Vortrag zum Fortgeschrittenenpraktikum

Ein Werkzeug zur interaktiven Bearbeitung von Und-Oder-Bäumen

von Matthias Nickles

gehalten am 2.10.1997

am Institut für Informatik der Ludwig-Maximilians-Universität München

Lehr- und Forschungseinheit für Programmier- und Modellierungssprachen
(Prof. Dr. François Bry)

Inhalt

Ziel der Arbeit 2
Kategorien und Kategoriensysteme 3
Und-Oder-Bäume 3
Selektionen 5
Weitere Knotentypen 9
Textblätter 10
Nicht-exklusive Oder-Knoten 10
Kanten und Attribute 11
Constraints 12
Benutzermodell 14

Ziel der Arbeit

Das Fortgeschrittenenpraktikum hatte das Ziel, ein interaktives Werkzeug zur Erstellung, Bearbeitung und Visualisierung von *Kategoriensystemen* und *Kategorien* zu entwickeln. Dieses Tool sollte der Beschreibung *natürlicher Sprachen* dienen (ohne aber auf dieses Anwendungsgebiet beschränkt zu sein). Kategoriensysteme sollten durch sogenannte *Und-Oder-Bäume* repräsentiert werden, deren besondere Eignung zur Kategorisierung von Sprachen anhand von linguistischen Charakteristika sich erwiesen hat ([1]). Das zu erstellende Werkzeug sollte es hierzu erlauben, *Entscheidungsprozesse* zur Auswahl einzelner Kategorien in einer interaktiven, graphischen Umgebung mit Hilfe solcher Bäume zu modellieren und durchzuführen.

Kategorien und Kategoriensysteme

Eine Kategorie läßt sich als eine Klasse von Merkmalen auffassen, die einen bestimmten Gegenstand charakterisieren, ein Kategoriensystem als Klasse zusammengehöriger Kategorien. Um eine Beschreibung eines Gegenstandes durchführen zu können, wird ein passendes Kategoriensystem vorgegeben. Die Aufgabe des Beschreibers ist es dann, aus diesem Kategoriensystem eine geeignete Kategorie auszuwählen, der sich der Gegenstand der Beschreibung zuordnen läßt. Diese Kategorie dient dann als formale Repräsentation der charakterisierten Sache und kann so z.B. in einer Datenbank abgelegt, maschinell ausgewertet werden o.ä.. Beispielsweise wird ein Auto vom Typ Golf von der Kategorie "PKW mit Ottomotor" charakterisiert.

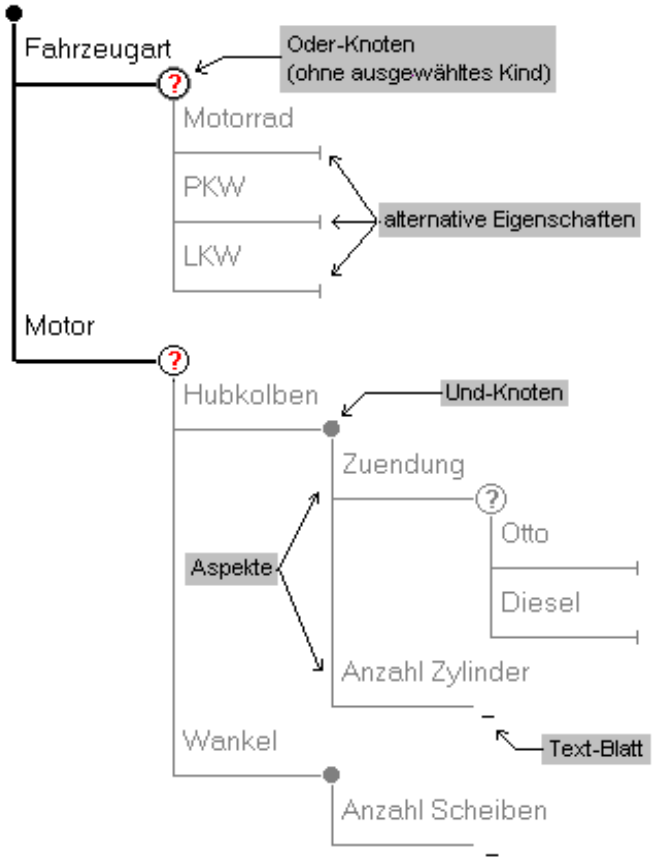
Und-Oder-Bäume

Wie kann ein Kategoriensystem modelliert und dargestellt werden? Kategoriensysteme sollten möglichst wenig redundant sein und leicht zu warten (d.h. Änderungen und Hinzufügungen sollten mit möglichst geringem Aufwand verbunden sein), sowie vor allem den Beschreibungsprozeß, d.h. die Auswahl einer geeigneten Kategorie, unterstützen. Natürlich hängt die Realisierung dieser Anforderungen auch vom zu untersuchenden Gegenstand ab.

Für die Beschreibung von natürlichen Sprachen hat sich die Verwendung von sogenannten Und-Oder-*Bäumen* als sinnvoll erwiesen ([1]). Das besondere Merkmal eines solchen Baums ist, daß alle inneren Knoten entweder *Und-Knoten* (ramifications) oder (exklusive) *Oder-Knoten* (decision nodes) sind. Die im Praktikum implementierte Variante dieses Baumtyps erlaubt darüber hinaus noch weitere Knotentypen, siehe 1.1.5.

Ein solcher Baum repräsentiert das Kategoriensystem: Aspekte und Eigenschaften der zu beschreibenden Gegenstände werden durch die Kanten bzw. Knoten dargestellt (die Begriffe "Kante" und "Knoten" (für den der Kante nachfolgenden Knoten) werden meist synonym verwendet, vgl. 1.1.6). *Aspekte* ordnet man als Kinder von Und-Knoten an, *Eigenschaften* (die möglichen *alternativen* Ausprägungen eines Aspekts) dagegen als Kinder von Oder-Knoten. Eigenschaften und Aspekte, die nur relevant sind, wenn in den Kategorien, in denen sie vorkommen können, auch bestimmte andere Merkmale enthalten sind, werden im selben Teilbaum als Nachkommen dieser Merkmale angeordnet.

Beispiel: Kategoriensystem "Fahrzeuge" (motorisiert) als Und-Oder-Baum



Jedes Fahrzeug besitzt die Aspekte "Fahrzeugart" *und* "Motor", die *beide* spezifiziert werden müssen. Die Fahrzeugart ist entweder "Motorrad" *oder* "PKW" *oder* "LKW", der Motor entweder vom Typ "Hubkolben" *oder* vom Typ "Wankel". Ein Hubkolben-Motor wird durch seine Aspekte "Zündung" *und* "Anzahl Zylinder" festgelegt, die Zündung wiederum erfolgt nach Art eines "Otto"- *oder* "Diesel"-Hubkolbenmotors (d.h. sie wird durch Zündkerzen *oder* Verdichtung ausgelöst). Ein "Wankel"-Motor wird hier dagegen nur durch seine "Anzahl Scheiben" näher beschrieben.

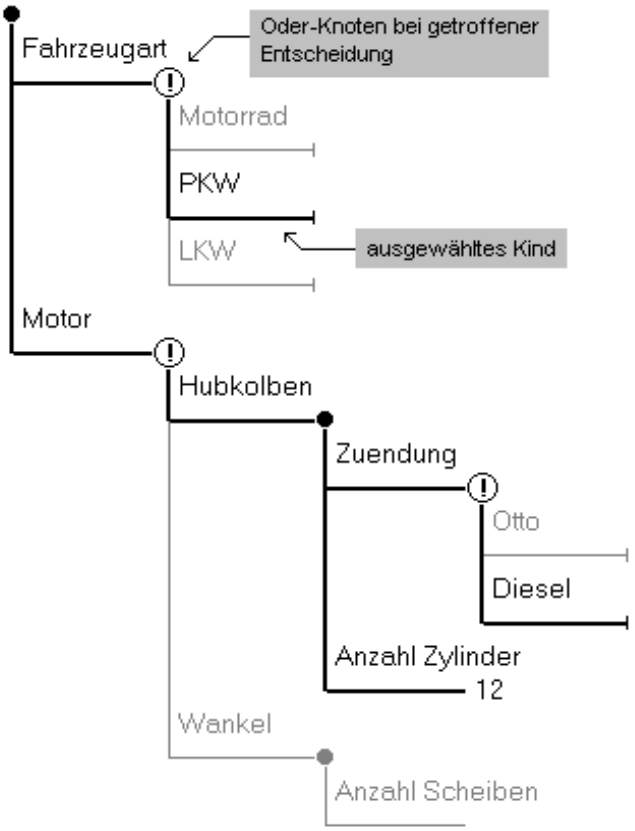
Selektionen

Um mit einem Und-Oder-Baum einen Gegenstand zu beschreiben, müssen passende zusammenhängende Kanten ausgewählt werden, angefangen bei der Wurzel: Wie man gesehen hat, verbinden Oder-Knoten *alternative Eigenschaften* der zu beschreibenden Gegenstände. Bereits ausgewählte Oder-Knoten repräsentieren somit *Entscheidungs-Orte*, an denen jeweils *eines* der Kinder des Knotens auszuwählen ist. Dagegen müssen bei einem bereits ausgewählten Und-Knoten, der ja *gleichzeitig* relevante Aspekte verbindet, auch jeweils *alle* Kinder ausgewählt werden. Eine Entscheidung muß hier also nicht getroffen werden. Somit kann der gesamte Beschreibungsvorgang als eine Folge von getroffenen *Entscheidungen* bei Oder-Knoten verstanden werden. Der Beschreiber folgt dabei dem hierarchischen Aufbau des Baums (beginnend beim Wurzelknoten), weil Entscheidungen ggfs. erst dann relevant werden, wenn andere Entscheidungen, die im selben Teilbaum näher bei der Wurzel liegen, bereits getroffen wurden.
Um auf diese Weise anhand des Beispielbaums "Fahrzeuge" ein konkretes Fahrzeug zu charakterisieren (d.h. eine geeignete Kategorie festzulegen, der das Fahrzeug zugeordnet werden kann), folgt man bei der Auswahl von Knoten der hierarchischen Aufeinanderfolge der möglichen Entscheidungen von der Wurzel zu den Blättern: Bei Und-Knoten sind *alle* ausgehenden Kanten weiterzuverfolgen; "Fahrzeugart" und "Motor" sind also Aspekte, die *beide* beschrieben werden müssen. "Hubkolben" und "Wankel" sind dagegen nicht miteinander vereinbar, ebensowenig wie z.B. "Motorrad" und "PKW". Hier ist jeweils eine Entscheidungen über das weitere Vorgehen notwendig. Die Beschreibung der Eigenschaften des Aspektes "Zündung" (also "Otto" oder "Diesel" usw.) ist nur relevant, wenn als Motor-Typ bereits "Hubkolben" gewählt wurde. Dagegen kann die Entscheidung für eine Fahrzeugart (PKW, Motorrad oder LKW) unabhängig von der Wahl des Motortyps getroffen werden (zu Abhängigkeiten, die sich nicht direkt aus der Baumstruktur ergeben, siehe 1.1.7).

Möchte man nun z.B. einen Personenwagen mit einem 12 Zylinder-Dieselmotor beschreiben, so folgt man den vom Wurzelknoten ausgehenden Kantenzügen in Richtung der Blätter: Da der Wurzelknoten ein Und-Knoten ist, also Aspekte

verbindet, nimmt man seine beiden Kinder "Fahrzeugart" und "Motor" in die Definition der Kategorie auf. "Fahrzeugart" wird durch einen Oder-Knoten repräsentiert, weshalb hier nur dessen Kind "PKW" als konkrete Ausprägung des Aspekts "Fahrzeugart" aufgenommen wird. Beim Oder-Knoten "Motor" entscheidet man sich für das Kind "Hubkolben". Dieser Knoten ist ein Und-Knoten, was dazu führt, daß seine Kinder "Zündung" und "Anzahl Zylinder" *beide* ausgewählt und ihre Äste weiterverfolgt werden müssen. Der Aspekt "Zündung" wird nun noch durch die Eigenschaft "Diesel" näher spezifiziert. Zuletzt wird das Textblatt "Anzahl Zylinder", um eine konkrete Zahlenangabe ("12") erweitert, ausgewählt (Textblätter werden unter 1.1.5 erläutert).

Graphisch wird hier diese Kategorie innerhalb des Beispielbaums "Fahrzeuge" in Form von kräftigen schwarzen Linien und Symbolen dargestellt:



Formal ist ein derartiger Beschreibungsprozeß die Zusammenstellung einer geeigneten Teilmenge von Knoten des Und-Oder-Baums unter der Berücksichtigung der bereits angedeuteten und im folgenden eindeutig festlegenden Regeln. Eine solche Teilmenge heißt *Selektion* (in der Software-Dokumentation wird der Begriff auch für die konkrete textuelle Darstellung einer solchen Teilmenge verwendet, vgl. 1.2.2.3) und repräsentiert mindestens eine Kategorie aus dem Kategoriensystem. Selektionen sind durch folgende Eigenschaften gekennzeichnet:

- der Wurzelknoten ist enthalten

- ist ein Und-Knoten enthalten, so auch alle seine Kinder

- ist ein Oder-Knoten enthalten, so ist maximal eines seiner Kinder enthalten.

- ist ein Knoten nicht enthalten, so auch keines seiner Kinder

Falls zu jedem Oder-Knoten in der Selektion genau ein Kind enthalten ist, spricht man von einer *vollständigen* Selektion (oder einfach nur von "Selektion"). Ist dagegen mindestens zu einem Oder-Knoten kein Kind in der Selektion enthalten, so nennt man die Selektion *partiell* oder *unvollständig.*

Man erkennt, daß sich zwei Selektionen zum selben Baum überhaupt nur durch das Vorhandensein von Oder-Knoten unterscheiden können (falls man den Inhalt von Textblättern nicht berücksichtigt, siehe 1.1.5).
Eine vollständige Selektion repräsentiert genau eine Kategorie, einer partielle Selektion eine Menge von Kategorien. Partielle Selektionen sind z.B. nützlich, um mehrere Kategorien mittels einer einzigen Selektion "abgekürzt" darzustellen (vgl. 2.1.6) oder um bei einer Anfrage (query) an eine Selektions-Datenbank die Ergänzungsmöglichkeiten der partiellen Selektion hin zu mehreren vollständigen Selektionen zu erfragen. *Minimale Selektion* zu einem bestimmten Baum nennt man die Selektion mit der geringstmöglichen Anzahl von Knoten.
Die Kategorie zum obigen Beispiel (PKW mit Dieselmotor) würde als (vollständige) Selektion in Mengennotation z.B. so aussehen: { "Wurzelknoten" "Fahrzeugart" "PKW" "Motor" "Hubkolben" "Zündung" "Diesel" "Anzahl Zylinder=12" }.

Die grundlegenden Modellierungsregeln bei der Baumerstellung wurden bereits erwähnt. Da sie sich auch direkt aus den Eigenschaften von Selektionen ergeben, werden sie an dieser Stelle noch einmal zusammengefaßt dargestellt:

Ist die Auswahl einer Kante B nur sinnvoll, wenn bereits eine Kante A ausgewählt wurde, so ordnet man B in dem Teilbaum an, zu dessen Wurzel A führt. Betrachtet man nur die Oder-Knoten, erhält man auf analoge Weise die *Hierarchie der Entscheidungen:* Wurde ein Kind des Knotens A ausgewählt, ermöglicht dies nachfolgend die Auswahl eines Kindes des im Teilbaum unterhalb von A angeordneten Oder-Knotens B. Beispielsweiseist beim Baum "Fahrzeuge" der Aspekt "Zündung" nur relevant, wenn als Motortyp vorher "Hubkolben" gewählt wurde.
Schließt sich die gemeinsame Auswahl zweier Kanten aus, so sollte der kleinste Teilbaum, der beide Kanten enthält, einen Oder-Knoten als Wurzel haben. Z.B. sind im "Fahrzeuge"-Baum die Fahrzeugarten "Motorrad" und "PKW" nicht gleichzeitig wählbar. Weitere Einschränkungen bei der Auswahl sind mit Hilfe von *Constraints* erzielbar, siehe 1.1.7. Ist die Auswahl aber nicht exklusiv, *und* besteht die Verbindung zwischen den beiden Kanten nicht nur aus Und-Knoten, so können sie unabhängig voneinander selektiert werden. Die "Fahrzeugart" wird so z.B. unabhängig vom "Motor"-Typ beschreibbar.
Um den Baum abzuschließen, werden *Textblätter* (siehe 1.1.5) oder *Und-Knoten ohne Kinder* (sogenannte *endings*) verwendet. Aus Sicht des Kategorien-Beschreibers stellen diese Blätter den Endpunkt des Entscheidungsprozesses dar.

Weitere Knotentypen

Neben den erwähnten Und- und Oder-Knoten können Bäume, wie sie im Praktikum betrachtet wurden, auch noch *Textblätter* (text leaves) und *nicht-exklusive Oder-Knoten* (multi-decision nodes) enthalten.

Textblätter

Textblätter (text leaves) sind Blätter, denen ein *Inhalt* in Form einer Zeichenkette zugeordnet werden kann (also “plain text“, dieses Prinzip kann aber problemlos auf andere Inhaltstypen wie Bilder, HTML-Texte oder HTML-Formulare erweitert werden, was bei der Weiterentwicklung der Software im Anschluß an das Praktikum auch geschehen ist). Textblätter repräsentieren mittels dieser Zeichenkette eine Entscheidung. Anders als bei Oder-Knoten hat man hier aber keine vorgegebene Reihe von Alternativen, sondern wählt den *Inhalt* des Textblattes durch die Aufnahme des Knotens in die Selektion. Man kann dies als eine nähere Spezifikation des dem Blatt zugeordneten Aspektes des zu beschreibenden Gegenstandes sehen. Ansonsten gelten bezüglich der Aufnahme von Textblättern in eine Selektion die selben Regeln wie für jeden beliebigen anderen Blatt-Typ. Um zu einer vorgegebenen Baumstruktur verschiedene Textinhalte zu gestatten, wird im Zuge der Erstellung einer Selektion dieses Attribut eingegeben bzw. verändert. Im oben angegebenen Beispielbaum "Fahrzeuge" sind die Blätter nach den mit "Anzahl Zylinder" und "Anzahl Scheiben" gekennzeichneten Kanten Textblätter. Hier ist bei der Selektion des jeweiligen Blattes eine Zahl einzutragen, die als Eigenschaft (Anzahl der Zylinder bzw. Scheiben) des entsprechenden Motortyps aufzufassen ist. Das Textblatt repräsentiert also vor der Zuordnung einer konkreten Zahl die Menge der natürlichen Zahlen, aus der heraus man sich dann für *ein* Element zu entscheiden hat.

Nicht-exklusive Oder-Knoten

Nicht-exklusive Oder-Knoten (multi-decision nodes) entsprechen Oder-Knoten, ermöglichen aber die Auswahl von *mehreren* Kindern des Knotens. Wurde ein solcher Knoten ausgewählt, dürfen *beliebig viele* seiner Kinder in die Selektion übernommen werden. Wird keines der Kinder ausgewählt, gilt die Selektion als unvollständig (in diesem Punkt verhalten sich exklusive und nicht-exklusive Oder-Knoten also gleich). Bei einem solchen Knoten handelt es sich somit einfach um eine Abkürzung für die Knotenkombination zur Modellierung unabhängiger Entscheidungen (vgl. 1.1.4): Ein nicht-exklusiver Oder-Knoten mit den Kindern $K_{1..n}$ entspricht bzgl. der Selektionsmöglichkeit von $K_{1..n}$ einem Und-Knoten, der als Kinder n Oder-Knoten mit jeweils einem Kind K_i (i=1..n) besitzt. Die genannten Regeln zur Erstellung von Selektionen müssen für nicht-exklusive Oder-Knoten nicht erweitert werden.

Kanten und Attribute

Obwohl eine Kante genaugenommen die Verbindung zweier Knoten ist, kann jedem Knoten mit Ausnahme des Wurzelknotens genau eine hinführende Kante zugeordnet werden, die mit dem Knoten zusammen als ein einziges *Objekt* angesehen werden kann. Die Begriffe "Knoten" und "Kante" werden daher auch oft synonym verwendet; erst bei der graphischen Darstellung eines Baumes ist eine Differenzierung sinnvoll und notwendig. Die Kanten besitzen als Attribut eine Bezeichnung (*Label*) und einen Identifikator. Während das Label der Nennung des durch die Kante repräsentierten Merkmals (d.h. Aspekt bzw. Eigenschaft) dient, ist letzterer für die eindeutige textuelle Repräsentierbarkeit von Selektionen bei der automatischen Verarbeitung wichtig. Näheres hierzu siehe in der Quellcode-Dokumentation unter 1.2.2.3. Labels werden bei der graphischen Baumdarstellung zur Beschriftung der Kanten verwendet. Daher stellt sich für den Beschreiber die Erstellung einer Selektion subjektiv als Auswahl ("*Aktivierung"*) von Kanten dar.
Sowohl mit Knoten als auch mit Kanten kann optional ein Hilfetext verknüpft werden. Er dient vor allem zur Beschreibung der entsprechenden Eigenschaft bzw. des Aspektes, die bzw. der durch die Kante repräsentiert wird, fungiert also bei Kanten als

ausführliche Variante des Labels und bei Knoten als Beschreibung der Entscheidungsmöglichkeiten, die mit dem Knoten verbunden sind (dies macht natürlich nur bei Oder-Knoten und nicht-exklusiven Oder-Knoten Sinn). Weiterhin kann jedem Knoten optional ein Dateiname zugeordnet werden, was ihn zu einem sogenannten *Include-Knoten* macht. Teilbäume, die solche Knoten als Wurzeln haben, werden in jeweils einer eigenen Datei gespeichert. Näheres zu den Attributen ist unter 1.2.2 zu finden.

Constraints

Wie man gesehen hat, unterliegt die Aufnahme von Knoten in eine Selektion gewissen Zwängen (*constraints*): Das Vorhandensein von Und-Knoten kann als Verbot gesehen werden, einen Knoten auszuwählen, seine Geschwister aber nicht. Oder-Knoten verbieten dagegen umgekehrt die gleichzeitige Auswahl von Geschwisterknoten. Es kann nun vorkommen, daß diese Bedingungen nicht ausreichen, d.h. daß Entscheidungen über die Auswahl von Knoten gemäß den Selektionsregeln als voneinander unabhängig gelten, obwohl dies nach Ermessen des Baum-Modellierers nicht der Fall ist (weil ihre Kombination sinnlos oder unmöglich wäre und zu einer nichtexistenten Kategorie führte). Im Beispielbaum "Fahrzeuge" tritt dieser Fall bei der gleichzeitigen Auswahl der Fahrzeugart "Motorrad" und des Zündungs-Typs "Diesel" ein. Eine Selektion, in der beide Entscheidungen zusammen getroffen sind, macht keinen Sinn, da es keine Motorräder mit Dieselmotor gibt. Hier ist ein zusätzlicher *Constraint* notwendig, der eine derartige Selektion verbietet (oder zumindest vor ihr warnt). Liegt eine verbotene Selektion vor, so gilt der jeweilige Constraint als nicht erfüllt ("*verletzt*"). Bei der Formulierung solcher Regeln will man möglichst flexibel sein, d.h. man will nicht nur Selektionen *verbieten*, sondern auch (partielle oder vollständige) Selektionen *fordern* (ggfs. in Abhängigkeit von einer Vorbedingung). Hier bietet sich das "Wenn-Dann"-Schema an ("*wenn* Zustand X gilt, *dann* muß auch Zustand Y gelten"). Eine derartige Regel könnte für das oben genannten Beispiel so aussehen (informell): "*Wenn* in einer Selektion 'Motorrad' vorkommt, *dann* muß in der Selektion auch (als Zündungs-Typ) 'Otto' *oder* (als Motor-Typ) 'Wankel' enthalten sein, andernfalls gilt der Constraint als verletzt".

Denkbar wäre auch folgende kürzere Alternative: "*Wenn* in einer Selektion 'Motorrad' *und* 'Diesel' enthalten sind, *dann* muß darin auch die Kante 'Fehler' vorkommen": Da es die Kante 'Fehler' im Baum nicht gibt, gilt gemäß dieser Regel der Constraint immer als verletzt, wenn 'Motorrad' und 'Diesel' gemeinsam selektiert sind. Derartige "Wenn-Dann"-Aussagen über selektierte Kanten lassen sich schematisieren, wobei man sich dabei auf Selektionen beschränkt statt mit allgemeinen Kanten-Mengen zu arbeiten (eine echte Einschränkung liegt nicht vor, da Kanten sowieso nur als Elemente einer Selektion gemeinsam selektiert werden können): Teilaussagen der Form "in der Selektion s kommen die Kanten x*und* y *und* ... vor" kann man als "in der Selektion s ist als Teilmenge eine Selektion a mit den Elementen x, y, ... enthalten" schreiben. Teilaussagen der Form "in der Selektion s müssen (x_1 *und* y_1 *und* ...) *oder* (x_2 *und* y_2 *und* ...) *oder* ... vorkommen" werden analog zu "in der Selektion s ist als Teilmenge eine Selektion b_1 mit den Elementen x_1, y_1,... enthalten *oder* eine Selektion b_2 mit den Elementen x_2, y_2,... *oder* ... ".

Solche Teilaussagen werden nun folgendermaßen formalisiert und mittels Wenn-Dann-Beziehung verknüpft: Ein Constraint ist eine logische Formel der Form

$A => B_1 v ... v B_n$, deren Wahrheitswert angibt, ob der Constraint bei Vorliegen einer konkreten Selektion s erfüllt ist (d.h. "wahr" ist im Sinne von "er wird von der Selektion s nicht verletzt") oder nicht ("die Selektion s ist verboten"). A und $B_{1..n}$ sind dabei Aussagen über partielle oder vollständige Selektionen a, $b_{1..n}$. Eine solche Aussage ist genau dann "wahr", wenn die zugehörige Selektion eine Teilmenge von s ist. Somit gilt: Der Constraint ist genau dann erfüllt, wenn A nicht gilt *oder* mindestens eine Aussage B_k (k aus 1..n) schon. Die allgemeine Klauselform $A_1 ^ ... ^ A_k => B_1 v ... v B_n$ ist nicht erforderlich, denn die Selektionen der linken Seite der Implikation lassen sich zu einer einzigen Selektion a zusammenfassen (die Vereinigung der $a_{1..k}$).

Bei den Selektionen a, $b_{1..n}$ darf es sich auch um Selektionen handeln, die nicht aus dem Baum erstellt werden können, aus dem s selektiert wird. Eine solche Selektion heißt *unmöglich*, die zugehörige Aussage ist daher immer "unwahr". Um beispielsweise eine bestimmte Selektion s zu verbieten, würde man a=s setzen, n=1, und b_1 gleich einer *unmöglichen Selektion* ({ "Fehler" } im obigen Beispiel ist eine solche). Soll dagegen eine Selektion s immer einer der Selektionen $b_{1...n}$ entsprechen,

so wird man für a die *minimale Selektion* wählen (vgl. 1.1.4).
Da die Selektionen a, $b_{1..n}$ partiell sein dürfen, lassen sich mit ihnen auch Kategorien-*Mengen* beschreiben. Partielle Selektionen im Zusammenhang mit Constraints nennt man auch *Selektionsmuster,* da bei der Auswertung des jeweiligen Constraints geprüft wird, ob die konkrete Selektion s zu einem solchen Selektionsmuster "paßt" (pattern matching). Zusammen mit einem Baum können außerdem beliebig viele Constraints angegeben werden, die bei der Selektionserstellung gleichzeitig erfüllt sein müssen.

Benutzermodell und praktische Anwendung

Die Modellierung von Kategoriensystemen in Form von Bäumen und das Auswählen von Kategorien (Selektionen) sind zwei grundverschiedene Vorgänge, die üblicherweise nicht nur in getrennten Arbeitsschritten, sondern auch von unterschiedlichen Benutzergruppen durchgeführt werden. Die Erstellung von Selektionen ist Aufgabe der *Beschreiber* (*User*), während Bäume nur von einem *Superuser* modelliert und administriert werden können und dürfen. Eine Überschneidung zwischen den Aufgaben dieser Gruppen tritt allerdings gewissermaßen beim Zuordnen des Inhalts von Textblätter auf. Dies kann sowohl als Teil der Baumerstellung verstanden werden (der Superuser ordnet bestimmten Textblättern geeignete Default-Werte zu), als auch des Entscheidungsprozesses (vgl. 1.1.5).
In der Praktikums-Software ist die Trennung zwischen User und Superuser nur durch die entsprechende Vergabe von Zugriffsrechten geregelt (vgl. 2.2.1). Der Superuser arbeitet dadurch prinzipiell mit derselben Bedienungsoberfläche wie der User, hat aber mehr Funktionen zur Verfügung; insbesondere ist nur ihm die Erstellung von Bäumen erlaubt. Es ist allerdings vorgesehen, daß der User Vorschläge zur Baummodellierung machen kann, die dann vom Superuser geprüft und ggfs. umgesetzt werden (unterstützt von der Praktikums-Software, vgl. 2.5). Dies ist insbesondere dann denkbar, wenn der User im Baum keine passende Kategorie für den Beschreibungs-Gegenstand findet.

Literatur

[1] Heribert Schütz, Dietmar Zaefferer: *Eine linguistische Wissensbank,* Institut für Informatik der Universität München (PMS-FB-1996-17), 1996. URL: http://www.informatik.uni-muenchen.de/publikationen/#PMS-FB-1996-17

[2] David Flanagan: *Java in a Nutshell*, O'Reilly&Associates, 1996. URL: http://www.ora.com/catalog/books/javanut

[3] Stefan Münz: *HTML-Dateien selbst erstellen*, 1997. URL: http://www.netzwelt.com/selfhtml

[4] Gerhard Hingerl: *Eine deduktive Datenbank für die Linguistik,* Institut für Informatik der Universität München (Diplomarbeit), 1996. URL: http://www.pms.informatik.uni-muenchen.de/publikationen/diplomarbeiten/Gerhard.Hingerl/ausarbeitung.ps.gz