aufgefasst. Die Konjunktion zweier Konzepte entspricht der Schnittmengenbildung zwischen den durch die Konzepte beschriebenen Mengen. Die Subsumption zwischen zwei Konzepten ist gleichbedeutend mit der Teilmengenbeziehung zwischen den durch die Konzepte repräsentierten Mengen.

Im Unterschied zu beliebigen semantischen Netzen ist also die Bedeutung eines KL-ONE-Netzes unabhängig von den darauf operierenden Algorithmen festgelegt. KL-ONE-Darstellungen erlauben damit implementierungsunabhängige Darstellungen von Begriffsdefinitionen und eignen sich deshalb auch als Kommunikationsmedium.

2.9.5 Subsumption und Klassifizierung

Die Frage nach der Subsumption zwischen zwei Konzepten ist die wichtigste Frage, die ein KL-ONE-Netzwerk beantworten kann. Semantisch entspricht dies, wie bereits erläutert, der Frage nach der Teilmengenbeziehung zwischen zwei durch die Konzeptdefinitionen beschriebenen (aber nicht aufgezählten!) Mengen.

Diese nur mit Aufwand beantwortbare Frage reduziert sich auf die Frage nach Existenz einer Kante im semantischen Netz, falls alle Subsumptionsbeziehungen zwischen Konzepten im KL-ONE-Netz explizit eingetragen sind. Zu diesem Zweck existiert ein eigener Algorithmus, der sogenannte *Classifier*. Seine Aufgabe ist es, neue Konzeptdefinitionen so in das bestehende Netzwerk einzutragen, dass alle Subsumptionsbeziehungen korrekt vermerkt werden. Ein neues Konzept wird dabei in einer Form eingetragen, dass es Superkonzept all seiner Subkonzepte und Subkonzept all seiner Superkonzepte ist, wobei natürlich die Vererbung zu berücksichtigen ist.

Der zweite wichtige Dienst, den ein KL-ONE-System bereitstellt, wird als *Realisierung* bezeichnet: Bei gegebener Beschreibung eines unbekannten Objektes kann das Konzept aus dem Netzwerk ermittelt werden, das dem beschriebenen Objekt entspricht oder es

subsumiert. Je nach Wunsch kann das speziellste Superkonzept oder das allgemeinste Subkonzept zu der Beschreibung ermittelt werden.

Findet in unserem Beispiel etwa das KL-ONE-System Fakten, die aussagen, dass Rudi Chef einer Gruppe ist, zu der noch vier weitere Angestellte als Mitarbeiter aufgelistet sind, so würde es logisch folgern, dass diese Gruppe eine Taskforce (wie schon definiert) darstellt.

Es zeigt sich allerdings, dass der Subsumptionstest – und damit die automatische Klassifizierung – mit wachsender Ausdrucksmächtigkeit der Repräsentationssprache sehr schnell logisch unentscheidbar wird. Gleiches gilt für die Realisierung, die sich als Subsumptionstest auf einer Menge von Fakten verstehen lässt. Damit sind der Anwendung für die Modellierung und vollständige Verarbeitung komplexer Realitäten Grenzen gesetzt. Verschiedene Nachfolgeentwicklungen haben versucht, jeweils geeignete Lösungen für praktische Anforderungen zu finden. So werden etwa zusätzliche Modellierungsprimitive eingeführt, um die betrachtete Realität einfach darzustellen, wobei auf die Entscheidbarkeit des Subsumptionstests verzichtet wird. Die verschiedenen Nachfolge-Systeme unterscheiden sich daher in ihrer Ausdrucksmächtigkeit, den vorgesehenen Operationen und den Reasoningfähigkeiten über den zur Modellierung der Realität gespeicherten Fakten. Ein Überblick über die Familie und die jeweiligen Kompromisse ist etwa in (Woods und Schmolze 1992) zu finden.

Beschreibungslogik ist also auf die Tatsachen zurückzuführen, dass zum einen wichtige Begriffe einer Anwendungsdomäne durch Konzeptbeschreibungen dargestellt werden, indem komplexere Ausdrücke durch atomare Konzepte und atomare Rollen durch die jeweiligen Konstruktoren gebildet werden. Zum anderen besitzen Beschreibungslogiken im Gegensatz zu Semantischen Netzen und Frames eine wohldefinierte, logikbasierte Semantik. Das bedeutet, dass eine strukturelle Unterscheidung zwischen der Terminologie einer Anwendungsdomäne und der Menge von individuellen Objekten dieser Terminologie vorgenommen wird und komplexe oder zusammengesetzte Konzepte und Rollen rekursiv durch Konstruktoren aufgebaut werden können. KL-ONE und seine Verwandten haben unsere Vorstellungen über den Umgang mit begrifflichem Wissen im Computer maßgeblich geprägt. Mit dem System KRYPTON (Brachman et al. 1983) wurden erstmals die Begriffe T-Box (für das terminologische Wissen in der gerade diskutierten formalen Darstellung) und A-Box (für die Aussagen (assertion)) über die zu modellierende Welt eingeführt.

Diese Sprechweise und das damit verbundene Verständnis, dass Fakten, welche die Welt beschreiben, vor dem Hintergrund terminologischen Wissens interpretiert und verstanden werden (und umgekehrt das terminologische Wissen als Vorlage zur Formulierung von Fakten verwendet werden kann), und dass die Verbindung zwischen diesen Bereichen durch logische Schlüsse geschieht, ist grundlegend und begegnet uns in den Semantischen Technologien in vielfältiger Weise. Der generelle Ansatz, Wissen über Begriffe und damit über die im menschlichen Denken verwendeten Konzepte in formal präziser und damit für den Computer zugänglicher Weise darzustellen, um auf dieser Grundlage dann Aussagen korrekt interpretieren zu können, ist geradezu das Fundament der Semantischen Technologien. Die Verwendung einer solchen formal

spezifizierten Konzeptmenge als Grundlage der Kommunikation (insbesondere zwischen Computern) führt direkt zum Ansatz der *Ontologien*.

2.10 Ontologien

Bereits im einleitenden Kapitel hatten wir gesehen, dass die Grundlage jeder erfolgreichen Kommunikation die Verwendung von Begriffen ist, die von Sender und Empfänger in gleicher Weise verstanden werden können. Aus der alltäglichen Kommunikation zwischen Menschen wird deutlich, welche Aufgabe zu bewältigen ist, damit ein solcher Austausch überhaupt und insbesondere mit Computern funktioniert: Jeder verwendete Begriff und jede sprachliche Äußerung bringen umfangreiches Wissen über die Realitäten der Welt, der jeweiligen Vorgeschichte, der aktuellen Situation usw. mit sich, das zum gegenseitigen korrekten Verständnis unabdingbar ist.

Die Erfassung und Beschreibung der realen Welt sowie des Wesens der uns umgebenden Dinge interessiert die Menschheit bereits seit Jahrtausenden: Die Ontologie – die Lehre vom Seienden – sucht als Disziplin der Philosophie seit jeher nach Möglichkeiten, die Grundstrukturen der Realität korrekt und allgemeingültig zu beschreiben. Freilich ist eine vollständige, allgemein akzeptierte und für alle Zwecke ausreichende Darstellung alles Seienden noch in weiter Ferne.

2.10.1 Der Ontologiebegriff in der Informatik

Aus der Sicht der Informatik und der Semantischen Technologien erlauben wir uns eine pragmatischere Herangehensweise: Um einen effektiven Austausch von Informationen, also Kommunikation, gemeinsames Verständnis und geeignete Aktionen möglich zu machen, konzentrieren wir uns auf das Schaffen einer für die jeweiligen Kommunikationsteilnehmer gemeinsamen Konzeptualisierung als Grundlage der Kommunikation. Hierzu versuchen wir, das zur erfolgreichen Kommunikation nötige Wissen über Begriffe und Zusammenhänge in einer für Computer nutzbaren, also formalen Weise zu beschreiben – wir stehen also vor einer Aufgabe der Wissensrepräsentation, zu deren Lösungen alle in diesem Kapitel beschriebenen Ansätze beitragen können.

Die am häufigsten gebrauchte Definition von Ontologie in Bezug auf die Informatik ist die von Tom Gruber (Gruber 1993):

„An ontology is an explicit specification of a conceptualization.“

Eine weitere Definition, welche die gemeinsame Nutzung von Ontologien in den Vordergrund stellt ist die folgende (Uschold und Gruninger 1996):

„An ontology is a shared understanding of some domain of interest.“

Üblicherweise werden die Nuancen beider Definitionen zusammengenommen und man sagt: *Eine Ontologie ist eine formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung*. Im Grunde genommen geht es also um:

- die Nutzung gemeinsamer Symbole und Begriffe im Sinne einer Syntax,
- das gemeinschaftliche (Ein-)Verständnis bezüglich deren Bedeutung, also der Semantik,
- die Klassifikation der Begriffe in Form einer Taxonomie,
- die Vernetzung der Begriffe mit Hilfe von assoziativen Relationen bei gleichzeitiger
- Festlegung von Regeln und Definitionen darüber, welche Relationen sinnvoll und erlaubt sind.

Eine Ontologie erfordert also eine Begriffsfindung im Sinne der Erstellung eines abstrakten Modells durch die Identifizierung der relevanten Begriffe und Beziehungen zwischen diesen. Diese Begriffe (auch Konzepte genannt) müssen gemeinsam gefunden und benutzt werden, d. h. eine Gruppe von Nutzern muss diese Begriffe akzeptieren. Dabei heißt formal, dass die Darstellung in maschinenverständlicher Form ist. (Man beachte: Diese Definitionen betonen, dass Ontologien für ein gewisses Interessensgebiet formuliert werden, also für einen begrenzten Ausschnitt der Welt und ggf. mit ausgewählten Zielsetzungen. Im Gegensatz zur Philosophie sprechen wir daher von mehreren unterschiedlichen Ontologien, je nach Domäne oder Zielsetzung!)

In der Regel bestehen Ontologien aus Komponenten, die wir im Verlaufe des Kapitels allesamt kennengelernt haben. Diese sind Klassen, Relationen und Regeln zur Beschreibung der Konzeptualisierung sowie Instanzen, die dann individuelle Elemente der Domäne bezeichnen und interpretieren:

- *Klassen* (oft auch Konzepte genannt) beschreiben die verschiedenen Begriffskategorien. Diese sind in Ontologien meistens als Taxonomien (Begriffshierarchien) organisiert, so dass die bereits betrachteten Vererbungsmechanismen anwendbar sind.
- *Relationen* beschreiben Abhängigkeiten zwischen den Klassen. In der Taxonomie ist die „ist-ein(e)“-Relation quasi standardmäßig definiert. Alle anderen Relationen müssen explizit festgelegt werden. Oft wird auch die „ist-verbunden-mit“ verwendet. Dies geschieht oft, um eine explizite – aber nicht weiter verfeinerbare Beziehung – auszudrücken. Gleichzeitig beschreiben die Relationen weitere Eigenschaften der Klassen (wie die Slots eines Frames oder die Rollen eines KL-ONE-Konzepts).
- *Regeln* (oder Axiome) werden benutzt, um Gegebenheiten in der Domäne zu beschreiben, die immer wahr sein müssen.
- *Instanzen* repräsentieren real existente Elemente einer Domäne.

Ontologien finden immer dann ihren Einsatz, wenn Daten- und Informationsbestände von mehreren Teilnehmern mit unterschiedlichem Wissen und Fähigkeiten benutzt und verändert werden. Ontologien stellen dabei Domänenwissen so dar, dass es wiederverwendbar wird und, als gemeinsam akzeptiertes Begriffslexikon, von unterschiedlichen Benutzergruppen in Anspruch genommen werden kann. Besonders interessant sind dabei Anwendungen, die auf der Basis des Internet verteilten Nutzergruppen den Zugriff auf große, sich ständig ändernde Informationswelten bieten und dabei automatisierte Dienste ermöglichen.

2.10.2 Aufbau von Ontologien

In gewisser Weise repräsentiert eine Ontologie eine gemeinsame Sprache bezüglich eines Anwendungsgebietes. Dies impliziert, dass eine Benutzergruppe im Sinne eines gemeinsamen Verständnisses von bestimmten Begriffen eine Übereinstimmung erreicht haben muss und setzt eine Angleichung möglicher Vorstellungen verschiedener Nutzer voraus. Worüber aber müssen sich die Teilnehmer einigen?

- Zunächst gilt es, die Form zu bestimmen, in der die Ontologie beschrieben werden soll. Dieser Repräsentationsformalismus kann bereits selbst als eine Ontologie verstanden werden; diese legt fest, wie Klassen, Relationen und Regeln dargestellt werden und welche Bedeutung ihnen zuzuordnen ist. In der Praxis begegnen uns hier die bereits weiter oben beschriebenen Ansätze und Formalismen. Für die Nutzbarkeit der jeweiligen Ontologiesprache sind dabei insbesondere die Mächtigkeit der Repräsentation, die möglichen Schlüsse und die Annahmen über die Vollständigkeit wichtig. Für letztere gilt es, zwischen der Annahme der vollständig beschriebenen abgeschlossenen Welt (*closed-world assumption*) einerseits und der Annahme einer unvollständig beschriebenen, offenen Welt (*open-world assumption*) zu unterscheiden. Deutlich wird dieser Unterschied, wenn ein gesuchter Fakt nicht gefunden werden kann: Unter der Annahme einer closed world bedeutet „nicht finden“ die Ungültigkeit des Fakts (da ja alles Gültige vollständig aufgeschrieben ist); in einer open world kann nur festgestellt werden, dass über diesen Fakt nichts bekannt ist. In den folgenden Kapiteln werden wir z.B. die Formalismen RDF (für einfache Datengraphen) und OWL (für Konzeptdefinitionen im Sinne der description logic) kennenlernen.
- Sodann werden die Konzepte, Relationen und Axiome modelliert. Damit wird festgelegt, mit welchen Begriffen die betrachtete Welt beschrieben wird, welche Unterscheidungen getroffen werden und welche Bedingungen und Zusammenhänge für die Elemente der betrachteten Welt gelten. Diese Festschreibung der Sprechweise einer Anwendung bzw. Nutzergruppe bezeichnet man auch als *ontological commitment*.

Aus pragmatischen Gründen ist hier eine weitere Unterscheidung nach dem Grad der Allgemeingültigkeit der Ontologie sinnvoll:

- *Upper Ontologies* beschreiben Begriffe von hoher Allgemeingültigkeit, die von vielen Anwendern in unterschiedlichen Szenarien sinnvoll eingesetzt werden können. Beispiele sind etwa Konzepte zur Beschreibung von Raum und Zeit, von physikalischen Eigenschaften oder auch von weitverbreiteten Informationselementen.
- *Domain Ontologies* beschreiben dann die spezielleren, für einzelne Anwendungsbereiche brauchbaren Konzepte. Diese enthalten typischerweise detailliertes und umfangreiches Wissen über den spezifischen Anwendungsbereich, sind aber anderweitig oft nicht sinnvoll einzusetzen.

Neben Detaillierungsgrad und Präzision der domänenspezifischen Begriffe und dem Grad der möglichen Wiederverwendung in vielfältigen Bereichen (Sharing Scope) ist schließlich noch die Stabilität einer Ontologie von großer praktischer Bedeutung: Begriffe, die von vielen Nutzern in unterschiedlichen Szenarien wiederverwendet werden, sollten sich über lange Zeit weder in ihrer Form noch in ihrer Bedeutung verändern, da sonst eine Vielzahl von Teilnehmern diese Änderungen anpassen müssen. Für hoch spezifische Ontologien in kleinen Anwendungsbereichen ist hingegen eine rasche dynamische Erweiterung oder Veränderung unproblematisch.

Betrachten wir einen Ausschnitt aus der Lebenswirklichkeit von Rudi Baispilot und versuchen, einen Ausschnitt einer für Rudi relevanten Ontologie zu skizzieren: Die Dinge, mit denen Rudi üblicherweise umgeht, umfassen etwa Dokumente, Orte (Adressen), Termine, Unternehmen, Menschen, Themen, Ereignisse und Aufgaben. Wie bereits in Abbildung 2.9 dargestellt, erfassen wir diese Konzepte etwas detaillierter und schreiben beispielsweise:

Ding → (Dokument, Ort, Termin, Unternehmen, Mensch, Thema, Ereignis, Aufgabe)

Mensch → (Erwerbstätiger → (Arbeitnehmer → (Arbeiter, Angestellter → (**Sachbearbeiter**, **Geschäftsführer**))), Selbstständiger))

Dokument → (**E-Mail**, **KFZ-Vers.Antrag**, ...)

Unternehmen → (Forschungsinst., Finanzdienstleister, ...)

Aufgabe → (**KFZ-Versicherung**, ...)

Die Klammerung beschreibt in diesem Fall die Struktur der jeweiligen Klassenbäume. Nachvollziehbar wird dies wenn man sich nochmals Abbildung 2.9 ansieht, wo die graphische Repräsentation des Klassenbaums für „Erwerbstätiger“ gezeigt wird.

Jede der verwendeten Klassen wird durch Attribute bzw. Beziehungen zu anderen Klassen weiter konkretisiert. Neben der vertikalen „ist-ein(e)“-Relation, welche die Taxonomie implizit liefert, kommen für die CarFS-Domäne etwa in Betracht:

- ist-Kunde(Unternehmen, CarFS): ein Unternehmen ist Kunde der CarFS
- arbeitet-für(Arbeitnehmer, Unternehmen): ein Arbeitnehmer arbeitet für ein Unternehmen
- bearbeitet(Sachbearbeiter, Aufgabe): ein Sachbearbeiter bearbeitet eine Aufgabe
- hat-Sitz(Unternehmen, Ort): ein Unternehmen hat seinen Sitz an einem Ort
- gesendet-von(Dokument, Mensch): ein Dokument wurde von einem Menschen gesendet

- `gesendet-an(Dokument, Mensch)`: ein Dokument wurde an einen Menschen gesendet
- `hat-Anlage(Email, Dokument)`: ein Dokument hat ein Dokument als Anlage
- `abgeschlossen(Aufgabe, Termin)`: Ein Vorgang muss zu einem Termin abgeschlossen sein

Wir könnten die Auflistung beliebig weiterführen, wollen uns jedoch auf diese Relationen beschränken.

Für unser Beispiel konzentrieren wir uns zunächst auf die oben fett gedruckten Konzepte und nutzen zur einfachen Darstellung die in Abbildung 2.17 dargestellten Symbole.

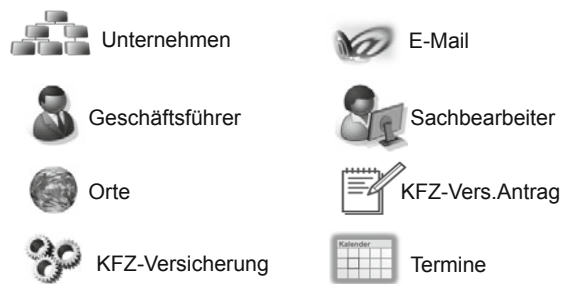


Abb. 2.17 Zu Illustrationszwecken eingeführte Klassensymbole.

Weiteres Wissen über die CarFS-Domäne erfassen wir in Regeln (Axiomen), die Aussagen über gültige Instanzen von Konzepten machen. Beispiele sind etwa

Wenn der KFZ-Vers.Antrag unterschrieben ist, dann ist die Versicherung gültig.

Um den Leasing-Rabatt zu erhalten, muss ein Unternehmen mindestens zwei Jahre Kunde sein.

Ein Geschäftsführer ist berechtigt, eine Unterschrift zu geben.

Ein Angestellter betreut mindestens zwei und höchstens fünf Kunden.

Ein Sachbearbeiter darf höchstens drei Kunden betreuen. (Da ein Sachbearbeiter ein Angestellter ist, gilt auch die ererbte Regel, dass mindestens zwei Kunden betreut werden ...)

Damit haben wir einige Aspekte der Berufswelt von Rudi Baispilov abstrakt beschrieben. Betrachten wir nun die Anwendung dieser ontologischen Modellierung im Beispiel: Nehmen wir an, Rudi schickt auf die Anfrage von Gesine hin eine E-Mail zurück, mittels der er die Anfrage beantwortet und einen entsprechenden Antrag als Attachment übersendet. Eine solche E-Mail könnte so aussehen, wie die in Abbildung 2.18 dargestellte.

Alle relevanten Fakten dieser Aktion können wir nun durch Instanzen der eingeführten ontologischen Konzepte beschreiben. Wir erhalten so ein semantisches Netz, dessen Knoten und Kanten uns bestens bekannt und mit formal definierten Regeln und Eigenschaften verbunden sind (Abbildung 2.19).

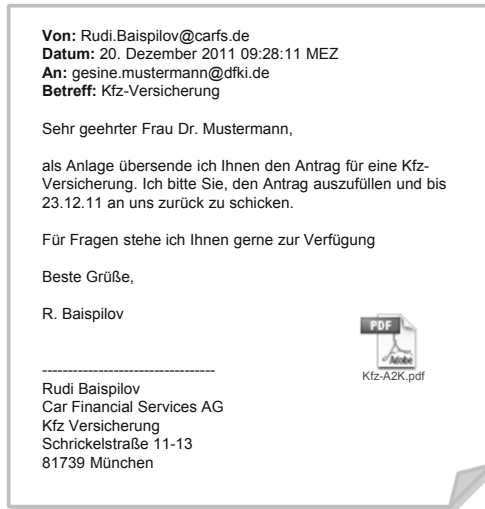


Abb. 2.18 E-Mail von Rudi an Gesine.

Damit ist sichtbar – und für den Computer zugänglich – dass etwa München einen Ort bezeichnet, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) ein Unternehmen darstellt oder die E-Mail über Sender, Empfänger und Anlage verfügt; die Interpretation dieser Elemente etwa für Suche, Vererbung, Überprüfung von Gültigkeiten oder Berechtigungen usw. ist durch die Axiome der Ontologie festgelegt.

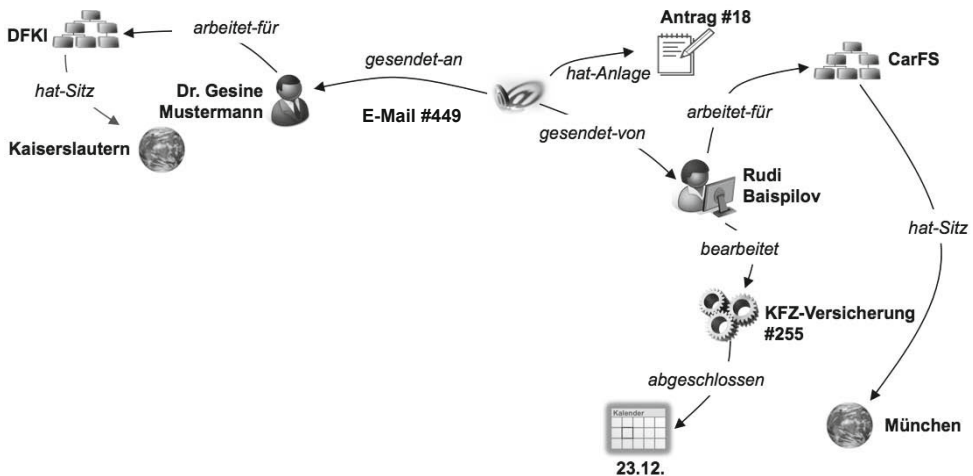


Abb. 2.19 Beschreibung der die E-Mail in Abb. 2.18 betreffenden Fakten durch Ontologie-Instanzen.

2.10.3 Nutzen von Ontologien

Die Konsequenzen dieser Modellierung und Instanziierung für die mögliche automatische Verarbeitung – und damit die Bedeutung der Wissensrepräsentation durch Ontologien als wesentliche Grundlage Semantischer Technologien – wird in den folgenden Kapiteln an zahlreichen Anwendungsbeispielen weiter verdeutlicht.

Allgemein betrachtet erfordert Entwicklung und Einsatz von Ontologien zunächst einen Modellierungsaufwand (wobei u.U. von der Wiederverwendung vorhandener Ontologien profitiert werden kann). Die Vorteile, welche die Entwicklung und den Einsatz von Ontologien rechtfertigen, liegen in erster Linie in der Unterstützung der Kommunikation zwischen Menschen, die in einer Anwendungsdomäne zusammenarbeiten, aber auch und insbesondere in der Interoperabilität zwischen Computersystemen. Diese Vorteile resultieren aus der Tatsache, dass Ontologien die den Datenstrukturen zugrunde liegenden Annahmen explizit und computerlesbar bereitstellen und so Flexibilität und Wiederverwendung auf verschiedenen Ebenen von Informationssystemen unterstützen, wie etwa:

- geringerer Aufwand bei Spezifikation und Systemrealisierung durch Wiederverwendung und ontologiegestützt strukturierte Kommunikation
- höhere Zuverlässigkeit der Modellierung (die Ontologien sind gewachsen und auf breiter Front akzeptiert)
- höhere Flexibilität der Lösung, da etwa Details von Konzepten erst zur Laufzeit nachgeschlagen werden, also beim Entwurf noch nicht bekannt sein müssen
- bessere Nutzerfreundlichkeit, z. B. durch Unterstützung von Suchfunktionen (man muss nicht das vollständige Datenbankschema kennen),
 - bei der Wartung (da die Realisierung über die Ontologie dem Servicetechniker transparent wird) und
 - bei der Erhebung von Informationen (durch einfaches Klassifizieren).

Die Definition und Verwendung von Ontologien erlaubt uns insbesondere, vielfältige Informationen im Internet mit maschinell verarbeitbaren Angaben zu ihrer Bedeutung zu versehen und die Grundlagen dieser Bedeutung wiederum anderen in formaler Weise zugänglich zu machen. Es wird also möglich, dass Nutzer, Entwickler und Anwender fremde Daten automatisch verstehen und verarbeiten können – einfach, indem die entsprechenden Konzepte in den jeweils angegebenen verwendeten Ontologien nachgeschlagen und den Regeln entsprechend verarbeitet werden. Ontologien realisieren also die Wörterbücher einer umfassenden, vielsprachigen und erdumspannenden Informationswelt und sind damit wesentliche Grundlage für ein automatisches Verständnis durch Semantische Technologien.

2.11 Fazit

Eine grundlegende Aufgabe Semantischer Technologien ist es, Wissen für den Computer zugänglich zu machen. Die Formalisierung von Wissen auf dem Computer basiert auf der Projektion eines Realitätsausschnitts in eine entsprechende semantische Repräsentation.

Die Vorschrift dieser Überführung sollte dabei den Vorgaben der allgemeinen Modelltheorie (Stachowiak 1973) folgend einige Bedingungen erfüllen:

- Sie sollte weitestgehend strukturerhaltend sein und im Verhalten dem Original ähnlich, wobei die entstehenden semantischen Modelle eine syntaktische und eine semantische Dimension umfassen. Die Abgrenzung des Realitätsausschnittes erfolgt kollaborativ und subjektiv je nach Zielsetzung durch die Entscheidung, welche der konstituierenden Objekte in das Modell übernommen werden.
- Die entstehenden semantischen Modelle sind i. d. R. weniger umfassend bzw. komplex als das Original. Sie stellen daher eine Vereinfachung dar, welche durch die inhärente Abstraktion erreicht wird.
- Die menschlichen Modellierer oder Wissensingenieure bestimmen die Inhalte, die je nach Zielsetzung, welche Aufgaben der Computer lösen soll, unterschiedlich sein können, was jedoch auch impliziert, dass die Auswahl von Eigenschaften des Realitätsausschnitts zu einem anderen Zeitpunkt anders ausfallen kann, z. B. wenn neues Wissen über die enthaltenen Anschauungsobjekte entsteht.

Die künstliche Intelligenz stellt hierzu eine Vielzahl an Formalismen zur Verfügung, die es erlauben, semantische Modelle zu erstellen und im Sinne der Wissensarbeit unterstützend einzusetzen. Angefangen von einfachen logischen Fakten und Regeln, von semantischen Zusammenhängen bis hin zu umfangreichen Plänen oder komplexen Netzwerken strukturierter Objekte lassen sich unterschiedliche Formen der Wissensdarstellung für die Konzeption und Implementierung semantischer Technologien nutzen. Allen gemeinsam ist das Prinzip der Deklarativität sowie die Anlehnung an geläufige Beschreibungen der für den Menschen wahrnehmbaren Welt: Objekte, ihre Eigenschaften und Beziehungen, Aktionen und Prozesse. Passende Inferenzmechanismen befähigen den Computer, aus solchen Beschreibungen sinnvolle Aktionen und neue Erkenntnisse abzuleiten. Einige dieser Inferenzen sind dabei nur für die konkrete Aufgabe nützlich, andere, wie die Verwaltung von Objekthierarchien mit Vererbung und Klassifikationsdiensten, begegnen uns als universelle Werkzeuge immer wieder.

2.12 Weiterführende Literatur

In der Literatur finden sich zahlreiche Veröffentlichungen rund um das Thema Wissensrepräsentation. Beispiele sind hier etwa die Bücher von Brachmann und Levesque (Brachmann und Levesque 2004) oder aber auch deutsche Veröffentlichungen wie z. B. Heinsohn und Socher-Ambrosius (Heinsohn und Socher-Ambrosius 1999). Beide Werke geben umfassende Übersichten zu den unterschiedlichen Ansätzen. Empfehlenswert ist auch ein Blick in den grundlegenden Artikel von Minsky (Minsky 1974), der die Grundideen zum Framesystem darlegt. Als Standardwerk sind sicherlich auch die Arbeiten von John Sowa (Sowa 1991) zu betrachten, die sich umfassend mit semantischen Netzen und Wissensrepräsentation auseinandersetzen. Eine umfassende Zusammenfassung zum Thema Semantik aus der Sicht der Computerlinguistik findet sich in (Carstensen et al. 2009b).

Andreas Dengel, Ansgar Bernardi, Ludger van Elst

Semantische Technologien

Grundlagen. Konzepte. Anwendungen.

Dengel, A. (Hrsg.)

2012, XVI, 458 S. 100 Abb., Hardcover

ISBN: 978-3-8274-2663-5