

Hauptseminar Informatik der Systeme

Thema:

Benutzermodellierung in natürlichsprachlichen Systemen

Planerkennung

Mika, Adalbert¹

¹ Email: a.mika@uni-duisburg-essen.de

Inhaltsangabe

- 1 Motivation
- 2 Einleitung
- 3 Plan Hierarchien
- 4 Methoden zur Bewertung von Hypothesen
 - 4.1 Der erste Ansatz (Kautz's formale Theorie)
 - 4.2 Bayes'sche Netzwerke
 - 4.3 Dempster-Shafer-Theorie
 - 4.4 Zielgraphen (Goal Graphs)
- 5 Schlusswort
- 6 Literaturverzeichnis
- 7 Bildnachweis

1 Motivation

Mit zunehmender Technologisierung in allen Bereichen des Lebens nehmen Computersysteme immer wichtigere Positionen ein und lösen mittlerweile manchmal sogar einige Berufe ab, die üblicherweise nur von Menschen ausgeübt werden. Vor allem die Domäne des Servicebereichs in der es auf Kommunikation ankommt, z.B. wenn ein Kunde eine Hotline anruft, um sich über ein Produkt zu informieren, etc.

Mit wachsender Leistungsfähigkeit moderner Computersysteme wird nun auch dieser Bereich so allmählich durch intelligente und performante Dialogsysteme pioniert. Systeme, die einen Diskurs analysieren und sich auf die Benutzer einstellen können. Eine weitere Hilfe im Umgang mit Computersystemen wäre nun z.B. eine „Führung“ von Benutzern durch ihre Aufgabe bis hin zum angestrebten Ziel. Ferner noch, die relativ frühe Erkennung der Intention um möglichst angepasste Hinführung zu ermöglichen.

2 Einleitung

Ein Plan ist eine (partiell) geordnete Menge von Planschritten, die Exemplare von Planoperatoren sind. Diese haben eine Vorbedingung und einen Effekt bei Operatorabstraktion kann ihnen eine Dekomposition zugeordnet werden.

D.h. im Klartext, dass ein Plan im Grunde nur eine geordnete (und möglichst minimale) Menge von Aktionen zur Erreichung eines bestimmten Zieles ist.

Wenn von einem Planerkennungssystem die Rede ist, wird ein entsprechendes System durch eine in der Regel recht große Anzahl an Plänen (sprich Planbibliothek) versuchen, das aktuelle Ziel des Agenten herauszufinden. Obwohl diese Planbibliothek nur Pläne aus der *Domäne*² für die das System konzipiert wurde beinhaltet, ist die Ermittlung des Ziels durch die vollständige Traversalion des Suchraums nicht praktikabel. Der trivialste Fall der Planerkennung wäre natürlich eine Planbibliothek mit nur einem Plan für nur ein Ziel, jedoch wird in der Praxis ein System erst dann flexibler, wenn es möglichst viele Pläne beinhaltet. Die dann durch Beobachtung von Agenten-Aktionen herleitbare Plan-Hypothesen lassen das System auf die Intentionen des Benutzers schließen. Dies kann - mit gewissen Einschränkungen allerdings - schon als eine Art von *Verstehen* angesehen werden. Damit ist die „Bewertung von Beobachtungen anhand bekannter Zusammenhänge“ und das Wissen (Ziel ist die Ermittlung des Inhalts einer Äußerung)³ gemeint. Das Wissen über alle in Betracht zu ziehenden Intentionshypothesen ermöglicht dem Rechner, den Benutzer zu verstehen ohne auf eine exakte Rekonstruktion der eigentlichen Aktionen angewiesen zu sein (die hierzu nötige Wissensrepräsentation ist Aufgabe der Intensionsmodelle, die durch Bayes'sche Netze realisiert werden können; siehe dazu Kapitel 3.2.2). Model hierfür ist der Mensch mit seiner Fähigkeit, mittels Transferdenken und dem Wissen um das aktuelle Gesprächsthema, auf den Inhalt einer unvollständigen Aussage zu schließen.

Erste Arbeiten zum Thema Planerkennung kamen von Cohen, Perault, und Allen, später auch Kautz, Carberry, Lesh und neuerdings Bauer.

Lambert (siehe [L91]) schlägt eine robuste maschinelle Interpretation von Diskussionen und Dialogen vor, in der nach jeder Aussage eines Gesprächsteilnehmers die Intentionen der Gesprächspartner abgeschätzt wurden, um somit die Thesen einer

² Eine Domäne ist ein Diskursthema bzw. ein Gebiet von Thematiken die das beobachtete Problem umfassen

³ Siehe dazu auch [H03]

Diskussion zu ermitteln. Dabei wird der mentale Zustand bzw. die Intention des Agenten derart modellieren, dass eine Meinungs-/Absichtsänderung im Laufe des Dialogs erkannt werden kann.

Ein weiterer Ansatz zur Planerkennung ist das VERMOBIL-System⁴ (im Rahmen des BMBF-Projekts). Es hat eine maschinelle Dolmetscherfunktionalität für natürlichsprachliche Terminvereinbarungen und verwendet eine intentionsbasierte Betrachtungsweise, da natürlichsprachliche und spontane Sprache eher zu Satzfragmenten führt als zu syntaktisch-semantisch korrekten Sätzen.

⁴ Das Nachfolgerprojekt ist SMARTKOM, mit dem Schwerpunkt eines multiodalen Mensch-Maschine-Dialogs.

3 Plan-Hierarchien

Eine Planerkennung wird in der Regel nur dann möglich, wenn sowohl für den Betrachter als auch für den Betrachteten angenommen wird, dass beide einen Teil des Wissens über mögliche Pläne gemeinsam haben. Ferner wird zusätzlich angenommen, dass der Agent rational handelt, d.h., dass er auch wirklich einen Plan verfolgt⁵.

Eine grundlegende Rolle in der Planerkennung spielt daher die sogenannte *Plan Hierarchie*. Sie repräsentieren sozusagen die (gemeinsame) Basis bzw. die Einschränkung aller möglichen Pläne eines Agenten und des Planerkennters in der betrachteten Domäne⁶. Das hier verwendete Schema modelliert übrigens keine temporalen Relationen, d.h. es werden keine Informationen über das zeitliche auftreten von Events repräsentiert.

Definition:

$$\mathcal{H} = \langle \mathcal{P}, Dec, Ab \rangle \quad (A1)$$

mit $\mathcal{P}$ der Menge aller (*Basis*)Pläne dieser Hierarchie, *Dec* Dekompositions- und *Ab* Abstraktions-Hierarchien.

Dec besteht aus einer Menge aus Plan-Dekompositions-Tupeln der Form $\langle p, Act(p), C(p) \rangle$ mit $p \in \mathcal{P}$ (der Plan), $Act(p)$ die Menge der Aktionen (entweder basis oder abstrakt) und $C(p)$ eine Menge von Einschränkungen (z.B. Bedingungen über temporale Beziehungen zwischen verschiedenen Elementen, etc.). Die *Dec*-Hierarchie enthält die komplette Information wie ein Plan ausgeführt werden muss.

Die *Ab*-Hierarchie teilt sich wiederum in Ab_p und Ab_a , wobei Ab_p aus Paaren von $\langle p_1, p_2 \rangle$ mit $p_1, p_2 \in \mathcal{P}$ besteht, mit der Aussage, dass (der Plan) p_1 von (dem Plan) p_2 abstrahiert wird. Analog dazu lässt sich Ab_a definieren, mit $\langle a_1, a_2 \rangle$ mit $a_1, a_2 \in A(p)$.

Wenn ein Plan bzw. eine Aktion Abstraktionen besitzt jedoch selbst keine solche ist, dann spricht man von einem sogenannten *Basis-Plan*, bzw. *Basis-Aktion*.

⁵ Letztere Annahme ist natürlich obligatorisch für jeden Planerkenner (s. auch [W02]).

⁶ Die Modellierung eines Systems für generische Planerkennung ist aufgrund der Komplexität aller möglichen Pläne aller möglichen Domänen z.Z. kaum realisierbar, deswegen ist jeder Planerkenner auch nur in seiner Systemumgebung anwendbar.

Beispiel am Fall eines konkreten Systems (nach Bauer, [B95]):

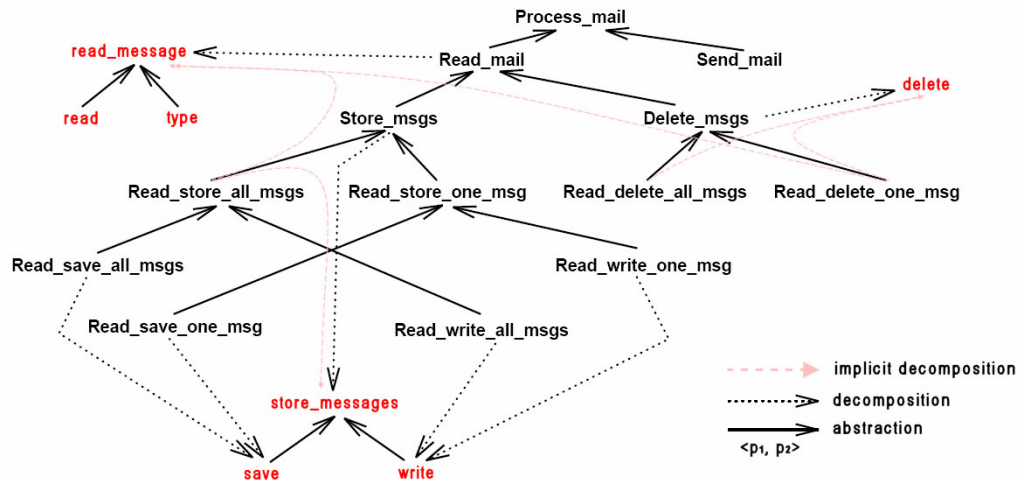


Abb. 3.a: Beispiel Plan-Hierarchie

Das Repräsentations-Schema der Plan-Hierarchien ist sehr mächtig zur Modellierung komplexer Verhältnisse verschiedener Pläne und der damit verbundenen Aktionen.

Allerdings fällt bereits am obigen Beispiel (aus der Praxis) auf, dass es zu Mehrdeutigkeiten kommen kann und wird. Das liegt zum einen daran, dass der beobachtete Agent (im einfachsten Falle) zwar einen Plan verfolgt, dieser aber vom System nicht sofort erkannt werden kann⁷. Z.B. könnte der Agent zwar den Plan verfolgen, eine Email *lesen und dann löschen* (Read_and_store_one_msg) zu wollen, oder *lesen und dann speichern* (Read_and_delete_one_msg). Wenn eine konkrete Email-Anwendung nun gestartet und den Planerkenner auf die initiale Plan-Hypothese (im obigen Fall die Wurzel und abstrakte Plan *Process_Mail*) setzen würde, so hat das System nun erst mal keine Möglichkeit direkt herauszufinden, welche der obigen Basis-Pläne nun vom Agenten intentioniert werden.

Die Disambiguierung der potentiellen Hypothesen ist wohl die aufwendigste und komplizierteste Komponente des ganzen Planerkenners, an der übrigens sehr heftig weitergeforscht wird, da die State-Of-The-Art-Implementationen aufgrund der Komplexität der Plan-Hierarchien einen recht hohen (d.h. exponentiellen)

⁷ Eine sinnvolle Anwendung wird in der Regel nicht nur eine sondern meist eine Fülle von Funktionen und Unterprogrammen anbieten, um den Benutzer eine möglichst hohe Flexibilität zu ermöglichen. Daraus ergibt sich folglich auch eine Menge von potentiellen Zielen (Goals) für den Planerkenner.

Berechnungsaufwand mit sich bringen⁸. Nachfolgend werden einige Methoden zur Bewertung von beobachteten Aktionen eines Agenten vorgestellt. Diese setzen allerdings zumindest eine chronologische Aufzeichnung aller Aktionen voraus, da durch die verwendeten Verfahren das momentane a priori Wissen (Plan-Hypothese zu einem bestimmten Zeitpunkt) durch vergangene Beobachtungen *verbessert* werden kann (update). Das System sammelt dieses Wissen über eine längere Trainingsphase hinweg und kann dadurch die Präferenzen (Gewohnheiten bzw. Vorlieben) des Agenten ermitteln.

⁸ Die Methode der Zielgraphen erzielt eine Senkung auf polynomialen Aufwand (laut Hong, s. [H01b])

4 Methoden zur Bewertung von Hypothesen

Durch empirische Tests wurde ermittelt (nach [H03]), dass im Durchschnitt 38.6% der Aktionen eines Plans den optimalen Befehlen (der effizientesten Lösung) entsprechen, d.h. ein durchschnittlicher Plan beinhaltet 2.6-mal mehr Benutzereingaben als nötig. Hinzu kommen häufig falsche, jedoch mit dem korrekten Befehl verwandte Kommandos; anstatt des effizientesten Befehls oft nur eine synonyme Aktionssequenz und eine Menge Tippfehler. Ferner holen Benutzer Informationen manchmal über die Syntax des angewandten Befehls ein, des Öffern aber auch über die Verzeichnis- Struktur und über die darin enthaltenen Dateien. Sie verfolgen manchmal mehrere Pläne gleichzeitig und sehen im Allgemeinen einen Computer grundsätzlich nicht als vollwertigen Dialogpartner an; auch dann nicht, wenn sie ihre Fragen in natürlicher Sprache formulieren können.

Schon aufgrund dieser Schwierigkeiten werden leistungsfähige Verfahren benötigt, um die erzeugten multiplen Hypothesen zur Erklärung des Plans (des beobachteten Users/Agenten) korrekt oder zumindest aussagekräftig zu bewerten.

Eine Möglichkeit dafür bietet Carburry's „rational and coherent“-Heuristik, die die Auswahl einer Hypothese aus einer Konkurrenz-Menge durchführt oder die Fokussierung (Plausibilisierung von beobachteten Aktionsfolgen, d.h. Aktionen die für bestimmte Pläne gedacht bzw. sinnvoll sind funktionieren nicht oder führen nicht zum Ziel bei anderen Plänen und können dadurch verworfen werden)⁹. Weitere Verfahren wie (die häufig verwendete) Abduktion, Diagnoseansätze und Adaption an den Agenten werden im Rahmen dieser Arbeit jedoch nicht weiter behandelt.

⁹ Diese Heuristiken und simple Selektionsverfahren sind allerdings nicht mit den extrem leistungsfähigen Bewertungsverfahren aus den nachfolgenden Kapitel zu vergleichen.

4.1 Der erste Ansatz (Kautz's formale Theorie)

Einen der ersten Ansätze zur Bewertung von Hypothesen gibt Kautz (s. [K91]) mit seiner formalen Theorie an. Diese definiert eine formal-logischen Sprache und beschreibt sogenannte *Events*, die in einer Event-Hierarchie¹⁰ H organisiert werden. Diese besteht aus der nicht weiter zerlegbaren Menge H_{EB} aller Basis-Pläne, H_E (für *AnyEvent* und *End*), der Menge H_A von Abstraktionsaxiomen, der Menge H_D aller Dekompositionsaxiomen und einer Menge H_G von *generellen* Axiomen (zur Modellierung einiger Restriktionsrelationen).

Definition (Hierarchie)¹¹:

$$H = \langle H_E, H_{EB}, H_A, H_D, H_G \rangle \quad (\text{A2})$$

Die Erkennung von Plänen bei diesem Ansatz wird durch Inferieren von logischen Schlüssen mit Axiomen auf Events bzw. deren Abstraktionen. Allerdings hat diese Theorie eine Menge Einschränkungen hinsichtlich der ihrer Anwendbarkeit. Sie kann z.B. nicht für in einem flexiblem System eingesetzt werden, da durch die Closed-World-Assumption¹² keine neuen Pläne erzeugt und zur Hierarchie hinzugefügt werden können, d.h., alle Pläne müssen für den Erkennungsprozess bereits vorhanden. Kautz gibt sogar zu, dass ohne eine weitere Modifikation der verwendete Formalismus eher gut zum Erkennen von stereotypischem Verhalten geeignet ist.

¹⁰ Diese ist mit der Plan-Hierarchie vergleichbar.

¹¹ Für genaueres zur Kautz's Theorie und Definitionen siehe [K91]

¹² Annahme über vollständiges Wissen der Domäne

4.2 Bayes'sche Netze

Durch die Einführung der Wahrscheinlichkeitstheorie in die Planerkennung lassen sich zumindest einige Unzulänglichkeiten logischer Inferenzengines umgehen, in dem Wahrscheinlichkeiten zur Modellierung der Unsicherheit von Hypothesen oder Beobachtungen herangezogen werden können.

Darüber hinaus wird durch die Verwendung von bedingten Wahrscheinlichkeiten eine genauere Bewertung der Hypothesen ermöglicht (Berechnung der Wahrscheinlichkeitverteilung innerhalb des Suchraums). Diese Werte können zur Evaluierung der Glaubwürdigkeit und zur Ranglistengenerierung für Hypothesen herangezogen werden.

Definition (a posteriori Wahrscheinlichkeit, nach Bayes):

$$P(x | y) := \frac{P(y | x) \cdot P(x)}{P(y)} \quad (\text{A3})$$

Mit dem Likelihood $P(y | x)$ und a priori Wahrscheinlichkeit $P(x)$. Der Term $P(y)$ im Nenner wird zur Normalisierung verwendet.

Definition (Bayes'sches (Belief) Netzwerk):

Ein BN ist ein Modell, das (alle) möglichen Zustände der (betrachteten) Domäne angibt. Es ist ein azyklischer gerichteter Graph, dessen Kanten die kausalen Zusammenhänge und Effekte in einer Domäne modelliert. Die a-posteriori-Wahrscheinlichkeiten eines jeden Knoten werden durch die Bayes-Formel (aus A3) berechnet und in einer Tabelle (CPT, conditional probability table) abgelegt. Daraus ergibt sich auch die JPD (*Joint Probability Distribution* oder *gemeinsame Wahrscheinlichkeitsverteilung*) des Netzwerks die sich aus diesen CPTs zusammensetzt.

Die Faktorisierung der JPD in die konditionalen Wahrscheinlichkeitsverteilungen (repr. durch die CPT) wird durch Abb. 4.2.a veranschaulicht.

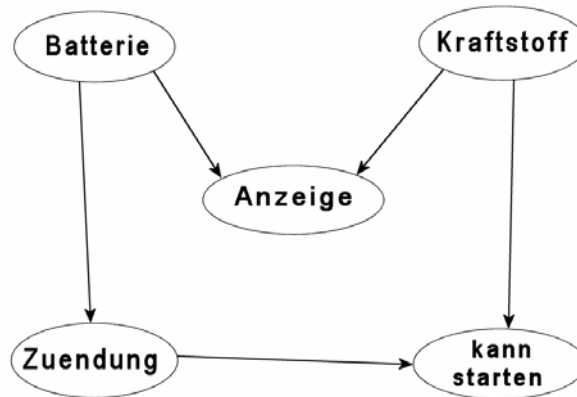


Abb. 4.2.a: Ein einfaches Beispiel-BN für Fahrzeugstart

Mathematisch ausgedrückt ergibt sich für das obige Netzwerk folgender Term¹³:

$$P(B, K, A, Z, S) = P(B)P(K)P(A|B, F)P(Z|B)P(S|K, Z)$$

BN sind kompakte Netzwerke von (bedingten) Wahrscheinlichkeiten die propabilistische Relationen zwischen Variablen und deren historische Informationen erfassen können. Ihr großer Vorteil ist, dass die sie nicht zwingend aktuelle Evidenzen oder exakte historische Informationen kennen müssen. Allerdings beachten sie keine unvorhergesehenen Events und es besteht die Möglichkeit, dass die Berechnung u.U. auch NP-hart¹⁴ wird, wodurch die Effizienz des ganzen Systems in Mitleidenschaft gezogen würde. Dieser Aufwand lässt sich meistens jedoch durch die Markovsche Annahme¹⁵ ein bisschen verbessern.

Für das in Abb. 4.2.b dargestellte (leicht modifizierte) Netzwerk wird genau das angenommen, wodurch sich die Faktorisierung (u.U.) von

$$P(B, V, E, K, A, Z, S) = P(B)P(V|B)P(E|V)P(K)P(A|E, F)P(Z|E)P(S|K, Z)$$

auf

$$P(B, V, E, K, A, Z, S) = P(E|V)P(K)P(A|E, F)P(Z|E)P(S|K, Z)$$

¹³ Der Einfachheit halber wurden hierfür die Anfangsbuchstaben der Knotenbezeichnungen verwendet.

¹⁴ nont-deterministisch polynomial hard

¹⁵ Hierbei wird vorausgesetzt, dass nur der Knoten x_{i-1} einen Einfluß auf die a-posteriori-Wahrscheinlichkeit hat und nicht die gesamte Beobachtungsreihe. Dadurch läßt sich die Berechnung auf die unmittelbar vorhergehenden Knoten beschränken. Siehe auch [W00].

verbessern lässt.

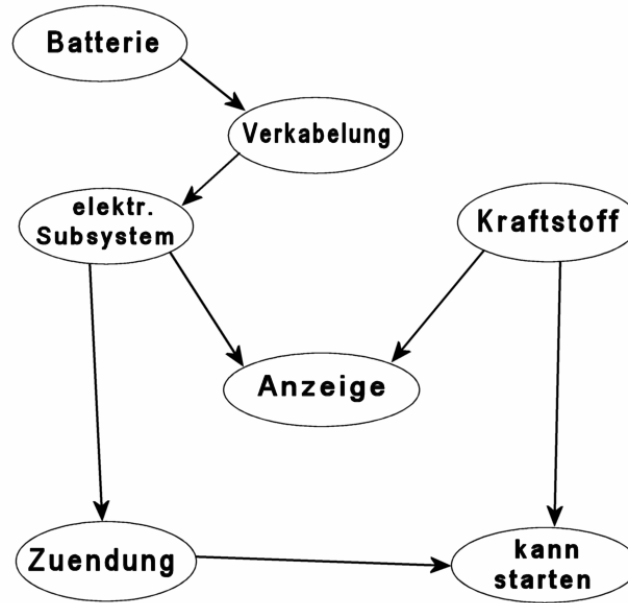


Abb. 4.2.b: Beispiel-BN mit tieferem Abhängigkeitspfad

Wenn man nun ein Bayes'sches Netz einem Plan zuweist, so kann durch die Berechnung der Wahrscheinlichkeiten eine Bewertung erreicht werden. Eine Applikation dieses BN-Ansatzes wird nun entweder für alle Pläne der Domäne entsprechend viele Netze enthalten, oder aber nur ein einziges zusammengesetztes Netzwerk.

Verglichen mit dem formalen Ansatz aus dem letzten Kapitel, ist eine Erweiterung auf *dynamische Bayes'sche Netze* (DBN) möglich, wodurch bereits ein enormer Grad an Flexibilität erreicht werden kann (siehe dafür auch [A98]).

4.3 Dempster-Shafer-Theorie (DST)¹⁶

Eine weitere Methode zur Modellierung der Kernelemente eines Planerkennters basiert auf einer Mathematik, die die klassische Wahrscheinlichkeitsrechnung erweitert (Bauer et al, [B95]). In der Stochastik werden Events betrachtet und die Wahrscheinlichkeiten dieser Beobachtung werden zur Rechnung herangezogen wobei traditionellerweise ein reellwertiges Intervall zwischen Null und Eins gültig ist. Ferner bedeutet der numerische Wert 0, dass das Auftreten eines Ereignisses völlig unmöglich ist und ein Wert von 1 sagt dementsprechend das sichere Auftreten des Ereignisses voraus.

Anders verhält es sich bei der DST. Hier wird die *Unwissenheit* als Grad des Vertrauens in Evidenzen modelliert und quantifiziert, mit der Wahrscheinlichkeitstheorie werden jedoch nur Unsicherheiten behandelt. Der numerische Wert 0 sagt bei der DST lediglich aus, dass es keine Evidenzen gibt, die den Wahrheitswert unterstützen können. Im Falle von unzureichenden Informationen wäre die Bewertung von Hypothesen in der klassischen WR nur mit einer vorher (empirisch) ermittelten Wahrscheinlichkeitsverteilung durchführbar. Die Dempster-Shafer-Theorie löst das Problem durch ihren Counterpart zur Wh.-Verteilung, die *Basiswahrscheinlichkeiten* (bpa, basic probability assignment¹⁷).

Die DST kann auch als eine Generalisierung der Bayes-Theorie und als Alternative der Bayes'sian (Belief) Networks.

Definition Basiswahrscheinlichkeit (Evidenzmasse):

Die Abbildung $m: 2^\Omega \rightarrow [0,1]$ ist eine Basiswahrscheinlichkeit (bpa) unter der Annahme:

$$m(\emptyset) = 0 \tag{A4}$$

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{A5}$$

Die Elemente aus $A \subseteq \Omega$ mit $\text{bpa} > 0$ werden „Fokalelemente von m“ genannt. Hierbei wird 2^Ω als Hypothesen-Frame¹⁸ bezeichnet.

¹⁶ Für einleitende und weiterführende Literatur zur Dempster-Shafer-Theorie siehe [S90], [S76] und [B98].

¹⁷ In der Literatur wird die Basiswahrscheinlichkeit manchmal auch als BBA (Basic Belief Assignment) aufgeführt

¹⁸ nicht zu verwechseln mit dem *Evidenz-Frame* 2⁹

Definition Belief:

$$Bel_m(A) = \sum_{B \subseteq A} m(B) \quad (A6)$$

Definition Plausibilität:

$$Pl_m(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad (A7)$$

Mit A6 und A7 wird nun das Belief-Intervall angegeben:

$$[Bel_m(A), Pl_m(A)]$$

Das Belief-Intervall, genauer eigentlich der *Grad der Unwissenheit*, (DOI, degree of ignorance) kann nun als numerischer Klassifikator genutzt werden:

$$DOI(A) = Bel_m(A) - Pl_m(A) \quad (A8)$$

Definition Dempster's Kombinations-Regel:

Zwei unabhängige Evidenzen können über diese Regel verknüpft werden (Kombination mit Normalisierung):

$$(m \oplus m')(A) = \begin{cases} \frac{\sum_{B \cap B' = A} (m(B) \cdot m'(B'))}{1 - \sum_{B \cap B' \neq \emptyset} (m(\emptyset) \cdot m'(\emptyset))} & X \neq \emptyset \\ 0 & X = \emptyset \end{cases} \quad (A9)$$

Annahme für die Wissensbasis: Closed-World-Assumption, d.h. es wird *vollständiges Wissen* über die Domäne angenommen Die kann geschrieben werden durch:

$$(\tilde{m}(\Omega) = 1, \tilde{m}(A) = 0) : A \neq \Omega$$

Wenn man nun für $\mathcal{P}$ (aus A1) den Hypothesen-Frame 2^Ω (alle basis- und abstrakt-Pläne) annimmt, so kann die DST an eine Plan-Hierarchie geknüpft werden. Initial gelten außerdem alle Elemente von 2^Ω als mögliche Hypothesen.

4.4 Zielgraphen (Goal Graph)

Ein sehr großes Problem bei der Planerkennung ist die Berechenbarkeit. Die Verfahren aus den letzten Kapiteln führen (im schlimmsten Fall) zu 2^n Berechnungen, d.h., dass es in der Regel exponentiellen Aufwand gibt. Mit den Zielgraphen (s. [H01a], [H01b]) wird der Aufwand beweisbar auf ein polynomiales Maß herabgesetzt, was z.B. gegenüber der Dempster-Shafer-Theorie bereits eine wesentliche Senkung bietet.

Eine Planbibliothek wird auch nicht benötigt wodurch auch die Akquisition, die manuelle Erzeugung komplexer Bibliotheken und der damit verbundene exponentielle Speicherbedarf entfallen. Der Zielgraph wächst inkrementell mit den protokollierten Aktionen des Benutzers und den dadurch erreichten Zuständen des Systems, wodurch eine enorme Flexibilität erreicht wird. Auch ist eine Erkennung von mehreren konkurrierenden jedoch konsistenten Zielgraphen möglich (wenn mehrere Pläne bzw. beobachtete Aktionen Gemeinsamkeiten in der Zielbeschreibung aufweisen), was bei anderen Verfahren überhaupt nicht denkbar ist.

Die Arbeit von Hong vertieft sich allerdings eher in der Erkennung von teilweise oder komplett erreichten Zielen anstatt dem Versuch, zukünftige Ziele vorherzusagen. Dies ist besonders gut für das Verstehen von „Geschichten“ geeignet. In einer praktischen Anwendung könnte ein Agent zuerst beobachtet und seine Aktionen analysiert werden um im nachhinein eine Verbesserung des ausgeführten (und durch das System erkannten) Plans herzuleiten (Planoptimierung), bzw. es könnte ein optimaler Plan für den Agenten vorgeschlagen werden, z.B. nach einer abgeschlossenen Sequenz von Datenbank-Queries könnte das System die Absicht des Agenten herausfinden und einen optimierten Plan/Query erstellen.

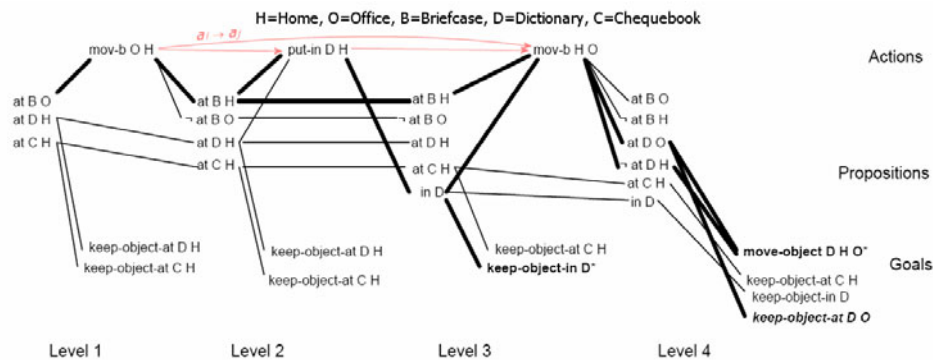


Abb. 4.4.a: Beispiel-Zielgraph

Ein Zielgraph wie in Abb. 3.2.4.a (nach Hong, s. [H01b]) repräsentiert die beobachteten Aktionen, (wahre bzw. explizit falsche) Propositionen und die in den jeweiligen Zeitframes voll oder teilweise erreichten Ziele.

Definition (gültiger Plan):

$$P = \langle A, C, L \rangle \quad (\text{A10})$$

mit A , der Menge beobachteter Aktionen, der Menge C von Einschränkungen (Constraints) und einer Menge L von kausalen Links der Form $\{ a_i \rightarrow a_j \}$ über A .

Sei ferner I die Menge von Initialbedingungen und g das Ziel (goal), dann ist P nur dann ein gültiger Plan, wenn, in der zu C konsistenter (jedoch darin beliebiger) Reihenfolge, alle Aktionen aus A ausgeführt werden können, und, der Plan vollständig erreicht ist.

Das folgende kurze Beispiel (aus [H01b]) zeigt einen Plan für den Zielgraphen aus Abb. 3.2.4.a.

Seien $I = \{ (\text{at B O}), (\text{at D H}), (\text{at C H}) \}$ und

$$\begin{aligned} P = (\{ a_1 = (\text{mov-b O H}), a_2 = (\text{put-in D H}), a_3 = (\text{mov-b H O}) \}, \\ \{ a_1 < a_2, a_2 < a_3 \}, \\ \{ a_1 \rightarrow a_2, a_1 \rightarrow a_3, a_2 \rightarrow a_3 \}) \end{aligned}$$

P ist ein gültiger Plan für das Ziel $g = (\text{move-object D H O})$.

Der Planerkennungsprozess bei einer Applikation mit diesem Ansatz teilt sich in zwei Stufen (Ansatz von Hong), in den Zielgraphen-Konstruktor und die Zielgraphen-Analyse, wobei beide Stufen für jeden Zeitschritt (Level) erneut ausgeführt werden.

In der ersten Stufe erweitert der Konstruktor den bereits vorhandenen Zielgraphen um die neu beobachteten ausgeführten Aktionen, bzw. erstellt einen neuen Graphen wenn noch keine vorhanden ist (z.B. in der Init-Phase).

In der zweiten Stufe analysiert das System den Graphen und erkennt vollständig oder partiell erreichte Ziele, und zwar nur diejenigen, die für die aktuell beobachteten Aktionen konsistent sind.

5 Schlusswort

Trotz der Tatsache, dass es sich beim Gebiet der Planerkennung um ein gerade mal ca. 20 Jahre junges Forschungsgebiet handelt, gibt es bereits eine kaum zu überschauende Anzahl von wissenschaftlichen Papers. Wenn man noch das Gebiet der Plangenerierung hinzunimmt (das als alleinstehende Disziplin bereits einen größeren Umfang hat als die Planerkennung selbst) wird einem schnell bewusst, dass ein ungeheuer großes Interesse darin liegt, Intelligente Systeme mit Interaktions-Komponenten auszustatten und ihnen die Fähigkeit zu geben, so natürlich wie nur möglich zu wirken – ja am liebsten sogar so natürlichsprachlich wie man es von Menschen gewohnt ist. Ob es in naher Zukunft jedoch tatsächlich Systeme geben wird, die solche Dialoge führen können (und sich dabei genauso verhalten bzw. artikulieren wie Menschen) kann – trotz dem rasanten Tempo der Computerentwicklung und all der hier vorgestellten Techniken und den Aussichten auf deren mögliche Verbesserungen – wohl nicht direkt angenommen werden.

6 Literatur

- [A98] Albrecht, D. W., Zukerman, I., Nicholson, A. E., „Bayesian Models for Keyhole Plan Recognition in an Adventure Game”, Department of Computer Science, Monash University, Clayton, Australia, 1998
- [B95] Bauer, M., „A Dempster-Shafer Approach To Modeling Agent Preferences For Plan Recognition”, German Research Center for Artificial Intelligence (DFKI), 1995
- [B96a] Bauer, M., „Acquisition of User Preferences for Plan Recognition”, German Research Center for Artificial Intelligence (DFKI), 1996
- [B96b] Bauer, M., „Ein evidenztheoretischer Ansatz zur Planerkennung“, Dissertation, Saarbrücken, 1996
- [B98] Ruthven, I., Lalman, M., „Experimenting on Dempster-Shafer’s Theory of Evidence in Information Retrieval”, Department of Computer Science, University of Glasgow, 1998
- [H01a] Hong, J., „Plan Recognition without Plan Libraries: A Plan Graph Based Approach”, School of Information and Software, Engineering University of Ulster at Jordanstown, Newtownabbey, Co. Antrim BT37 0QB, UK
- [H01b] Hong, J., „Goal Recognition through Goal Graph Analysis”, *in: Journal of Artificial Intelligence Research*, Vol. 15, 2001, pp. 1-30
- [H03] Hoffmann, M.; „Intentionsbasierte maschinelle Interpretation von Benutzeraktionen“, Lehrstuhl für Mensch-Maschine-Kommunikation der TU-München, Dissertation, 2003, München
- [K02] Küpper, D., „Benutzermodellierung fuer benutzerspezifische Plangenerierung und -presentation“, Dissertation, Universität Essen, 2002

- [K91] Kautz, H.A., „A Formal Theory of Plan Recognition and its Implementation”, in: *Reasoning About Plans*, Allen, J.F., Kautz, H.A., Pelavin, R.N., Tenenber, J.D., Kaufmann, M., (pbl.), 1991, pp.69-126
- [L91] Lambert, L., „Modifying Beliefs in a Plan-Based-Dialog”, Department of Computer and Information Sciences, 1991
- [L98] Lesh, N., Rich, C., Sidner, C.L., „Using Plan Recognition in Human-Computer Collaboration”, in *7th International Conference on User Modeling*, Banff, Canada, 1998
- [S76] Shafer, G., „Dempster-Shafer-Theory”, School of Business, University of Kansas, 1976
- [S90] Smets, P., “What is Dempster-Shafer’s model?”, Université Libre de Brussels, 1990
- [W00] Wiegerinck, W., „Introduction to Bayesian Networks”, SNN, Radbound University Nijmegen, 2000
- [W02] Wobcke, W., „Two Logical Theories of Plan Recognition”, in: *Journal of Logic Computation*, Vol. 12, No. 3, pp. 371-412, 2002
- [Z93] Zwiesler, R., „Planen und Planerkennung“, 1993, reiner@zwiesler.de, <http://www.zwiesler.de/scripts/Planen.html>

7 Bildnachweis

- 3.a entnommen aus [B95] und nachbearbeitet
- 4.2.a basierend auf der Abbildung von Seite 12 in [W00]
- 4.2.b basierend auf der Abbildung von Seite 12 in [W00]
- 4.4.a entnommen aus [H01b] und ergänzt