Lehrstuhl für Mensch-Maschine-Kommunikation
Technische Universität München

Intentionsbasierte maschinelle Interpretation von Benutzeraktionen

Marc Hofmann

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. J. Hagenauer
Prüfer der Dissertation:	1. Univ.-Prof. Dr. rer. nat. M. Lang, i.R.
	2. Univ.-Prof. Dr.-Ing. J. Eberspächer

Die Dissertation wurde am 16.04.2003 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 28.09.2003 angenommen.

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Assistent am Lehrstuhl für Mensch-Maschine-Kommunikation der Technischen Universität München.

Besonderer Dank gebührt meinem Doktorvater *Prof. Manfred Lang* für die Gelegenheit, an den Forschungsaktivitäten seines Lehrstuhls mitwirken zu dürfen. Ausgehend von einer interessanten Themenstellung ließ er mir genügend Freiraum bei der Gestaltung meiner wissenschaftlichen Tätigkeit und war jederzeit für wertvolle Diskussionen und ermutigende Worte verfügbar. Mit viel Engagement initiierte er Industriekooperationen, durch die meine Arbeit eine spannende, motivierende Praxisrelevanz erhielt. Auch von der Teilnahme an nationalen und internationalen Fachtagungen, die von ihm stets unterstützt wurde, profitierte diese Arbeit.

Mein Dank gilt auch meinen ehemaligen Kollegen am Lehrstuhl für Mensch-Maschine-Kommunikation, die ein freundliches Arbeitsklima erzeugt haben. Hervorzuheben sind dabei Michael Geiger, Jörg Hunsinger, Ralf Nieschulz, Robert Neuss und Martin Zobl, die jederzeit mit wissenschaftlichen und kreativen Inputs meine Arbeit bereicherten. Besonderer Dank gilt dabei Robert Neuss für das Überlassen seiner Spracherkenneranbindung und Martin Zobl für tatkräftiges Mitwirken bei der Realisierung des Gestikerkenners. Peter Brand danke ich für die Bereitstellung einer jederzeit intakten technischen Infrastruktur.

Abschließend danke ich allen Personen aus meinem persönlichen Umfeld, die mich bei dieser Arbeit unterstützt haben.

München, im April 2003

Marc Hofmann

Zusammenfassung

Hauptgegenstand der vorliegenden Arbeit ist ein Ansatz zur Interpretation von Benutzeraktionen, ausgehend von allen potenziellen *Intentionen* des Benutzers. Dabei kann eine Intention einem unmittelbaren Ziel entsprechen, das durch eine Aktion erreichbar ist, oder einem übergeordneten Ziel, dessen Umsetzung eine Aktionssequenz erfordert.

Für das Verständnis von Benutzeraktionen verzichtet der *intentionsbasierte* Ansatz darauf, die eigentliche Aktion zu rekonstruieren, sondern setzt die Merkmale einer beobachteten Aktion direkt mit den in Frage kommenden *Intentionshypothesen* in Bezug. Dies ermöglicht eine robuste Klassifikation der Intention selbst auf Basis unvollständiger und verrauschter Beobachtungsfolgen. Der Ansatz orientiert sich somit an der menschlichen Fähigkeit, die Aussage einer Äußerung oder Handlung auf Grund von Wissen bezüglich des Gesprächsthemas zu erfassen, da Benutzeraktionen in der Regel den Zweck haben, das System von der Intention des Benutzers in Kenntnis zu setzen.

Kernkomponenten des intentionsbasierten Ansatzes sind Intentionsmodelle, die eventuelle Merkmalsbeobachtungen in der für eine Intention *syntaktisch-semantisch* charakteristischen Weise in Bezug setzen. Realisiert werden die Intentionsmodelle anhand von *Bayes'schen Netzen*, die eine effiziente Modellierung syntaktisch-semantischer Beziehungen erlauben.

Der Ansatz zur intentionsbasierten Interpretation wird zunächst allgemein gehalten und schließlich für vier unterschiedliche Kerngebiete der Mensch-Maschine-Kommunikation konkret umgesetzt. Dabei entstehen innovative Systeme aus den Bereichen Sprachverstehen, Planerkennung, Benutzermodellierung und Gestikerkennung.

Das *sprachverstehende System* ist in der Lage, durch syntaktisch-semantische Gewichtung von Wortbeobachtungen den Inhalt einer gesprochenen Äußerung zu ermitteln. Da Wortbeobachtungen direkt auf die Intentionsmodelle abgebildet werden, ist eine korrekte Klassifikation der Intention auch für stark verrauschte Wortketten und somit für undeutlich artikulierte Äußerungen möglich.

Die Umsetzung des intentionsbasierten Ansatzes für eine übergeordnete Interpretation unvollständiger Aktionssequenzen resultiert in ein System zur *Planerkennung*, das auch nicht optimales oder fehlerhaftes Handeln des Benutzers für die Klassifikation der Intention berücksichtigt.

Im Rahmen der Umsetzung des intentionsbasierten Ansatzes für eine Interpretation von Situationen und Aktionen entstand ein System zur *Benutzermodellierung*, das die Fähigkeit besitzt, ausgehend von der aktuellen Situation die Intention des Benutzers vorherzusagen. Das entwickelte Navigationsassistenzsystem ist somit in der Lage, nicht eindeutige Zielorteingaben zu verstehen.

Für die intentionsbasierte Interpretation *dynamischer Handgesten* wurde ein sehr einfach strukturiertes Intentionsmodell und eine neuartige Merkmalsextraktion entwickelt. Diese Kombination erlaubt sowohl eine robuste Klassifikation unterschiedlichster Arten von dynamische Gesten als auch die Echtzeit-Adaption des Intentionsmodells an den aktuellen Benutzer.

Die hohen Erkennungsraten der auf Basis des intentionsbasierten Ansatzes entwickelten Systeme bestätigen die Leistungsfähigkeit und das Potenzial des vorgestellten Verfahrens.

Inhaltsverzeichnis

1

Einleitung

1.1 Motivation

Ziel der Forschungsaktivitäten auf dem Gebiet der Mensch-Maschine-Kommunikation ist eine möglichst natürliche Interaktion zwischen Mensch und Maschine, die sich am zwischenmenschlichen Dialog orientiert und somit bestmöglich intuitiv gestaltet werden kann, da sie dem Benutzer aus dem Alltag vertraut ist. Das bisherige Verständnis dieser Interaktion beschränkt sich dabei vor allem auf die Auswahl der Eingabemodalitäten. Neben den klassischen haptischen Bedienelementen werden zukünftig die natürlichen Kommunikationskanäle wie Sprache, Gestik und Mimik eine tragende Rolle spielen [Lan02]. Für eine adäquate Systemreaktion ist die Ermittlung der Intention des Benutzers anhand mustererkennungsbasierter Verfahren unumgänglich. Unabhängig von der Eingabemodalität folgen die mustererkennungsbasierten Ansätze im Allgemeinen der in Abbildung 1.1 dargestellten Struktur.

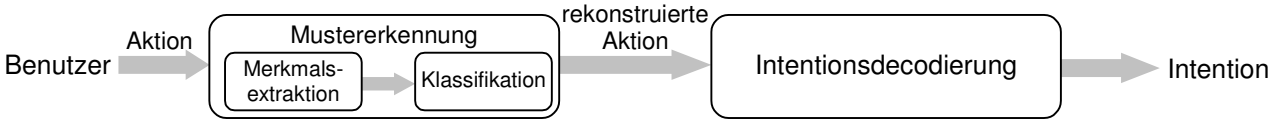


Abbildung 1.1: Klassischer Ansatz zum Ermitteln der Benutzerintention

Der Benutzer artikuliert seine Absicht anhand einer oder mehrerer Aktionen, die als Beobachtungsfolge durch die Komponenten *Mustererkennung* und *Intentionsdecodierung* analysiert werden. Dabei dient die Mustererkennung, bestehend aus Merkmalsextraktion und Klassifikation, der Rekonstruktion der eigentlichen Benutzeraktion auf Basis dieser Beobachtungsfolge. Darauf aufbauend wird schließlich mittels Intentionsdecodierung die eigentliche Intention bestimmt. Nachteil der strikten Trennung von Mustererkennung und Intentionsdecodierung sind Fehlklassifikationen der Intention im Falle fehlerhafter Aktionsrekonstruktionen.

Die Fähigkeiten des Menschen gehen über die bloße Rekonstruktion und Interpretation einer Aussage weit hinaus. Folgende Aspekte zeichnen die Kommunikationsfähigkeit des Menschen aus:

- Der Mensch ist in der Lage, die Aussage einer Äußerung bereits zu erfassen, bevor diese abgeschlossen bzw. vollständig ist.

- Eine sehr ausgeprägte menschliche Fähigkeit ist das intuitive Trennen von Nutzinformation und Störinformation. Ein typisches Beispiel hierfür ist der Cocktailparty-Effekt: Der Mensch konzentriert sich auf seinen Gesprächspartner und filtert Aussagen anderer anwesender Menschen aus dem akustischen Gesamtsignal.

- Durch Störeinflüsse nicht verstandene Wörter oder Satzfragmente können interpoliert werden.

Diese Phänomene lassen den Schluss zu, dass die Robustheit der zwischenmenschlichen Kommunikation nicht allein auf eine leistungsfähige „Mustererkennung“ zurückzuführen ist, wie sie in Abbildung 1.1 schematisiert wurde. Die Tatsache, dass selbst stark verrauschte oder unvollständige Beobachtungsfolgen zu einer korrekten Interpretation führen, stützt die Annahme, dass der Mensch auf eine explizite Rekonstruktion der ursprünglichen Äußerung verzichtet und zum Verstehen von Inhalten die Beobachtungsfolgen auf Basis semantisch höherwertiger Informationen analysiert. Bei diesen Informationen kann es sich um Wissen über das Gesprächsthema, den Gesprächspartner, die Situation oder generell um Weltwissen handeln. Dieses Wissen befähigt den Menschen, auch die Äußerungen eines Gesprächspartners auch unter nicht idealen Bedingungen zu erfassen und zu verstehen. *Verstehen* unterscheidet sich in diesem Zusammenhang von der *Erkennung* durch eine Bewertung von Beobachtungen anhand bekannter Zusammenhängen und Wissen. Ziel ist die Ermittlung des Inhalts einer Äußerung.

1.2 Zielsetzung und Lösungsansatz

Hauptaufgabe einer Mensch-Maschine-Schnittstelle ist die Analyse von Benutzeraktionen mit dem Ziel, die Absicht des Benutzers zu ermitteln. Um den Mensch-Maschine-Dialog ähnlich robust und leistungsfähig wie den zwischenmenschlichen Dialog zu gestalten, ist es unumgänglich, dem Rechner die Fähigkeit zu geben, Benutzeraktionen zu verstehen und den Inhalt auf Plausibilität zu überprüfen ohne eine exakte Rekonstruktion der Benutzereingabe. Die Entwicklung eines derartigen Verfahrens ist Hauptgegenstand dieser Arbeit.

Eine verstehende Eingabeschnittstelle setzt Wissen über mögliche Interaktionsinhalte voraus. Eine maschinelle Modellierung des gesamten Wissens, auf das der Mensch im Rahmen der Interaktion mit einem Rechner zugreifen kann, ist aus Komplexitätsgründen nicht möglich. Dies ist im vollen Umfang auch nicht nötig, da ein Benutzer eine Applikation durch Aktionen nicht beliebig manipulieren kann, sondern nur im dem Umfang, wie es die Software zulässt bzw. unterstützt. Die potenziellen Absichten des Benutzers im Umgang mit der entsprechenden Applikation beschränken sich somit auf die Funktionalität des Systems. Da es sich bei der Absicht des Benutzers, der *Intention*, um die eigentliche Motivation handelt, sich überhaupt mit einem technischen System auseinander zu setzen, wird im Rahmen des hier vorgestellten Verfahrens die *Benutzerintention* in den Mittelpunkt der Betrachtungen gerückt, da jede Aktion des Benutzers als direkte Folge seiner Intention interpretiert werden kann. Dies entspricht der Definition des Intentionsbegriffs durch den Philoso-

phen Bratman [Bra90], der die Rolle der Intention für die Interaktion mit einem technischen System als Erster analysiert hat.

Ziel dieser Arbeit ist die Entwicklung eines Ansatzes zur *intentionsbasierten Interpretation von Benutzeraktionen*. Das Verfahren ist in der Lage, Benutzereingaben mit dem Wissen über mögliche Intentionen zu evaluieren, um damit Aussagen über die Benutzerintention zu treffen. Die Grundidee wird an dieser Stelle nun knapp vorgestellt.

Der Benutzer artikuliert seine Intention durch eine Aktion oder eine Aktionssequenz, die als Beobachtungsfolge von der Merkmalsextraktion in Hinblick auf die informationstragenden Elemente analysiert wird. Im Unterschied zu den in Abbildung 1.1 dargestellten klassischen Komponenten der Intentionsbestimmung werden die Merkmale nicht zur Rekonstruktion der Aktion herangezogen, sondern direkt mit potenziellen Intentionen in Bezug gesetzt. Die Komponenten *Klassifikation* und *Intentionsdecodierung* werden durch die *intentionsbasierte Interpretation* und durch *Intentionsmodelle* ersetzt. Abbildung 1.2 schematisiert den intentionsbezogenen Ansatz.

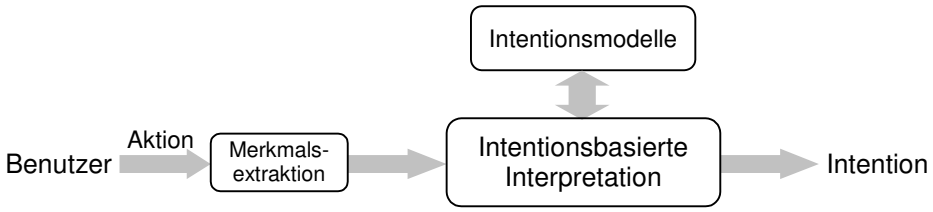


Abbildung 1.2: Grundidee des intentionsbasierten Ansatzes zur Interpretation von Benutzeraktionen

Da Handlungen des Benutzers darauf abzielen, den Rechner von dessen Absicht in Kenntnis zu setzen, gleicht die Modellierung aller in Frage kommenden Intentionen der Berücksichtigung aller potenziellen Interaktionsinhalte. 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 *Intentionsmodelle*, die durch Bayes'sche Netze [Pea88][Jen92][Rus94] realisiert werden.

Die Intentionsmodelle repräsentieren alle Intentionshypothesen in Abhängigkeit von Aktionssequenzen, die der Umsetzung des entsprechenden Ziels dienen könnten. Bayes'sche Netze erlauben eine effiziente, statistische Modellierung syntaktisch-semantischer Zusammenhänge und ermöglichen somit die Repräsentation komplexer Beziehungen zwischen Merkmalen und Intentionshypothesen. In erster Linie agieren die Intentionsmodelle bzw. die Bayes'schen Netze aber als Klassifikatoren, indem sie einen quantitativen Bezug zwischen Aktionsmerkmalen und den Intentionshypothesen herstellen können. Zudem wurden Algorithmen zur Online-Adaption der Intentionsmodelle an neue Aktionen oder Personen erarbeitet.

Da die Interpretation einer Benutzeraktion im Allgemeinen nicht die einzige Komponente einer Mensch-Maschine-Schnittstelle ist, müssen ihre Ergebnisse eindeutig interpretierbar sein. Auch aus diesem Grund werden Bayes'sche Netze als Klassifikatoren gewählt, da die Wahrscheinlichkeitstheorie eine sehr einfache und allgemein verständliche Interpretierbarkeit quantitativer Aussagen zulässt.

Den eigentlichen Klassifikationsprozess steuert die Komponente *intentionsbasierte Interpretation*, indem sie Merkmale auf die Intentionsmodelle abbildet und somit Aussagen über die wahrscheinlichste Intentionshypothese treffen kann. Die Klassifikationslogik ist somit verteilt auf diese Komponente und auf die Topologie der Intentionsmodelle.

Da der Ansatz zur intentionsbasierten Interpretation von Benutzeraktionen Zugriff auf Wissen über die potenziellen Intentionen, deren charakteristische Merkmale und über die syntaktisch-semantischen Beziehungen zwischen den Aktionsmerkmalen hat, handelt es sich um ein *verstehendes System*. Unabhängig von der verwendeten Eingabemodalität orientiert sich die Grundidee am robusten zwischenmenschlichen Dialog und verspricht daher eine effizientere Interaktion des Menschen mit einem Softwaresystem.

Der Fokus liegt bei der vorliegenden Arbeit auf der Entwicklung eines *allgemeinen* Ansatzes zur *Intentionsbasierten Interpretation von Benutzeraktionen*, dessen Grundidee unabhängig von der zu behandelnden Problematik möglichst allgemein gültig sein soll. Diese Arbeit zielt also nicht darauf ab, neue Konzepte für Teilbereiche der Mensch-Maschine-Kommunikation, wie zum Beispiel Sprachverstehen oder adaptive Systeme, zu entwickeln, sondern einen potenziellen Nutzen des intentionsbasierten Verstehens zu überprüfen. Dies erfordert eine Umsetzung und Realisierung des intentionsbasierten Ansatzes für eine Reihe völlig unterschiedlicher Kerngebiete der Mensch-Maschine-Kommunikation. Der Schwerpunkt liegt auf der theoretischen sowie auf der praktischen Umsetzung, weshalb auf jedes im Rahmen dieser Arbeit entwickelte Softwaresysteme gesondert und sehr ausführlich eingegangen wird. Darüber hinaus weisen die vier entwickelten Systeme diverse Innovationen auf.

Der intentionsbasierte Ansatz wurde für folgende vier Forschungsgebiete der Mensch-Maschine-Kommunikation umgesetzt: Sprachverstehen, Planerkennung, Benutzermodellierung und Gestikerkennung. Die einzelnen Systeme werden in Kapitel 1.4 kurz angesprochen und schließlich in den Kapiteln 3 bis 6 im Detail vorgestellt.

1.3 Stand der Technik

In diesem Kapitel wird dargelegt, wie der Begriff der *Intention* in technischen Systemen bisher interpretiert und umgesetzt wurde. Für die in den Kapitel 3 bis 6 vorgestellten konkreten Systeme wurde jeweils ein eigenes Unterkapitel zum Stand der Technik auf den jeweiligen Forschungsgebieten formuliert, sodass an dieser Stelle die grundlegenden Ideen hinter den angesprochenen Verfahren und nicht die Algorithmen im Fokus stehen.

Es gibt eine Reihe klassischer Anwendungsgebiete intentionsbasierter Ansätze. Häufig werden dabei die Begriffe *intentionsbasiert* und *planbasiert* in der Literatur synonym verwendet, wobei der erste Begriff dem übergeordneten Ziel im Sinne Bratmans [Bra90] entspricht und der zweite Begriff die Betonung auf die Aktionsabfolge selbst legt. Planbasierte Ansätze haben somit selten die Bestimmung des Benutzerziels, sondern eher die Planung zukünftiger Aktionen im Blickfeld. Intentionsbasierte Verfahren sind dann von Nutzen, wenn das höherwertige Ziel des Benutzers interessant ist. Auf einige dieser Systeme wird nun eingegangen.

Die ersten intentionsbasierten Verfahren gehen auf die Arbeiten von Cohen, Perault und Allen zurück [Coh79][Per80][All80]. Diese Wissenschaftler erreichten eine robustere maschinelle Interpretation von Diskussionen und Dialogen, indem nach jeder Aussage eines Gesprächsteilnehmers die Intentionen der Gesprächspartner abgeschätzt wurden, um somit die Thesen einer Diskussion zu ermitteln. Die Umsetzung erfolgte durch eine formale logische Beschreibung von Kontextwissen, Aktionen und Systemzuständen, um regelbasiert eventuelle semantische Verbindungen einer Aussage zu vorhergehenden Äußerungen herstellen zu können. Dabei handelt es sich um eine sehr anspruchsvolle Aufgabenstellung, da die intentionsbasierte Komponente herauszufinden hat, ob sich die Äußerung eines Sprechers auf seine eigene vorhergehende Aussage oder auf Aussagen anderer Gesprächsteilnehmer bezieht.

Aufbauend auf dieser Grundidee gibt es eine große Zahl von Systemen zur Interpretation von Dialogen und Diskussionen, die zum Ziel haben, grundlegende Meinungen und Intentionen der Gesprächspartner zu erfassen, um somit ein robusteres Abschätzen des Gesprächsinhalts zu ermöglichen. Der Tatsache, dass sich Meinungen von Gesprächspartnern im Laufe einer Diskussion ändern können, wurde von Lambert [Lam91] Rechnung getragen, dessen Verfahren in der Lage ist, den mentalen Zustand bzw. die Intention von Gesprächspartnern derart zu modellieren, dass dies berücksichtigt werden kann.

Ähnliche Mechanismen machte man sich im Rahmen des BMBF-geförderten Projekts *VERBMOBIL* [Wah00] zu Nutze. Ziel des Projekts war eine maschinelle Dolmetscherfunktionalität für natürlichsprachliche Terminvereinbarungen. Da gerade natürliche und spontane Sprache eher zu Satzfragmenten führt als zu syntaktisch-semantisch korrekten Sätzen, wird eine intentionsbasierte Betrachtungsweise herangezogen, um die für die Intentionshypothesen bedeutungstragenden Elemente einer Äußerung herauszufiltern. Auf eine exakte Rekonstruktion einer Aussage zu verzichten, bietet sich in der *VERBMOBIL*-Domäne an, da spontansprachliche Äußerungen häufig nicht syntaktisch korrekt sind, ihre Übersetzung allerdings grammatikalisch richtig sein sollte. Zudem ist die Anzahl potenzieller Äußerungsinhalte bzw. Intentionen für die Domäne Terminvereinbarung sehr begrenzt, sodass keine Komplexitätsprobleme zu erwarten sind.

Die Dialogkomponente von *VERBMOBIL* teilt einen typischen Gesprächsverlauf in drei Teile ein: die Begrüßungsphase, die Verhandlungsphase und die Verabschiedungsphase. Die Erkennung der Phasen erfolgt anhand eines graphenbasierten Planerkenners [Ale95][Ale96][Bub96]. Durch Ermitteln der Gesprächsphase können bereits die möglichen Inhalte einer Äußerung eingegrenzt werden, d.h., die Anzahl der Intentionshypothesen wird reduziert. Auf Basis dieser Intentionshypothesen wird schließlich der Inhalt der aktuellen Äußerung graphenbasiert bestimmt, wobei dies durch Einbeziehen vorheriger Äußerungsinhalte und Kontextwissen unterstützt wird. Dieses Ergebnis wird schließlich noch für eine einfache Bigramm-basierte Vorhersage des Inhalts der nächsten Äußerung verwendet, um die Übersetzung der aktuellen Äußerung natürlicher zu gestalten. Auch an dieser Stelle ist die übergeordnete Intention von Interesse.

Diese im Rahmen von Verbmobil entwickelten Ideen dienen auch im Nachfolgeprojekt SMARTKOM als Ausgangsbasis für Weiterentwicklungen mit dem Schwerpunkt auf einem multimodalen Mensch-Maschine-Dialog [Wah01].

In den beschriebenen Systemen zur Dialogmodellierung dient die intentionsbasierte Vorgehensweise der Integration von domänenspezifischem Wissen. Dies trifft auch auf den im Rahmen dieser Arbeit vorgestellten Ansatz zur intentionsbasierten Interpretation von Benutzeraktionen zu.

Neben dem Einsatz in Dialogmodellen haben intentionsbasierte Verfahren eine Tradition in intelligenten Tutorsystemen. Um auf Probleme eines Benutzers bei der Lösung einer Aufgabe mit einer adäquaten Hilfestellung reagieren zu können, muss ein Tutorsystem die Schwierigkeiten des Benutzers nicht nur erkennen, sondern auch verstehen können. Aufbauend auf dieser Information kann schließlich eine Strategie erarbeitet werden, um den Benutzer gezielt zur richtigen Lösung zu führen. In vielen intelligenten Tutorsystemen dient die Intention somit als Grundlage für eine didaktische Dialogmodellierung [Zho99][Fre00]. Im Rahmen eines Tutorsystems gleicht eine Intention einer möglichen Lösung der gestellten Aufgabe. Die intentionsbasierte Komponente muss also deshalb typische Fehler bei der Bearbeitung gestellter Aufgaben abdecken. Somit wird nicht nur festgestellt, dass eine Lösung eventuell nicht korrekt ist, sondern auch eine Interpretation der falschen Lösung durchgeführt. Dies ist die Voraussetzung für ein erfolgreich lehrendes Tutorsystem. Die Erkennung der Intention geschieht in diesem System meist regelbasiert.

Eine weiteres Forschungsgebiet, das auf das Verstehen übergeordneter Ziele von Personen aufbaut, ist die automatische Bewertung von Handlungsabläufen. Ein derartiges System ist *Asgaard* [Adv98], das in der Lage ist, von Ärzten getroffene medizinische Maßnahmen zu bewerten. Die Intention entspricht hierbei der Diagnose des Arztes. Um die Intentionen mit den Aktionen, den medizinischen Maßnahmen, in Bezug zu setzen, wurde die formale Sprache *Asbru* [Mik97] [Adv98] entwickelt, anhand derer eine regelbasierte Planerkennung betrieben werden kann und getroffene Maßnahmen beurteilt werden können. Ähnliche Verfahren wurden auch zur automatischen Bewertung von Benutzerschnittstellen von Softwaresystemen entwickelt mit dem Zweck, Usability-Untersuchungen [Nie94] auf ein Minimum zu reduzieren, um den Rapid-Prototyping-Prozess von Benutzungsschnittstellen zu beschleunigen.

In der Robotik finden planbasierte Ansätze vor allem für das Planen zukünftiger Aktionen häufig Anwendung. In diesen Systemen dienen Pläne als Leitfaden für die Aktuatorik des Roboters, um ausgehend vom aktuellen Systemzustand, einen Wunschzustand herzustellen. Auf der Sensorikseite sind planbasierte bzw. intentionsbasierte Ansätze vor allem dann interessant, wenn der Roboter auf Menschen oder auf andere Roboter reagieren muss, d.h. für so genannte Mehragentensysteme [Rus95]. Ein typisches Beispiel hierfür ist der bekannte RoboCup-Wettbewerb, bei dem die wichtigsten Robotik-Forschungsgruppen ihre Roboterteams gegeneinander Fußball spielen lassen. Bei diesen Mehragentensystemen sollten die Roboter einer Mannschaft in einem kooperativen Verhältnis zueinander stehen, um ein gutes Ergebnis zu erzielen. Die Roboter müssen durch eine intentionsbasierte Interpretation der Aktionen der Mannschaftskollegen deren Ziele und damit übergeordnete Strategien verstehen. Da es sich in einem Team in der Regel um baugleiche Roboter handelt, sind mögliche Aktionen und die potenziellen Intentionshypothesen der Mannschaftskollegen bekannt [Kit97][Sch01]. Diese Informationen liegen bezüglich der gegnerischen Mannschaft nicht vor, sodass die Einordnung der Intention unbekannter und nicht kooperativer Agenten erheblich anspruchsvoller ist [Bee00].

Gofuku & Tanaka [Gof01] entwickelten den *Semantic Information Presentation Agent* (SIPA), um Informationen intentionsbasiert zu visualisieren. Ziel ist die Anzeige ausschließlich der Informationen, die für den Benutzer in der aktuellen Situation interessant sind. Dabei steht sowohl die regelbasierte Ermittlung der Benutzerintention als auch die semantisch und grafisch sinnvolle Aufbereitung der darzustellenden Informationen im Vordergrund.

Das sprachverstehende System NASGRA [Sta97][Mue97] bezieht Syntax und Semantik in die Mustererkennung mit ein und erzeugt auf Basis einer gesprochenen Äußerung eine semantische Gliederung, die schließlich als Grundlage zur weiteren Evaluierung der Äußerung auf Signalebene dient. Im Gegensatz zu dem hier vorgestellten Verfahren wird die ursprüngliche Äußerung rekonstruiert. Dies birgt die in Kapitel 1.1 diskutierten Gefahren in sich. Die rekonstruierte Aussage wird schließlich regelbasiert ausgewertet, sodass das System entsprechend reagieren kann. Die wesentlichen Unterschiede zu dem in Kapitel 3 vorgestellten sprachverstehenden System werden in dem in Kapitel 3.2 diskutiert. Hunsinger [Hun03] hat die Idee der probabilistischen semantischen Decodierung aufgegriffen und für ein multimodales System bestehend aus der handschriftlichen oder sprachlichen Eingabe mathematischer Formeln erweitert.

Die diskutierten Verfahren weisen intentionsbasierte Komponenten auf, die speziell auf die Architektur und auf die Problemstellungen der Systeme zugeschnitten sind. Eine Portierung auf andere Applikationen ist daher nicht oder nur unter sehr hohem Aufwand möglich. Gegenwärtig sind keine vergleichbaren Forschungsarbeiten bekannt, die einen allgemeinen Ansatz zur intentionsbasierten Analyse von Aktionen beschreiben, der die Flexibilität besitzt, ihn beispielsweise sowohl zur Dialogmodellierung als auch zum Sprachverstehen einzusetzen. Darüber hinaus wurden keine Hinweise auf Verfahren gefunden, die der Idee der Intentionsmodelle folgen und Aktionsmerkmale, Syntax, Semantik und die Intentionshypothesen in einer zentralen Komponente statistisch modellieren. Diese beiden Aspekte stellen den Neuigkeitswert des hier vorgestellten Verfahrens zur intentionsbasierten Interpretation von Benutzeraktionen dar. Darüber hinaus ergeben sich aus den vier konkreten Umsetzungen des Ansatzes zahlreiche Innovationen, auf die in den entsprechenden Kapiteln im Detail eingegangen wird.

1.4 Gliederung der Arbeit

In Kapitel 2 wird zunächst der Ansatz zur *Intentionsbasierten Interpretation von Benutzeraktionen* in Architektur und Einzelkomponenten allgemein beschrieben. Dieses Kapitel dient der Erläuterung der Grundidee des Verfahrens sowie der wichtigsten Begriffe. Die nachfolgenden vier Hauptkapitel beschreiben die Umsetzung des Verfahrens für vier unterschiedliche Fragestellungen der Mensch-Maschine-Kommunikation.

Zur besseren Übersicht wird die grobe Gliederung in Abbildung 1.3 grafisch dargestellt. Die Darstellung soll betonen, dass die in den Kapiteln drei bis sechs beschriebenen Systeme konkrete Umsetzungen des in Kapitel 2 vorgestellten allgemeinen intentionsbasierten Grundkonzeptes sind. Alle Systeme sind innovative Beiträge zu ihren Forschungsgebieten.

Die Kapitel 3 bis 6 folgen alle der Struktur von Kapitel 2, um die Verwandtschaft zwischen allen Systemen zu unterstreichen und die direkte Umsetzung der einzelnen Komponenten des intentionsbasierten Ansatzes transparenter darzustellen.

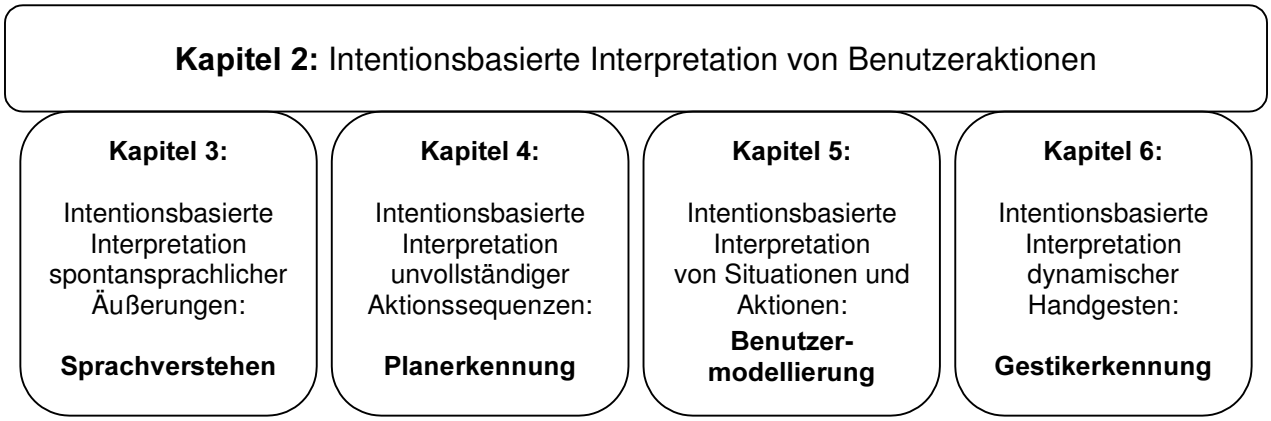


Abbildung 1.3: Überblick über die Kernkapitel

Kapitel 3 behandelt die *Intentionsbasierte Interpretation spontansprachlicher Äußerungen* und erläutert, wie das Verfahren zur syntaktisch-semantischen Evaluierung von Wortkettenhypothesen eingesetzt werden kann. Das resultierende System stellt eine neuartige Herangehensweise an das Thema *Sprachverstehen* dar.

Kapitel 4 beschreibt den Einsatz des Verfahrens zur *Intentionsbasierten Interpretation unvollständiger Aktionssequenzen* mit dem Ziel einer adaptiven Benutzerführung. Bei dem entwickelten Algorithmus handelt es sich um einen probabilistischen *Planerkenner*, der in der Lage ist, auch fehlerhafte Benutzerhandlungen für eine robuste Klassifikation auszuwerten.

Die in Kapitel 5 beschriebene Umsetzung des Verfahrens handelt von der *Intentionsbasierten Interpretation von Situationen und Aktionen* mit dem Ziel, situationsbezogene Aussagen über die Benutzerintention zu treffen. Diese *Benutzermodellierung* findet schließlich Anwendung in der nutzer- und situationsadaptiven Interaktion mit Fahrzeugnavigationssystemen. Das entwickelte Assistenzsystem ist nicht nur in der Lage, Zielorteingaben zu erkennen, sondern auch zu verstehen und zu interpretieren.

In Kapitel 6 wird schließlich mit der *Intentionsbasierten Interpretation dynamischer Handgesten* ein Ansatz zur *Handgestenerkennung* vorgestellt, der sich zum einen durch hohe Flexibilität bei der Wahl des Gestenvokabulars sowie durch die Fähigkeit auszeichnet, sich in Echtzeit an den Benutzer anzupassen.

2

Intentionsbasierte Interpretation von Benutzeraktionen

In diesem Kapitel wird der Ansatz zur *Intentionsbasierten Interpretation von Benutzeraktionen* allgemein vorgestellt und diskutiert. Darüber hinaus werden die wichtigsten Begriffe und Zusammenhänge zum Verständnis des Verfahrens erläutert.

2.1 Grundidee

Voraussetzung für eine effiziente Mensch-Maschine-Interaktion ist die korrekte Interpretation der Benutzeraktionen durch das System, da von einer adäquaten Systemreaktion auf eine Benutzereingabe ohne Kenntnis der Benutzerintention nicht ausgegangen werden kann. Nach einer Einzelaktion oder einer Aktionssequenz des Benutzers liegt der Applikation eine zu analysierende Beobachtungsfolge vor, die mittels Merkmalsextraktion auf die informationstragenden Elemente reduziert wird. Die Merkmalsextraktion ist dabei in erster Linie durch die verwendete Eingabemodalität geprägt, da beispielsweise für die Klassifikation einer Benutzeraktion auf Basis eines Bildsignals völlig andere Charakteristika von Interesse sind als für eine Analyse eines Audiosignals. Der klassische Ansatz zur Ermittlung der Benutzerintention nutzt den resultierenden Merkmalsvektor zur Klassifikation, um somit die Benutzereingabe zu rekonstruieren. Durch Interpretation der rekonstruierten Eingabe wird schließlich auf die eigentliche Absicht des Benutzers geschlossen und die Applikation kann entsprechend reagieren.

In der Praxis erweist sich der klassische Ansatz nur unter idealisierten Bedingungen als effizient, da die von der Applikation zu interpretierenden Eingaben sich häufig als so fehlerbehaftet und verrauscht erweisen, dass ein störender Einfluss auf die Klassifikation dieser Beobachtungsfolgen zu erwarten ist. Dies ist dann der Fall, wenn sowohl Merkmalsextraktion als auch Klassifikation weder durch algorithmische Maßnahmen noch durch Training auf eventuelle Sonderfälle unter den Beobachtungsfolgen vorbereitet sind. Verfälschte Beobachtungsfolgen können bei der Mensch-Maschine-Interaktion eine Reihe von Ursachen haben:

- **Benutzerfehler**
  Benutzer agieren in der Regel nicht optimal. Syntaxfehler, Leichtsinnsfehler, Wiederholungen und Korrekturen von Eingaben führen zu Beobachtungsfolgen, die neben den für eine eindeutige Klassifikation der Intention erwünschten idealen Aktionssequenzen auch Störinformationen beinhalten, die Fehlinterpretationen wahrscheinlicher machen. Vor allem das Verhalten von Benutzern, die mit der zu bedienenden Applikation nicht vertraut sind, ist aus Mangel an Wissen und Erfahrung häufig unvorhersehbar. Darüber hinaus beeinträchtigen äußere Einflüsse und der emotionale Zustand die Konzentration des Benutzers und fördern fehlerhaftes Agieren. Sehr störenden Einfluss auf die Klassifikation der Benutzereingabe haben Bedienfehler, wie zum Beispiel ein zu frühes oder zu spätes Betätigen der Push-to-Talk-Taste zur Aktivierung der Spracherkennung oder das Verlassen des Kamerasichtfeldes bei Eingaben mittels Handgesten.

- **Ungewöhnliche Artikulation**
  Da sich Menschen in ihrem Denken und Handeln durch eine so sehr große Vielfalt auszeichnen, ist es für eine Applikation nahezu unmöglich, alle potenziellen Merkmalskonstellationen, die für bestimmte Intentionen charakteristisch sein können, zu berücksichtigen. Merkmalsextraktion und Klassifikation sind zwar in der Regel auf die gängige Weise, eine bestimmte Absicht zu artikulieren, eingestellt, führen aber bei ungewöhnlichem Benutzerverhaltens zu Fehlinterpretationen der Intention. Ein typisches Beispiel hierfür ist die Analyse von in Dialekt gesprochenen Äußerungen. Ein Spracherkenner, der für hochdeutsche Sprache ausgelegt ist, ist bei starkem Dialekt des Benutzers kaum in der Lage, richtig zu klassifizieren. Problematisch dabei ist die Tatsache, dass die Vielfalt einer Sprache es unmöglich macht, sämtliche Dialekte zu berücksichtigen. Ein weiteres Beispiel, ebenfalls aus der Spracherkennung, ist die Verwendung von Worten, die nicht Teil des Spracherkenner-Vokabulars sind und zu dem bekannten *Out-of-Vocabulary*-Phänomen führen. Generell ist die Art und Weise eines Menschen sich zu artikulieren stark durch den eigenen Kulturkreis bestimmt, was gerade für Applikationen mit dem Anspruch einer möglichst natürlichen Bedienbarkeit zu erheblichen Problemen bei der Klassifikation der Benutzereingabe führt.

- **Ungewöhnliche Rahmenbedingungen**
  Mustererkennungsbasierte Verfahren zur Klassifikation von Benutzeraktionen setzen ein Training des Systems unter den Voraussetzungen, die schließlich im praktischen Betrieb herrschen werden, voraus. Für die Sprachverarbeitung bedeutet dies beispielsweise, dass der Applikation nicht bekannte akustische Störeinflusse, wie zum Beispiel sprechende Personen im Hintergrund, zu Fehlklassifikationen führen. Analog können sich bei der Bildverarbeitung unbekannte oder veränderliche Lichtverhältnisse negativ auf den Klassifikationsprozess auswirken.

Abbildung 2.1 dient der Erläuterung typischer Probleme bei Verwendung des klassischen Ansatzes bestehend aus Merkmalsextraktion, Klassifikation und Intentionsdecodierung. In dem dargestellten Beispiel verfolgt der Benutzer eine konkrete Intention I_4 und teilt der zu bedienenden Applikation entsprechende Kommandos mit. Die inkorrekten Elemente der Beobachtungsfolge **o** und des Merkmalsvektors **m** sind dunkel dargestellt, die fehlerfreien Merkmale sind hell. Die Klassifikation besitzt den Merkmalsvektor als einzige Eingangsgröße und bezieht alle Elemente des Merkmalsvektors mit ein. Somit können „falsche“ Merkmale entscheidenden Einfluss auf das Klassifikationsergebnis nehmen und die Aktionsfolge entsprechend verfälscht rekonstruieren. Die

ergebnis nehmen und die Aktionsfolge entsprechend verfälscht rekonstruieren. Die Intentionsdecodierung dieser Aktionsfolge wird somit eine falsche Intention zum Ergebnis haben.

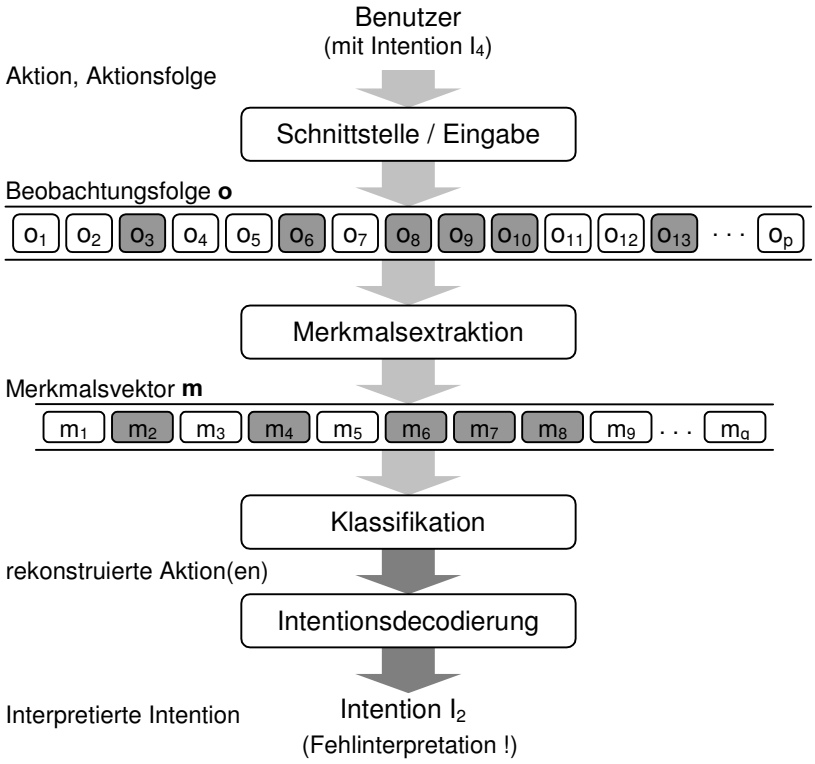


Abbildung 2.1: Klassische Interpretation von Benutzeraktionen

Die drei beschriebenen Hauptursachen für Fehlinterpretationen von Benutzeraktionen werden sich mit herkömmlichen Verfahren zur Mustererkennung nicht beheben lassen. Benutzerfehler kann man zwar durch möglichst intuitive, einfach erfassbare Schnittstellen in Grenzen halten, es wird jedoch nicht möglich sein, Fehlverhalten auszuschließen. Der störende Einfluss untypischer Artikulation lässt sich nur durch Adaption der Mustererkennung an den aktuellen Benutzer reduzieren. Systeme, die alle potenziellen Benutzereigenheiten berücksichtigen, sind aus Komplexitätsgründen nach dem derzeitigen Stand der Technik nicht in Sicht. Ebenso wird es im praktischen Einsatz mustererkennender Verfahren immer wieder Rahmenbedingungen geben, auf die das Verfahren weder algorithmisch noch durch Training vorbereitet ist.

Zusammenfassend kann man sagen, dass sich fehlerhafte Merkmalsvektoren nicht verhindern lassen und somit stets die Gefahr einer nicht korrekten Rekonstruktion des Benutzerverhaltens besteht. Dies führt zu einem neuartigen – in dieser Arbeit vorgestellten – Ansatz, der eine exakte Rekonstruktion der Benutzeraktion überflüssig macht, da nicht deren Kenntnis, sondern der Inhalt für die Steuerung einer Applikation entscheidend ist.

Je nach Eingabemodalität und Applikation hat die Intention einen unterschiedlichen Stellenwert. Die Bandbreite reicht dabei von übergeordneten, längerfristigen Zielen, die durch komplexe Aktionssequenzen zu erreichen sind, bis hin zu Intentionen, die anhand einer einfachen Einzelaktion vermittelt werden können. Um dieser Vielfalt mit dem intentionsbasierten Ansatz gerecht zu werden, wurden zwei Ausprägungen des Verfahren entwickelt, die beide der gleichen Grundidee und

Grundstruktur folgen, sich aber in einigen Details unterscheiden. Beide Ansätze werden nun erläutert.

Die erste Ausprägung des intentionsbasierten Ansatzes ist darauf ausgerichtet, die anhand von Abbildung 2.1 diskutierten fehlerhaften Merkmale als Störinformation zu erkennen und diese für die Klassifikation zu ignorieren. Hierfür wird auf die Rekonstruktion der Benutzereingabe verzichtet und die Merkmale der Aktion direkt mit den potenziellen Intentionen in Bezug gesetzt. Daraus resultiert die Fähigkeit der Applikation, eine Beobachtungsfolge nicht nur zu erkennen, sondern vor allem zu verstehen. Vorbild für diese Systemeigenschaft ist der Mensch mit seiner Fähigkeit, mittels Transferdenken und Wissen um das aktuelle Gesprächsthema auf den Inhalt einer Aussage zu schließen, ohne dass diese vollständig ist. Da die erste Ausprägung des vorgestellten Ansatz für die Interaktion mit in ihrem Umfang begrenzten Applikationen bestimmt ist, ist hierzu lediglich die Kenntnis aller potenziellen Intentionen eines Benutzers im Umgang mit diesem System notwendig.

Der Algorithmus geht immer von der möglichen Intention aus und sucht den Merkmalsvektor nach den Elementen ab, die für die betrachtete Intention charakteristisch sind. Diese intentionsbasierte Interpretation von Benutzereingaben bietet somit den Vorteil einer intelligenten Auswahl der Merkmale, die für die betrachtete Intentionshypothese relevant sind. Dieses Phänomen wird im Folgenden als *Merkmalsselektion* bezeichnet. Alle übrigen Merkmale sind für die betrachtete Intentionshypothese Störinformationen. Sie werden für die Klassifikation nicht berücksichtigt und haben somit keinen verfälschenden Einfluss auf den Klassifikationsprozess.

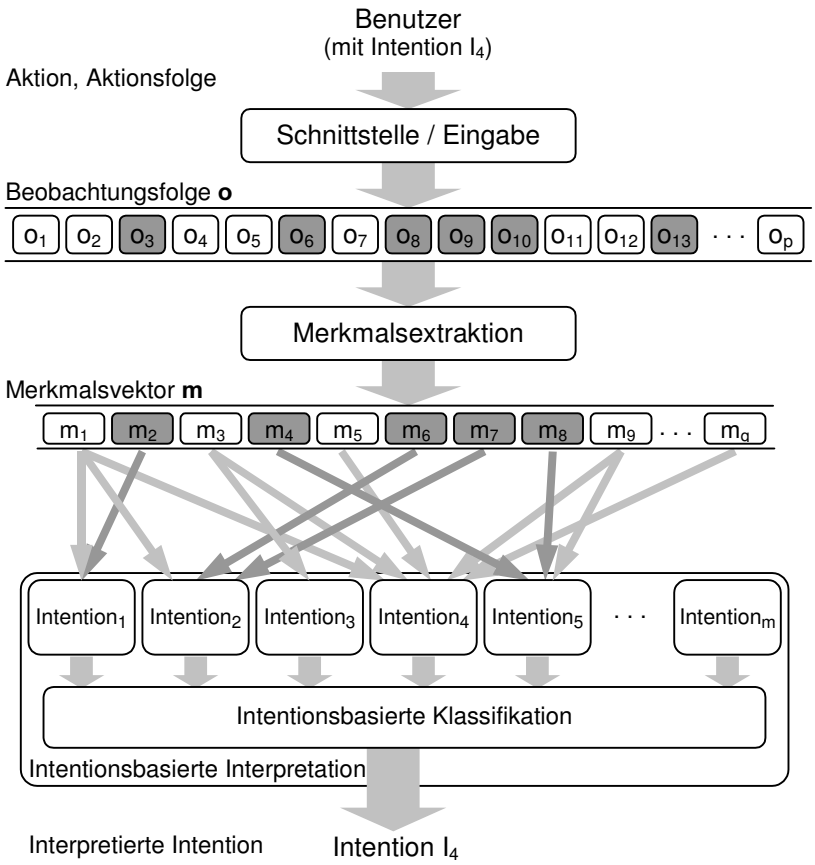


Abbildung 2.2: Intentionsbasierte Interpretation von Benutzeraktionen durch Merkmalsselektion

Die Grundidee der Merkmalsselektion wird in Abbildung 2.2 anhand eines Beispiels erläutert. Der Benutzer verfolgt die Intention I_4 und verhält sich mit seinen Eingaben entsprechend. Durch die oben erwähnten Ursachen könnte die Beobachtungsfolge und somit auch der Merkmalsvektor verrauscht sein. Die resultierenden Störinformationen sind dunkel visualisiert. Der Merkmalsvektor wird jetzt intentionsbasiert analysiert, d.h., ausgehend von einer Intentionshypothese werden ausschließlich deren charakteristische Merkmale für die Klassifikation herangezogen. Die Aufgabe der intentionsbasierten Klassifikation ist die Bewertung aller Intentionshypothesen auf Basis des Merkmalsvektors. Welche Merkmale die einzelnen Intentionshypothesen zur Klassifikation auswerten, zeigen die hellen Pfeile an. Die Nutzinformationen werden für die Bewertung der Hypothese I_4 genutzt, die Störinformationen sind entsprechend der Abbildung Merkmale, die für andere Intentionshypothesen bezeichnend sein können. Generell können Merkmale für die Bewertung mehrerer Intentionshypothesen ausgewertet werden. Die Intentionshypothese, die am besten durch den Merkmalsvektor modelliert wird, ist schließlich das Ergebnis des Interpretationsprozesses.

Dieser Algorithmus muss syntaktisch-semantische Zusammenhänge von Merkmalen und Intentionshypothesen effizient modellieren um aus einer Reihe von Merkmalsbeobachtungen Synergieeffekt erzielen zu können. Die hierfür verwendeten Intentionsmodelle werden in Kapitel 2.5 erläutert.

Diese Ausprägung ist in der Hauptsache für übergeordnete Ziele, deren Einzelaktionen syntaktisch-semantisch abhängig voneinander sind, interessant. Ist es möglich, das Ziel mit nur einer einzigen Aktion umsetzen, so können keine syntaktisch-semantischen Beziehungen aufeinanderfolgender Eingaben gewinnbringend ausgewertet werden. In diesem Fall hat bereits die Klassifikation direkt das Benutzerziel zum Ergebnis und eine Komponente zur Intentionsdecodierung wird somit überflüssig. Ein Beispiel hierfür ist zum Beispiel die Zieleingabe bei einem Fahrzeugnavigationssystem. Der Fahrer spricht oder tippt den gewünschten Ort und macht somit unmittelbar seine Absicht deutlich. Die durch die Klassifikation rekonstruierte Aktion ist mit der Intention äquivalent. Die diskutierten Phänomene, die zu fehlerhaften Elementen in den Merkmalsvektoren führen, können dadurch Fehlinterpretationen der Benutzerabsicht zur Folge haben. Auch in diesem Fall wäre ein Verzicht auf eine rein signalbasierte Rekonstruktion der Benutzereingabe also sinnvoll. Hierfür wurde die zweite Ausprägung des intentionsbasierten Ansatzes entwickelt.

Die zweite Ausprägung des intentionsbasierten Ansatzes kombiniert die in Abbildung 2.1 und Abbildung 2.2 dargestellten Ideen. Abbildung 2.3 visualisiert den Grundgedanken. Zunächst findet eine „klassische“ Klassifikation statt, wobei keine Entscheidung bezüglich der wahrscheinlichsten Intention getroffen wird, sondern eine Liste der n besten Intentionshypothesen generiert wird. Diese Informationen sind schließlich die Eingangsgrößen für die intentionsbasierte Interpretation, die auf Basis von Kontextwissen über Benutzerintentionen das wahrscheinlichste Ziel des Benutzers bestimmt. Dieses Hintergrundwissen wird durch die Intentionsmodelle erfasst, die benutzertypische syntaktisch-semantische Beziehungen zwischen Merkmalen und Intentionen erlernen. Die Idee besteht somit darin, den Hypothesenraum durch die Klassifikation einzuschränken und die Entscheidung für eine bestimmte Intentionshypothese wissensbasiert bzw. intentionsbasiert zu treffen. Somit ermöglicht auch diese Variante die Realisierung eines *verstehenden* Systems.

Beide Ausprägungen des intentionsbasierten Ansatzes folgen der selben Grundstruktur. Darauf und auf Unterschiede in der Realisierung der Algorithmen wird im folgenden Kapiteln eingegangen.

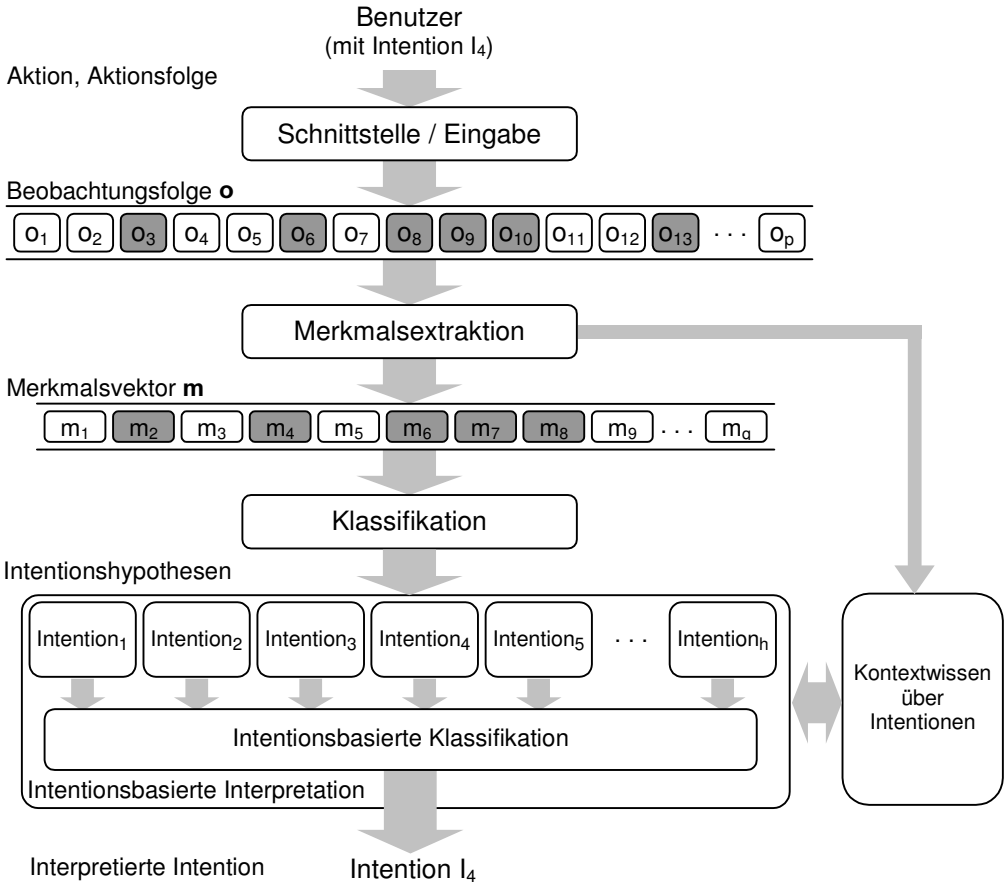


Abbildung 2.3: Intentionsbasierte Interpretation von Benutzeraktionen durch Berücksichtigung von Kontextwissen

2.2 Struktur zur intentionsbasierten Ansatzes

Der intentionsbasierte Ansatz zur Interpretation von Benutzereingaben wird der zentralen Rolle der Benutzerintention dadurch gerecht, dass die auf aussagekräftige Merkmale reduzierten Benutzeraktionen hinsichtlich potenzieller Intentionen analysiert werden. Hierfür wurde eine grundlegende Systemarchitektur entwickelt, die weitestgehend applikations- und modalitätenunabhängig ist und für beide Varianten des Ansatzes gültig ist. Abbildung 2.4 stellt die Systemarchitektur zur intentionsbasierten Evaluierung von Benutzeraktionen dar.

Die einzelnen Komponenten werden in den folgenden Abschnitten im Detail erläutert. An dieser Stelle steht ein allgemeiner Überblick über das Zusammenwirken dieser Einzelkomponenten im Vordergrund.

Die Intentionsbibliothek **I** beschreibt den vollständigen Intentionshypothesenraum des Verfahrens und enthält das Wissen über potenzielle Benutzerziele. Der gesamte Hypothesenraum wird durch Intentionsmodelle repräsentiert, die das Bindeglied zwischen den Intentionshypothesen und den aus der Beobachtungsfolge hervorgehenden Merkmalen darstellen. Sie enthalten das gesamte Wissen, das für den Verstehensprozess zur Verfügung steht.

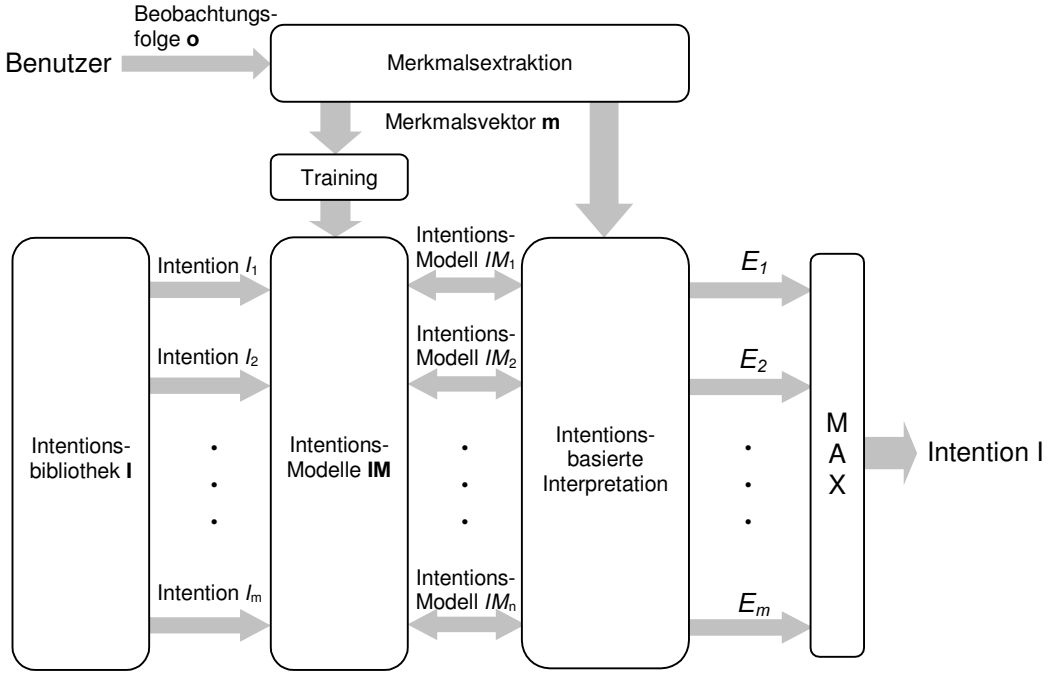


Abbildung 2.4: Systemarchitektur zur intentionsbasierten Interpretation von Benutzeraktionen

Benutzeraktionen erzeugen eine Beobachtungsfolge **o**, die anhand der Merkmalsextraktion auf die entscheidenden Informationen reduziert wird. Je nach Applikation kann der resultierende Merkmalsvektor **m** dem Training der Intentionsmodelle oder der Interpretation der Beobachtungsfolge dienen. Für die intentionsbasierte Interpretation wird der Merkmalsvektor auf die Intentionsmodelle abgebildet. Wie gut die Merkmale die einzelnen Intentionshypothesen repräsentieren, wird anhand des Evaluierungsergebnisses E quantitativ beschrieben. Das Ergebnis des Interpretationsprozesses ist schließlich die Intention mit dem maximalen Evaluierungsmaß.

Die Einzelkomponenten des Verfahrens werden in den nachfolgenden Abschnitten vorgestellt beginnend mit der Intentionsbibliothek.

2.3 Intentionsbibliothek

Die Intentionsbibliothek **I** beinhaltet alle für die Evaluierung von Aktionen in Betracht zu ziehenden Intentionen, d.h. alle potenziellen Absichten, die der Benutzer im Umgang mit der Applikation verfolgen kann. Sie umfasst das gesamte Wissen des verstehenden Systems und deckt alle möglichen Aktionsinhalte ab. Da davon ausgegangen wird, dass der Benutzer in der Regel nicht mehrere Ziele gleichzeitig verfolgt, entspricht immer genau ein Eintrag der Intentionsbibliothek der aktuellen Absicht des Benutzers. Zum Zeitpunkt der Erstellung der Intentionsbibliothek sind zunächst alle Intentionen gleich wahrscheinlich, d.h., es folgt keine a-priori-Gewichtung einzelner Einträge. Im Rahmen der intentionsbasierten Evaluierung können aber kontextabhängig Gewichtungen getroffen oder einzelne Intentionen ausgeschlossen werden.

Die Intentionsbibliothek kann sowohl kurzfristige Absichten, die dem System mittels einer Einzelaktion mitzuteilen sind, also auch semantisch hochwertige, übergeordnete Ziele, die nur mittels Ak-

tionssequenzen erreichbar sind, umfassen. Da sich die Komplexität des Funktionsumfangs für eine abgeschlossene Applikation, wie zum Beispiel ein Telefon, in der Regel in Grenzen hält, wird das Ansprechen jeder einzelnen Systemfunktion als Intention modelliert. Die Intentionsbibliothek entspricht somit der Liste der Systemfunktionen, auf die der Benutzer Einfluss nehmen kann. Wie eine Intentionsbibliothek mit bis zu 100000 Einträgen berücksichtigt werden kann, wird in Kapitel 5 deutlich.

Da es das Ziel des Verfahrens ist, aus den Einträgen der Intentionsbibliothek auf Basis von Beobachtungen die wahrscheinlichste Intention zu ermitteln, werden diese auch als *Intentionshypothesen* bezeichnet.

Die Intentionsbibliothek **I** für m Intentionshypothesen wird als Vektor definiert und durch folgende Nomenklatur beschrieben:

$$\mathbf{I} = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_{\mathrm{m}} \end{pmatrix} \tag{2-1}$$

Der Benutzer kann seine Intention durch Eingaben verwirklichen, die schließlich der Applikation als Beobachtungsfolgen vorliegen, auf deren Basis die Merkmalsextraktion Charakteristika bestimmt. Folgender Abschnitt behandelt die Merkmalsextraktion.

2.4 Merkmalsextraktion

Bei der Beobachtungsfolge **o** handelt es sich um die von der Applikation erfasste Benutzeraktion. Die Aufgabe der Merkmalsextraktion ist das Herausfiltern überflüssiger Informationen aus der Beobachtungsfolge, um somit eine Reduktion auf die charakteristischen, informationstragenden Elemente zu erreichen. Je nach Variante des Verfahrens es dabei zwei unterschiedliche Varianten von Merkmalsvektoren geben:

- Der Merkmalsvektor **m** besteht aus einer Abfolge von Merkmalsbeobachtungen, die entsprechend der Reihenfolge ihres Auftretens beschrieben werden. Die Elemente des Vektors sind somit beobachtete Ereignisse aus dem Raum aller möglichen Ereignisse. Diese Beschreibung ist in erster Linie für die Merkmalsselektion interessant. Im Rahmen der Merkmalsextraktion nicht beobachtete Ereignisse bleiben somit unberücksichtigt. Für den Fall, dass die Merkmalsextraktion fünf Merkmale von 20 potenziellen Merkmalsbeobachtungen (m_1 bis m_{20}) betrifft, könnte dieses Szenario folgenden Vektor ergeben, dessen Dimension von Fall zu Fall variieren kann:

$$\mathbf{m} = \begin{pmatrix} m_{16} \\ m_3 \\ m_{12} \\ m_9 \\ m_2 \end{pmatrix} \tag{2-2}$$

- Die Alternative zu den oben erläuterten Merkmalsvektoren ist eine klare Beschreibung der Beobachtungsfolge **o** anhand einer definierten Anzahl von Zustandsvariablen, deren Zustände schließlich den Inhalt des Vektors **m** bilden. Interessant ist diese Art der Modellierung vor allem für lernende Verfahren, da der Merkmalsvektor **m** als klar definierter, unter Umständen auch unvollständiger Datensatz herangezogen werden kann. Die Dimension von **m** ist durch die Anzahl der Zustandvariablen geprägt und somit stets konstant:

$$\mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_q \end{pmatrix} \tag{2-3}$$

Welche Merkmale im Speziellen aussagekräftig für eine Beobachtungsfolge sind, hängt stark von der zu lösenden Aufgabenstellung und damit von den Eingangssignalen ab. Die erste Form eines Merkmalsvektors wird in dieser Arbeit für die erste Ausprägung des Verfahrens herangezogen, während die zweite Darstellung von Beobachtungen für die zweite Variante des Ansatzes verwendet wird. Da an dieser Stelle keine weiteren allgemeinen Aussagen zur Merkmalsextraktion möglich sind, wird dieser Aspekt in den folgenden Kapiteln für jedes bearbeitete Gebiet der Mensch-Maschine-Interaktion sehr detailliert beschrieben.

Auf Basis des Merkmalvektors **m** wird die intentionsbasierte Interpretation der Beobachtungsfolge vorgenommen. Hierfür werden die Merkmale mit den Intentionshypothesen mit Hilfe der so genannten *Intentionsmodelle* in Bezug gesetzt.

2.5 Intentionsmodelle

Die *Intentionsmodelle* **IM** bilden den Klassifikator des Verfahrens und dienen als Bindeglied zwischen den Merkmalsvektoren **m** und den Intentionshypothesen **I**. Aus diesem Grund kommt ihnen die tragende Rolle für die intentionsbasierte Interpretation von Benutzeraktionen zu. Sie modellieren das gesamte Wissen über potenzielle Benutzerintentionen und ermöglichen dadurch das *Verstehen* von Benutzeraktionen.

Für jede Intentionshypothese sind bestimmte Merkmale charakteristisch, die im Rahmen der intentionsbasierten Interpretation auf das entsprechende Intentionsmodell abgebildet werden. Wie im vorherigen Abschnitt erläutert, gibt es sowohl Merkmalsvektoren, deren Dimension variieren kann, als auch Vektoren, die bei konstanter Dimension eventuell unvollständig sind. Aus diesem Grund benötigt ein Intentionsmodell die Fähigkeit, unvollständige Information zu verarbeiten, um eine intentionsbasierte Merkmalsselektion zu gewährleisten.

Neben der Verarbeitung unvollständiger Information ist die Fähigkeit des Intentionsmodells, unsichere Informationen zu verarbeiten, ausschlaggebend, da sich Merkmale in der Regel als unterschiedlich charakteristisch für bestimmte Intentionen erweisen. Ein Intentionsmodell muss daher in der Lage sein, den Einfluss bzw. den Wert eines Merkmals für eine Intention quantitativ zu gewichten.

Die mathematische Grundlage der Intentionsmodelle bilden *Bayes'sche Netze* [Rus94][Pea88] [Jen92], die mit der Möglichkeit, unvollständige und unsichere Informationen zu modellieren, ideale Voraussetzungen für die intentionsbasierte Interpretation bieten. Bayes'sche Netze bestehen aus Knoten zur Visualisierung von Zustandsvariablen und Kanten, um die Informationen, die den entsprechenden Zustandsvariablen zugeordnet sind, in einen qualitativen Zusammenhang zu stellen. Die Gewichtung der Kanten mittels statistischer Wahrscheinlichkeiten ermöglicht eine quantitative Modellierung von Korrelationen zwischen verschiedenen Informationen.

Im Rahmen der Intentionsmodelle ermöglichen die Bayes'schen Netze die Modellierung komplexer syntaktisch-semantischer Beziehungen zwischen Merkmalen und Intentionshypothesen. Da die Topologie eines Bayes'schen Netzes im Detail erheblich durch die zu lösende Aufgabenstellung geprägt ist, werden in diesem Kapitel lediglich die grundlegende Struktur und die Idee behandelt. In den folgenden Hauptkapiteln wird dann ausführlich erläutert, wie ausgehend von dieser allgemeinen Topologie konkrete Intentionsmodelle erzeugt werden.

Abbildung 2.5 zeigt eine allgemeine Struktur eines Intentionsmodells. Die Struktur ist hierarchisch in drei Ebenen gegliedert, wobei diese Ebenen nicht die Funktionsweise von Teilnetzwerken, sondern die Aufgabe der einzelnen Knoten und deren Abhängigkeiten beschreiben. Generell kann jede Intentionshypothese anhand eines oder mehrerer Bayes'schen Netze repräsentiert werden. Ebenso ist es möglich, alle Intentionshypothesen auf Basis eines einzigen Netzes zu berücksichtigen. Der in Abbildung 2.5 gezeigte Netzausschnitt ist somit als exemplarisch zu betrachten.

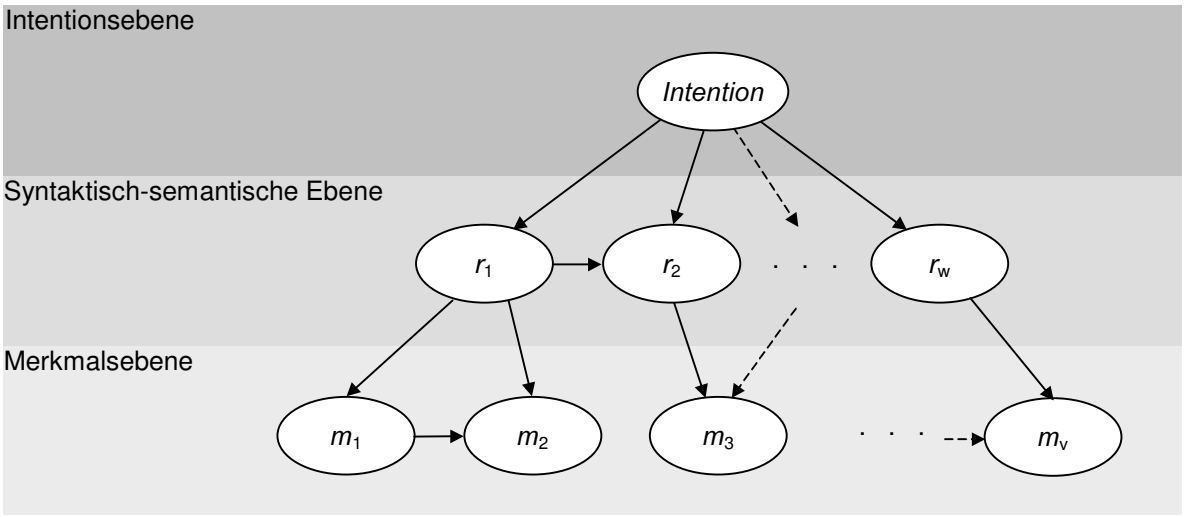


Abbildung 2.5: Allgemeine Struktur eines Intentionsmodells

Die drei Ebenen haben folgende Bedeutungen:

- **Intentionsebene**
  Auf der Intentionsebene sind alle Intentionshypothesen repräsentiert, die für die intentionsbasierte Interpretation berücksichtigt werden sollen. Je nachdem, ob alle Intentionshypothesen durch ein einziges Netz oder jede Hypothese durch jeweils ein Netz repräsentiert wird, besteht diese Ebene aus einem Knoten mit jeweils einem Zustand pro Hypothese oder aus einem Knoten mit Boole'schem Zustandsraum. Generell werden diese Knoten als *Intentionsknoten* be-

zeichnet. Dabei handelt es sich stets um Wurzelknoten, die wahrscheinlichkeitsbasierte Aussagen über Intentionshypothesen ermöglichen.

- **Syntaktisch-semantische Ebene**
  Die syntaktisch-semantische Ebene umfasst die statistischen Zusammenhänge zur Umsetzung des Klassifikationsmechanismus und ist daher stark von der lösenden Aufgabe bestimmt. Dieser Netzwerkteil verbindet die Merkmalsebene mit der Intentionsebene, indem Merkmalsbeobachtungen durch probabilistisches Schließen verarbeitet und bezüglich der mit dem Intentionsmodell korrespondierenden Intention evaluiert werden können.
  Da die Topologie eines Bayes'schen Netzes eine qualitative Beschreibung von Zusammenhängen erlaubt, eignen sich Bayes'sche Netze hervorragend zur einfachen und effizienten Integration von Wissen über die zu modellierende Domäne. Die Fähigkeit der Netze, semantische Zusammenhänge darzustellen, kann somit direkt für die Intentionsmodellierung genutzt werden. Dies führt allerdings dann zu Problemen, wenn Teilhypothesen unabhängig voneinander evaluiert werden sollen. Für diese Teilbetrachtungen sind dann eigene Netze vorzusehen, da sich eine Entkopplung der statistischen Abhängigkeiten zwischen verschiedenen Netzteilen kaum oder gar nicht bewerkstelligen lässt.

  Falls die Wahrscheinlichkeitstheorie nicht mächtig genug ist, um komplexere mathematische Zusammenhänge zu beschreiben, geschieht dies außerhalb eines Netzes auf Basis von Wahrscheinlichkeiten, die aus dem jeweiligen Netz abgeleitet werden. Ein Intentionsmodell besteht somit neben einem oder mehreren Bayes'schen Netzen unter Umständen auch aus einer Reihe zusätzlicher mathematischer Zusammenhänge.

- **Merkmalsebene**
  Die Merkmalsebene ist die unterste Ebene eines Bayes'schen Netzes zur Intentionsmodellierung. Sie besteht aus den so genannten *Merkmalsknoten*, die Beobachtungen bestimmter Merkmale repräsentieren. Damit ist die Merkmalsebene die einzige Schnittstelle eines Intentionsmodells zur Merkmalsextraktion.

Die Idee der Intentionsmodellierung besteht darin, durch die Topologie und die bedingten Wahrscheinlichkeiten von syntaktisch-semantischer Ebene und Merkmalsebene ein Intentionsreferenzmodell zu schaffen, das einem Ideal der zu repräsentierenden Intention entspricht. Hierfür werden die drei Ebenen eines Intentionsmodells unter Umständen auch anhand mehrerer, miteinander kommunizierender Netzwerke realisiert. Wie die Topologie und die bedingten Wahrscheinlichkeiten für eine konkrete Aufgabenstellung umgesetzt werden, wird in den Kapiteln 3 bis 6 im Detail dargestellt.

Auf die Zustandsräume der Knoten des exemplarischen Netzwerks aus Abbildung 2.5 wurde bisher nicht eingegangen, da im Rahmen der vorliegenden Arbeit zwei Ausprägungen des intentionsbasierten Ansatzes mit jeweils unterschiedlichen Intentionsmodellen untersucht wurden, die sich auch in den Zustandsräumen der Knoten unterscheiden. Beide Typen von Intentionsmodellen werden nun anhand von Abbildung 2.6 vorgestellt.

Abbildung 2.6 zeigt in der Mitte den Vektor **I** der Intentionsbibliothek für m Intentionshypothesen. Ziel der Intentionsmodelle ist die vollständige Repräsentation dieses Vektors. Hierfür werden zwei verschiedene Möglichkeiten untersucht:

- **Repräsentation jeder einzelnen Intentionshypothese durch jeweils ein Intentionsmodell**
  Bei den links dargestellten Bayes'schen Netzen handelt es sich um Intentionsmodelle, die jeweils nur eine Intentionshypothese darstellen. Die Intentionsknoten sind Boole'sche Zustandsvariablen, die Aussagen darüber ermöglichen sollen, wie vollständig der Merkmalsvektor die durch das Intentionsmodell dargestellte Intention modelliert. Hierfür wird die erste Variante der in Kapitel 2.4 vorgestellten Merkmalsvektoren verwendet. Die Merkmalsknoten entsprechen ebenfalls Boole'schen Zustandsvariablen und haben die Aufgabe, die Beobachtung der für die betrachtete Intention charakteristischen Merkmale zu erfassen. Somit wird in der Regel nur ein Teil des Merkmalsvektors auf ein Intentionsmodell abgebildet, da die übrigen Merkmale für Aussagen über die betrachtete Intention unberücksichtigt bleiben. Dadurch wird eine effiziente Merkmalsselektion gewährleistet. Diese Art Intentionsmodell ist vor allem dann interessant, wenn Intentionshypothesen unabhängig von den übrigen Einträgen der Intentionsbibliothek bewertet werden sollen. Dabei kann ein derartiges Intentionsmodell auch durch mehrere Bayes'sche Netze realisiert sein.

- **Repräsentation der gesamten Intentionsbibliothek durch ein einziges Intentionsmodell**
  Das rechts dargestellte Bayes'sche Netz ist in der Lage, alle Intentionshypothesen zu beurteilen, indem der Zustandsraum des Intentionsknotens einen Zustand für jede Hypothese bereithält. Die Anzahl der Merkmalsknoten entspricht dabei exakt der Dimension des Merkmalsvektors **m**, der im Rahmen der intentionsbasierten Evaluierung auf die Merkmalsebene abgebildet werden kann. Somit findet in diesem Fall die in Kapitel 2.4 vorgestellte zweite Variante von Merkmalsvektoren Verwendung. Die Zustandsräume der Merkmalsknoten können individuell gewählt werden, je nachdem, welche Ausprägungen ein bestimmtes Merkmal mit sich bringt. Der Vorteil dieser Art der Intentionsmodellierung liegt in der diskriminativen Klassifikation. Diese Modellierung ist im Falle rivalisierender bzw. sich gegenseitig ausschließender Intentionshypothesen interessant. Eine intentionsbasierte Merkmalsselektion ist mit dieser Art Intentionsmodell nicht oder nur durch gesonderte Maßnahmen möglich.

Werden alle Einträge der Intentionsbibliothek durch ein Intentionsmodell erfasst, so wird von dem Intentionsmodell *IM* gesprochen. Wird hierfür eine Reihe von Intentionsmodellen verwendet, so werden diese durch folgenden Vektor **IM** zusammengefasst:

$$\mathbf{IM} = \begin{pmatrix} IM_1 \\ IM_2 \\ \vdots \\ IM_{\mathrm{m}} \end{pmatrix} \tag{2-4}$$

Grundsätzlich wird ein Intentionsmodell in Struktur und Wahrscheinlichkeiten so definiert, dass es möglichst alle potenziellen Wege berücksichtigt, die entsprechende Intention durchzusetzen. Dadurch entspricht ein Intentionsmodell einem idealisierten Referenzmodell, das mit den realen Beo-

bachtungen verglichen wird. Dies geschieht mittels der intentionsbasierten Interpretation, die im nächsten Abschnitt erläutert wird.

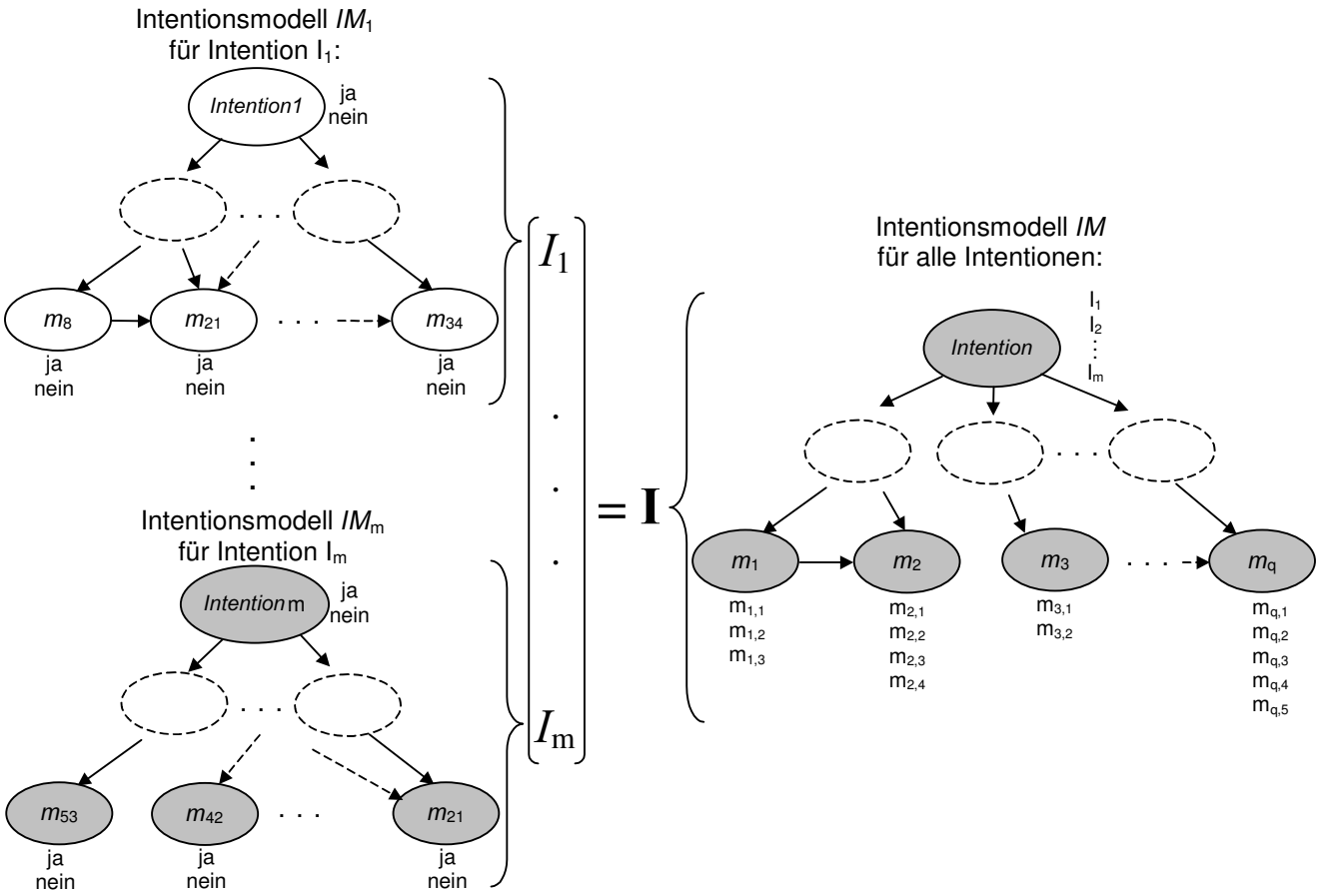


Abbildung 2.6: Zwei Ausprägungen von Intentionsmodellen

Die Erläuterung der Struktur eines Intentionsmodells wurde bewusst allgemein und knapp gehalten, da die konkrete Realisierung eines Intentionsmodells durch ein oder mehrere Bayes'sche Netze abhängig vom Einsatzgebiet des Ansatzes ist. Aus diesem Grund bildet die Umsetzung der Intentionsmodelle in den folgenden Kapiteln 3 bis 6 jeweils einen Schwerpunkt.

2.6 Intentionsbasierte Interpretation

Mit der Definition der Intentionsbibliothek, einer adäquaten Merkmalsextraktion und der Bereitstellung der Intentionsmodelle sind alle Voraussetzungen für die eigentliche Klassifikation, der intentionsbasierten Interpretation, geschaffen.

Vor jedem Interpretationsprozess können die Intentionshypothesen kontextabhängig unterschiedlich gewichtet werden. Da es sich bei den Intentionsknoten stets um Wurzelknoten handelt, müssen den entsprechenden Zustandvariablen lediglich die gewünschten Gewichtungen als a-priori-Wahrscheinlichkeiten $P(Intention)$ zugewiesen werden.

Die intentionsbasierte Interpretation unterscheidet sich bei den beiden vorgestellten Ausprägungen des Verfahrens in einigen Aspekten. Aus diesem Grund wird diese Systemkomponente für die beiden Ansätze zur Intentionsmodellierung nacheinander behandelt.

Die Aufgabe der intentionsbasierten Interpretation von Benutzeraktionen besteht darin, den Merkmalsvektor mit den Intentionsmodellen derart in Bezug zu setzen, sodass die wahrscheinlichste Intentionshypothese berechnet werden kann. Für die erste Variante des Verfahrens bzw. für Intentionsmodelle, die jeweils eine Intentionshypothesen repräsentieren, ist in Abbildung 2.7 das Abbilden des Merkmalsvektors **m** auf die Bayes'schen Netze dargestellt.

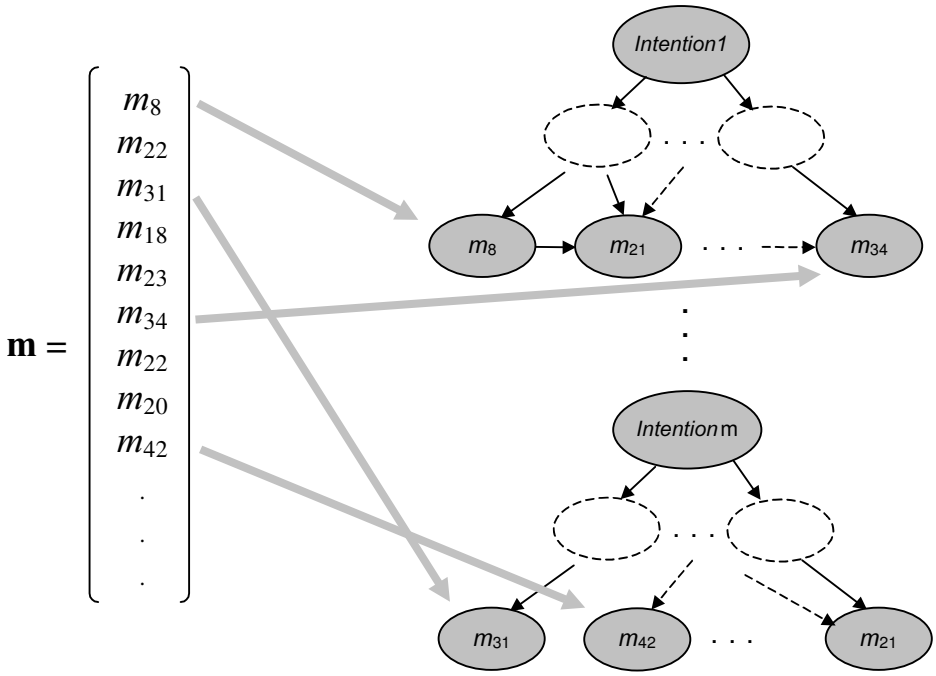


Abbildung 2.7: Abbilden des Merkmalsvektors auf die Merkmalsknoten von Intentionsmodellen, die jeweils eine Intentionshypothese repräsentieren

Die Abbildung zeigt links den Merkmalsvektor und rechts die Intentionsmodelle. Die Elemente des Merkmalsvektors werden nun mit den Merkmalen, die durch die einzelnen Merkmalsknoten repräsentiert werden, verglichen. Hierfür stehen jedem Merkmalsknoten zusätzliche Informationen zu Verfügung, die gewährleisten sollen, dass nur vollständige Übereinstimmungen erfasst werden. Dies ermöglicht, Bedingungen bezüglich der Reihenfolge von Beobachtungen zu stellen. Im Falle einer Übereinstimmung wird die Beobachtung des Merkmals im Intentionsmodell durch *Instanziierung* der korrespondierenden Zustandsvariable berücksichtigt, d.h., ihr wird ein konkreter Zustand zugewiesen. Dabei handelt es sich um den ja-Zustand. Alle anderen Merkmalsknoten bleiben unberücksichtigt. Auf diese Weise wird die intentionsbasierte Merkmalsselektion erreicht.

Nachdem das beschriebene Procedere für alle in Betracht zu ziehenden Intentionsmodelle durchgeführt wurde, müssen die durch Merkmalsbeobachtungen beaufschlagten Intentionsmodelle quantitativ evaluiert werden, um die wahrscheinlichste Intention zu ermitteln. Für jedes Intentionsmodell wird hierfür ein Evaluierungswert E berechnet, der den beobachteten Merkmalen Rechnung trägt. Dabei handelt es sich um die Wahrscheinlichkeit des ja-Zustands des Intentionsknotens mit allen für die entsprechende Intention beobachteten Merkmalen:

$$E = P(Intention = \text{ja} \mid \mathbf{m}) \tag{2-5}$$

Die Evaluierungsmaße werden für alle Einträge der Intentionsbibliothek berechnet:

$$\mathbf{E} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{pmatrix} \tag{2-6}$$

Das Ergebnis der intentionsbasierten Interpretation ist schließlich die Intentionshypothese mit dem größten Evaluierungsmaß.

Falls es sich bei einigen Intentionshypothesen um Untermengen anderer Intentionshypothesen handelt, sind Normierungen der Evaluierungsmaße notwendig. Dies wird in den folgenden Kapiteln an den entsprechenden Stellen ausführlich erläutert.

Für ein Intentionsmodell, das alle Intentionshypothesen repräsentiert, ist in Abbildung 2.8 das Abbilden des Merkmalsvektors **m** auf das Bayes'sche Netz dargestellt. Die Zustandsvariablen nehmen die Zustände ein, die ihnen durch den Merkmalsvektor vorgegeben werden. Dabei müssen die Datensätze nicht notwendigerweise vollständig sein, d.h., nicht jedem Merkmalsknoten muss ein konkreter Zustand zugewiesen werden.

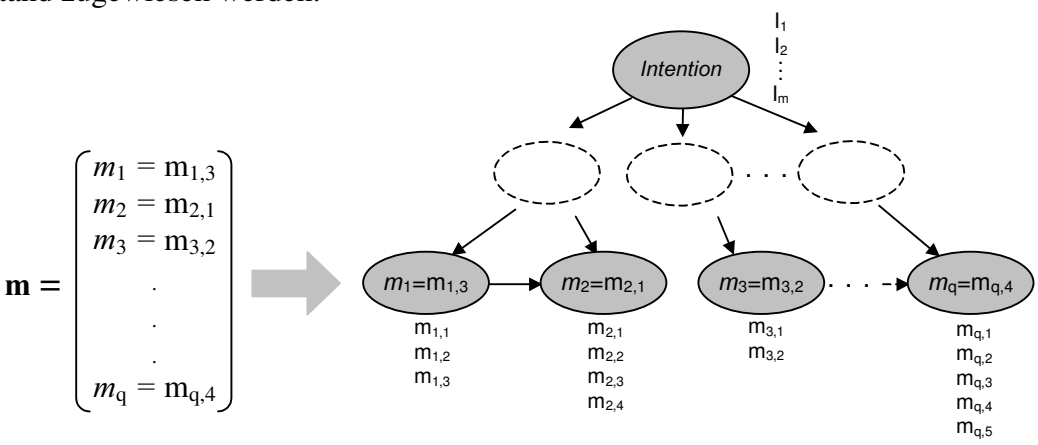


Abbildung 2.8: Abbilden des Merkmalsvektors auf die Merkmalsknoten eines Intentionsmodells, das alle Intentionshypothesen repräsentiert

Der Evaluierungswert einer Intention I_x kann schließlich über folgende a-posteriori-Wahrscheinlichkeit berechnet werden:

$$E_x = P(Intention = I_x \mid \mathbf{m}) \tag{2-7}$$

Das Ergebnis der intentionsbasierten Interpretation von Benutzeraktionen ist schließlich die Intention mit dem höchsten Evaluierungswert E. Da der Intentionsknoten bei der diskutierten Modellstruktur bereits alle Intentionshypothesen in Bezug setzt, erhält man durch folgende Maximum-a-posteriori-Beziehung direkt das Ergebnis der intentionsbasierten Interpretation von Benutzeraktionen, die Benutzerintention I:

$$\mathrm{I} = \arg\max_{Intention}[P(Intention|\,\mathbf{m})] \tag{2-8}$$

Je nach Einsatzgebiet des Ansatzes kann die Berechnung der Evaluierungswerte variieren und durch zusätzliche mathematische Operationen erweitert werden, die grundlegende Idee bleibt jedoch erhalten.

3

Intentionsbasierte Interpretation spontansprachlicher Äußerungen: Sprachverstehen

Dieses Kapitel beschreibt den Einsatz des intentionsbasierten Ansatzes für die Interpretation spontan- und natürlichsprachlicher Äußerungen. Die Intentionsmodelle ermöglichen hierbei eine syntaktisch-semantische und inhaltliche Evaluierung von beliebig vielen Wortkettenhypothesen, die durch einen Spracherkenner generiert werden. Bei dem hier vorgestellten Verfahren handelt es sich somit um ein sprachverstehendes System.

3.1 Grundidee

Die zentrale Komponente der Mensch-Maschine-Schnittstelle einer sprachgesteuerten Applikation ist ein Spracherkenner, der das akustische Signal einer Benutzeräußerung in eine Wortkette umsetzt. Auf Basis einer akustischen Signalanalyse vollzieht der Spracherkenner eine Rekonstruktion der Benutzeraktion, die schließlich zur Ermittlung der Benutzerintention interpretiert wird. Wie in den Kapiteln 1 und 2 erläutert, wird ist dies im Falle einer fehlerhaften Rekonstruktion mit erheblichen Risiken für den weiteren Dialog verbunden. Vor allem erweisen sich aktuelle Systeme zur Spracherkennung als immer noch zu wenig robust, um eine effiziente Mensch-Maschine-Interaktion auf Basis dieser Strategie zu gewährleisten.

Folgende Überlegung zeigt die Problematik des klassischen Ansatzes: Geht man davon aus, dass der Spracherkenner mit einer bestimmten Erkennungsrate R_W in der Lage ist, ein einzelnes Wort korrekt zu klassifizieren, so ist die Wahrscheinlichkeit, dass eine aus n_W Worten bestehende Wortkette fehlerfrei rekonstruiert wird, mit dem Wert $R_W{}^{n_W}$ zu quantifizieren. Dies stellt eine Vereinfachung unter der Annahme statistischer Unabhängigkeit zwischen der Erkennung aufeinander folgender Worte dar, die in der Realität nicht gegeben ist und sogar durch Sprachmodelle vermieden wird. Zur Erläuterung der Problematik bei der Rekonstruktion von Äußerungen ist diese vereinfachende Annahme jedoch legitim. Abbildung 3.1 zeigt die Wahrscheinlichkeit der vollständigen Rekonstruktion einer aus zwölf Worten bestehenden Äußerung in Abhängigkeit von verschiedenen

Einzelworterkennungsraten. Bereits bei einer Erkennungsrate von 90 Prozent lässt sich die gesamte Wortkette nur noch mit einer Wahrscheinlichkeit von 28,24 Prozent rekonstruieren. Eine Einzelworterkennungsrate von 75 Prozent schließt bereits eine vollständige Rekonstruktion der Äußerung nahezu aus. Unter idealen Bedingungen sind zwar Erkennungsraten von 95 Prozent durchaus zu erwarten; unter realen Bedingungen kann die Erkennungsrate jedoch deutlich sinken, da mit unsauberer Artikulation des Benutzers, nicht ideal ausgesteuerten Mikrofonen, Hintergrundgeräuschen und weiteren störenden Einflüssen zu rechnen ist. Dabei kann bereits ein einziges falsch klassifiziertes Wort den Sinn einer Aussage verändern.

Die Verwendung von Sprachmodellen trägt zwar zu einer höheren Klassifikationsleistung bei, kann sich aber bei *Out-of-Vocabulary*-Phänomenen auch sehr negativ auf die Rekonstruktion einer Äußerung auswirken.

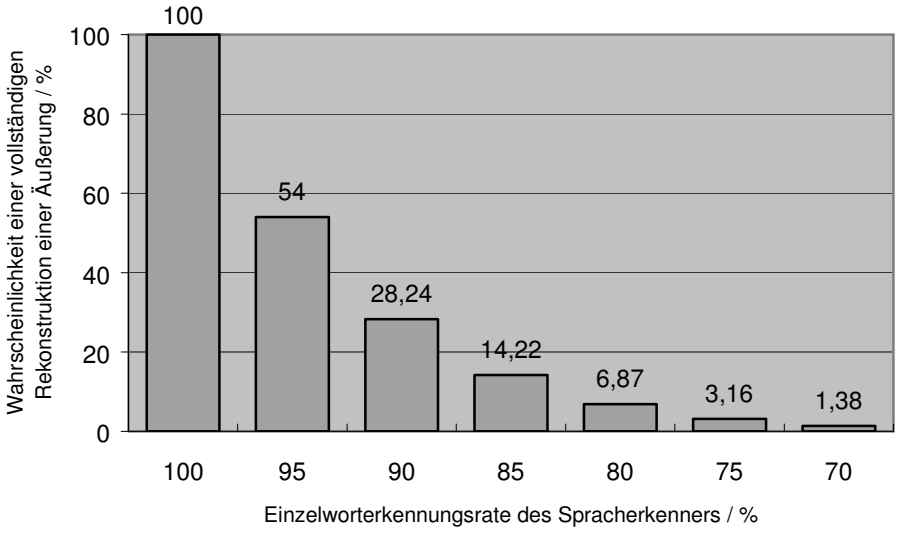


Abbildung 3.1: Wahrscheinlichkeit für eine vollständige Rekonstruktion einer aus zwölf Worten bestehenden Benutzeräußerung in Abhängigkeit von der Einzelworterkennungsrate des Spracherkenners

In diesem Kapitel wird das sprachverstehende System *Insense* (*In*tentionsbasierte *s*yntaktisch-semantische *E*valuierung *n*atürlich*s*prachlicher *E*ingaben) [Hof00][Hof01a] vorgestellt, das als Umsetzung des intentionsbasierten Ansatzes für die Interpretation spontansprachlicher Äußerungen die rein akustische Analyse des Spracherkenners durch eine syntaktisch-semantische und vor allem intentionsbezogene Evaluierung von Wortketten ergänzt.

Die Grundidee von *Insense* besteht darin, dass der Spracherkenner nicht die Entscheidung für eine bestimmte Wortkette treffen muss, da der intentionsbasierte Ansatz keine Rekonstruktion der Benutzeraktion voraussetzt. Die n besten Wortkettenhypothesen des Spracherkenners werden bezüglich ihrer Aussage evaluiert, indem sie mit dem Wissen um potenzielle Inhalte einer Äußerung in Bezug gesetzt werden. Wie in Kapitel 2 erläutert handelt es sich bei diesen Inhalten um die Benutzerintention selbst. Die Intentionsmodelle erlauben nicht nur ein direktes Abbilden von Wortkettenhypothesen auf Intentionshypothesen, sondern auch die Berücksichtigung von Syntax und Semantik bei beliebigen Satzstrukturen. Gängige Sprachmodelle, werden dadurch überflüssig. Dies unterstreicht die Fähigkeit von *Insense*, Spontansprache zu *verstehen*. Zudem kann der Einfluss von Out-

of-Vocabulary-Phänomenen auf das Klassifikationsergebnis durch konsequente Merkmalsselektion minimiert werden. Insense folgt er in Kapitel 2 diskutierten ersten Ausprägung des intentionsbasierten Ansatzes.

3.2 Stand der Technik

Aufgabe von sprachverstehenden Systemen ist die Analyse von gesprochenen Äußerungen mit dem Ziel, deren Inhalte zu ermitteln und diese in einer für einen Rechner interpretierbaren Form darzustellen. Diese Systeme bestehen in der Regel aus einer spracherkennenden Komponente, die durch Analyse des Sprachsignals versuchen, die ursprüngliche Äußerung zu rekonstruieren. Die resultierende Wortkette wird schließlich von einer zweiten Komponente auf deren Aussage hin untersucht. Hierfür wurde eine Reihe Algorithmen zur semantischen Analyse von Wortketten erforscht.

Die semantische Analyse wurde zunächst vor allem anhand von regel- oder grammatikbasierten Verfahren [Aus98] [Sen92] realisiert. Für grammatikalisch korrekte Wortketten haben sich diese Ansätze bewährt, da die Grammatik einer Sprache sich gut durch einen festen Regelsatz beschreiben lässt. Allerdings erwiesen sich diese Algorithmen im Fall von Spontansprache als nicht leistungsfähig genug, da Spontansprache häufig nicht grammatikalisch korrekt formuliert wird und jeder Mensch in der Art zu formulieren Eigenheiten aufzeigt. Somit müsste der Regelsatz auch eine Vielzahl unvorhersehbarer Fälle berücksichtigen; dies ist aber für die konkrete Realisierung eines sprachverstehenden Systems kaum möglich. Zudem wird unter Umständen unsauber artikuliert, was zu einer entsprechend verfremdeten Wortkette führt. Starre Regelwerke sind somit für Spontansprache kaum geeignet, um das komplexe Problem der semantischen Analyse zu lösen.

Mit dem Wissen um die begrenzten Möglichkeiten regelbasierter Ansätze für die semantische Analyse von Wortketten wird gegenwärtig der Fokus auf die Erforschung statistischer Verfahren zur Modellierung syntaktisch-sematischer Beziehungen zwischen Wörtern und Phrasen gelegt [Lev95]. In Analogie zur Spracherkennung nutzen einige dieser Ansätze Hidden-Markov-Modelle, um komplexe Regelwerke durch eine Reihe quantitativer, statistischer Zusammenhänge zu ersetzen [Mil94] [Sta97] [Mue97]. Sehr leistungsfähige Lernalgorithmen erlauben automatisches Lernen neuer syntaktisch-semantischer Beziehungen, neuer Domänen oder Sprachen. Dem steht allerdings der, verglichen mit regelbasierten Ansätzen, erheblich höhere Rechenaufwand gegenüber. Eine Kombination von regelbasierten und statistischen Herangehensweisen wurde u.a. von Wang [Wan02] untersucht.

Das bereits in Kapitel 1.3 kurz angesprochene System NASGRA von Stahl [Sta97] und Müller [Mue97] nutzt eine auf Hidden-Markov-Modellen basierende semantische Analyse zur Steuerung des Spracherkennungsprozesses. Ziel ist die robustere Berechnung der wahrscheinlichsten Wortkette, die schließlich regelbasiert auf deren Aussage hin untersucht wird. In die Analyse der Merkmale des Sprachsignals gehen somit zwar syntaktisch-semantische Aspekte von Phrasen mit ein, die Intention bleibt hierfür allerdings unberücksichtigt.

Macherey [Mac01] verwendet einen ursprünglich für die statistische, maschinelle Übersetzung von Äußerungen entwickelten Ansatz auf das Verstehen sprachlicher Äußerungen an. Anstatt eine unter

Umständen lückenhaft rekonstruierten Wortkette phrasenweise in eine grammatikalische korrekte Wortkette einer fremdem Sprache zu übersetzen, wird dieser Algorithmus benutzt, um die ursprüngliche Wortkette besser zu rekonstruieren.

Matsubara [Mat02] entwickelte einen intentionsbasierten Ansatz, der die von einem Spracherkenner erzeugten Wortketten mit Referenzäußerungen, die zur Artikulation bestimmter Intention dienen, vergleicht. Die Ähnlichkeit einer Wortkette mit einem Referenzmuster wird schließlich durch eine quantitatives Maß bewertet. Die Erkennungsrate dieses Systems für die Automobildomäne wird mit 68,9 Prozent angegeben.

Die erwähnten sprachverstehenden Systeme zielen alle darauf ab, die Originaläußerung zu rekonstruieren und auf Basis der dabei entstehenden Wortkette die Intention regelbasiert zu bestimmen. Indem *Insense* auf die Ermittlung der wahrscheinlichsten Wortkette verzichtet, unterscheidet sich das Verfahren grundsätzlich von in der Literatur bekannten Ansätzen.

3.3 Systemarchitektur

Abbildung 3.2 zeigt die grundlegende Struktur von *Insense*. Da es sich hier um eine konkrete Umsetzung des in Kapitel 2 erläuterten intentionsbasierten Ansatzes zur Interpretation von Benutzeraktionen handelt, wurde Abbildung 3.2 bewusst analog zu Abbildung 2.4 gehalten.

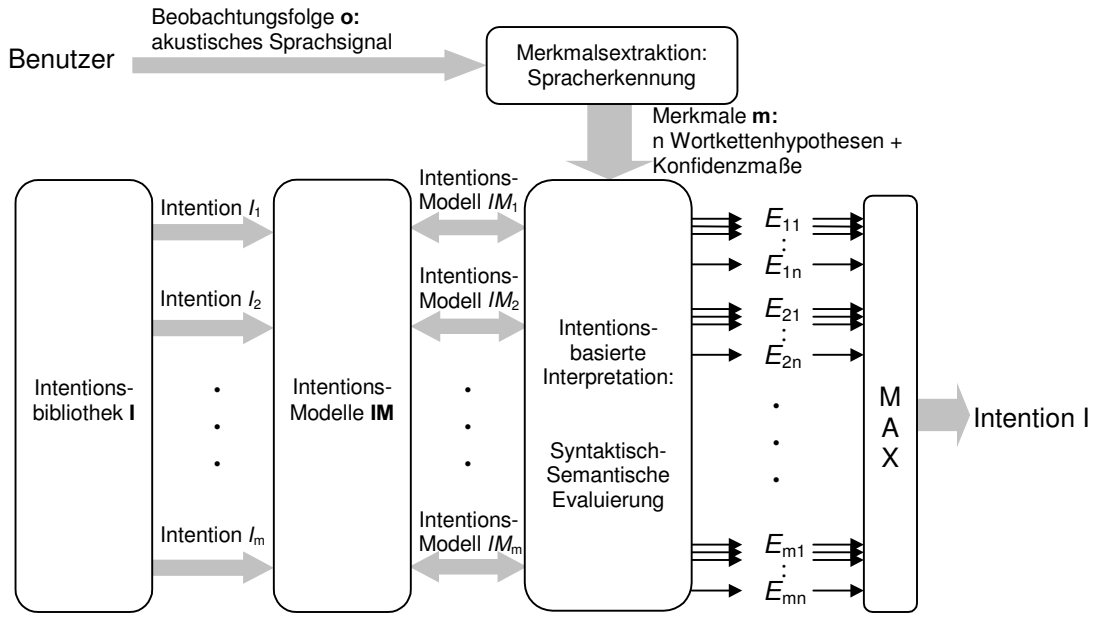


Abbildung 3.2: Ansatz zur intentionsbasierten Interpretation spontansprachlicher Äußerungen

Eingangssignal für das gesamte sprachverstehende System ist das akustische Signal einer Äußerung des Systembenutzers. Dieses Signal entspricht der Beobachtungsfolge **o**, die durch Merkmalsextraktion für die weiteren Komponenten interpretierbar gemacht wird. Die Merkmalsextraktion entspricht einer Analyse des akustischen Signals, d.h., aus dem akustischen Eingangssignal werden Wortkettenhypothesen generiert, die schließlich auf die Intentionsmodelle abgebildet werden. Mittels intentionsbasierter, syntaktisch-semantischer Interpretation der Wortkettenhypothesen werden

alle potenziellen Intentionen quantitativ evaluiert. Die Intention mit dem maximalen Evaluierungsmaß ist schließlich das Ergebnis des sprachverstehenden Prozesses.

In den folgenden Kapiteln werden nun alle Komponenten von *Insense* im Detail erläutert. Der Schwerpunkt wird dabei auf die Generierung der Intentionsmodelle sowie auf die syntaktisch-semantische Interpretation gelegt.

3.4 Intentionsbibliothek

Die Intentionsbibliothek **I** enthält alle Intentionen, die den Benutzer veranlassen könnten, sich mit der entsprechenden Applikation auseinander zu setzen. In der hier beschriebenen Implementierung handelt es sich um sprachgesteuerte, informationstechnische Einrichtungen eines Automobils. Die Elemente des Intentionsraums korrespondieren direkt mit den per Sprache manipulierbaren Systemfunktionen.

Abbildung 3.3 zeigt eine Darstellung des Intentionsraums für die verwendete Beispiel-Applikation, ein spontan- und natürlichsprachlich steuerbares Interface zur Kontrolle der wichtigsten informationstechnischen Geräte eines Automobils.

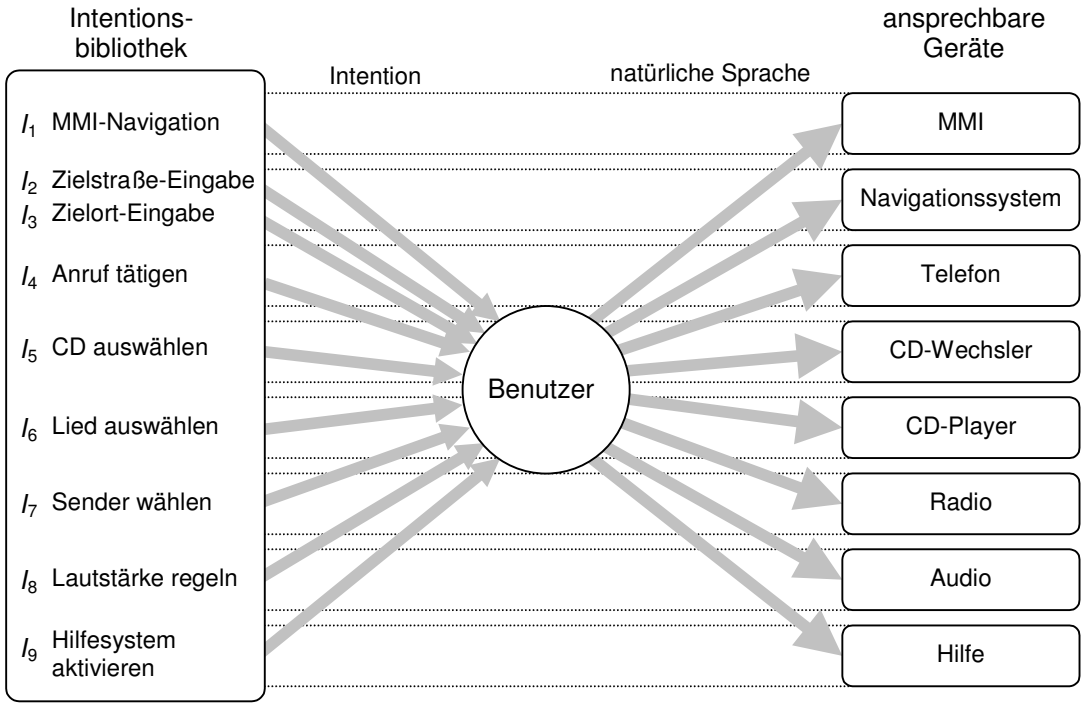


Abbildung 3.3: Intentionsbibliothek des sprachverstehenden Systems *Insense*

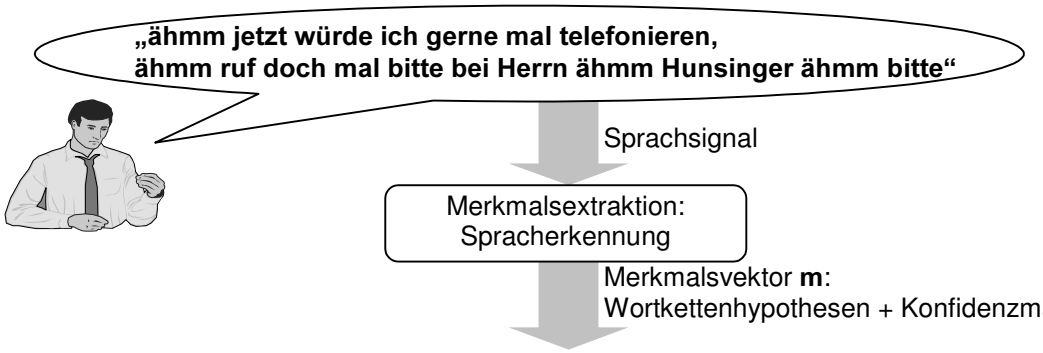
Die dargestellte Intentionsbibliothek umfasst neun Intentionen, wobei diese Anzahl die große Bandbreite unterschiedlicher Parameterkonstellationen pro Intention nicht berücksichtigt. Darüber hinaus sind Kombinationen verschiedener Intentionen möglich, die aber zugunsten der Erkennungsrate und eines für den Benutzer möglichst einfachen mentalen Modells bestimmten Richtlinien folgen. Der Benutzer kann seine Intention mit inhaltlich verwandten Unterintentionen näher spezifizie-

ren. So ist es zum Beispiel mittels einer einzigen Äußerung möglich, ein bestimmtes Lied anzusprechen und zur näheren Spezifikation die Nummer der CD und die Lautstärke anzugeben. Durch Kombination verschiedener Intention entstehen neue, übergeordnete Intentionshypothesen, die in Abbildung 3.3 nicht explizit dargestellt sind. Die Intentionsbibliothek kann auf diese Weise ca. 30 Intentionen berücksichtigen.

Die Darstellung der Intentionsbibliothek von *Insense* in Abbildung 3.3 macht deutlich, wie direkt die einzelnen Intentionshypothesen mit den per Sprache steuerbaren Geräten korrespondieren. Zu betonen ist, dass der Einsatz von *Insense* für beliebige Domänen und somit für beliebige Intentionsbibliotheken möglich ist. Die maximale Größe der Intentionsbibliothek ist dabei stark durch die zu steuernde Applikation geprägt. Bis zu 80 Intentionen sind bei zufrieden stellender Erkennungsrate realistisch.

3.5 Merkmalsextraktion

Aufgabe der Merkmalsextraktion von *Insense* ist die Umwandlung des akustischen Sprachsignals in eine Reihe von Wortkettenhypothesen. Hierfür wird ein handelsüblicher Spracherkenner, der *ASR 1600* von *Lernout&Hauspie* [L&H98], verwendet, der weder mit Sprachmodell noch mit einer zugrunde liegenden Grammatik betrieben wird, da syntaktische Zusammenhänge durch die Intentionsmodelle berücksichtigt werden.



1.	hören	ins	würde	stell	mal	telefonieren	rufe	Nummer	bitte	Wahl	mal	5	Wahl	Hunsinger	bitte	
	73.0	39.2	57.7	63.2	81.7	70.1	53.6	30.7	70.2	72.3	58.7	69.0	50.4	71.7	23.4	
2.	Köln	ins	würde	stell	mal	telefonieren	rufe	Nummer	bitte	Wahl	mal	5	Wahl	Hunsinger	bitte	
	71.2	39.2	57.7	63.2	81.7	70.1	53.6	30.7	70.2	72.3	58.7	69.0	50.4	71.7	23.4	
3.	hören	ins	würde	stell	mal	telefonieren	rufe	rück	bitte	Wahl	mal	5	Wahl	Hunsinger	mit	Ralf
	73.0	39.2	57.7	63.2	81.7	70.1	53.6	30.7	70.2	72.3	58.7	69.0	50.4	71.7	15.2	58.5
								⋮								
20.	hören	will	zur	ich	gerne	mal	telefonieren	rufe	Nummer	bitte	Wahl	mal	5	Wahl	Hunsinger	mit	8
	73.0	3.2	73.2	43.5	54.1	81.8	70.1	53.6	30.7	70.2	72.3	58.7	69.0	50.4	71.7	14.2	55.2

Abbildung 3.4: Wortkettenhypothesen und Konfidenzmaße bilden den Merkmalsvektor

Voraussetzung für das Sprachverstehen ist die Fähigkeit der Spracherkenners, die n besten Wortkettenhypothesen auszugeben und jedes einzelne Wort mittels eines Konfidenzmaßes zu gewichten. Dies ermöglicht der intentionsbasierten Interpretation die Bevorzugung von Wörtern, die zwar durch die Signalanalyse des Spracherkenners als unwahrscheinlich evaluiert wurden, allerdings

unter Einbezug syntaktisch-semantischer Kriterien und im Hinblick auf die Gesamtaussage der Äußerung, stärker gewichtet werden als die akustisch besser geeigneten Alternativen.

Abbildung 3.4 zeigt 20 Wortketten-Hypothesen als Ergebnis der Analyse des akustischen Signals durch die Merkmalsextraktion. Die Originaläußerung enthält mehrere typisch spontansprachliche Elemente wie „ähmm“ sowie einige im Spracherkenner-Vokabular nicht vorkommende Worte. Das Ergebnis sind 20 Merkmalsvektoren **m**, die sich jeweils aus den Worten einer Hypothese und deren Konfidenzmaße zusammensetzen.

Der Vergleich zeigt die starke Ähnlichkeit der aus Sicht des Spracherkenners 20 besten Wortkettenhypothesen. Kleine Unterschiede in der Wortfolge oder in den Konfidenzmaßen können bereits zu verschiedenen Aussagen führen. Zudem macht Abbildung 3.4 die Notwendigkeit einer intelligenten Merkmalsselektion deutlich, da eine semantische Analyse der Wortkettenhypothesen auf Basis jedes einzelnen erkannten Wortes in diesem Fall fehlschlagen würde.

3.6 Intentionsmodelle

Die Aufgabe eines Intentionsmodells besteht in der Repräsentation einer Intention derart, dass diese mit den Merkmalen **m** direkt in Bezug gesetzt werden kann. Da es sich bei den Merkmalen um Wortketten bzw. auf der untersten Ebene um Worte handelt, repräsentiert ein Intentionsmodell alle potenziellen Äußerungen des Benutzers, mit denen er das System über seine Absicht informieren möchte. In diesem Abschnitt wird zunächst die Struktur der Intentionsmodelle vorgestellt und schließlich die Realisierung der Intentionsmodelle auf Basis Bayes'scher Netze erläutert.

3.6.1 Struktur der Intentionsmodelle

Eine Intention kann bei sprachgesteuerten Applikationen durch eine Reihe verschiedener Äußerungen mitgeteilt werden; auf die Liste von Äußerungen wird nachfolgend als *Äußerungsbibliothek* einer Intention verwiesen. Da eine umfangreiche Äußerungsbibliothek das Risiko einer zu großen Komplexität eines Intentionsmodells birgt, wird die Äußerungsbibliothek eines Intentionsmodells durch Kombinieren einer Reihe von Phrasen verschiedener Phrasentypen erstellt.

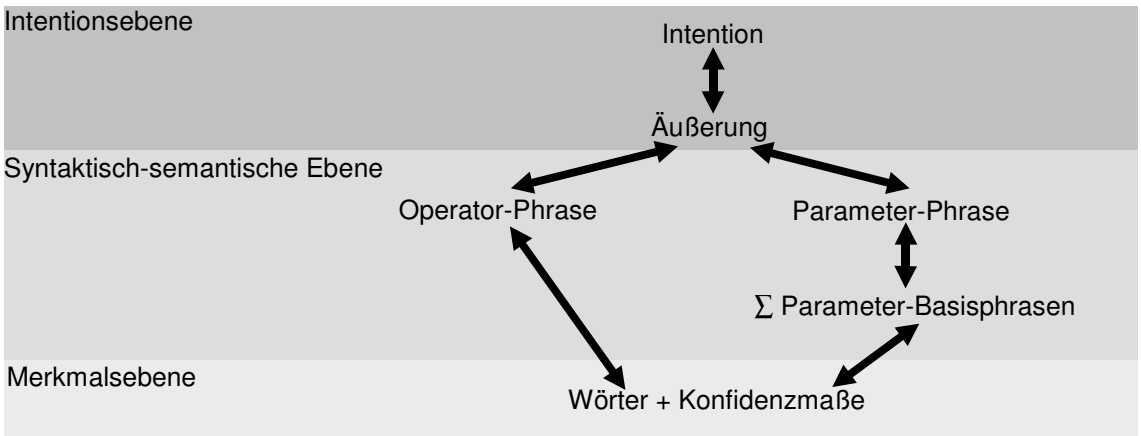


Abbildung 3.5: Struktur eines Intentionsmodells von *Insense*

Abbildung 3.5 stellt einen Überblick über die verwendeten Phrasentypen und über die Struktur eines Intentionsmodells dar.

Ein Intentionsmodell von *Insense* ist in drei Ebenen gegliedert: die Intentionsebene, die syntaktisch-semantische Ebene und die Merkmalsebene. Nachfolgend werden diese drei Ebenen und ihre Verknüpfungen zueinander beschrieben:

- **Intentionsebene**
  Anhand der obersten Ebene, der Intentionsebene, werden Aussagen über die Wahrscheinlichkeit der entsprechenden Intention getroffen, indem jeder Intention als Ergebnis der intentionsbasierten Analyse des Merkmalsraums ein quantitatives Evaluierungsmaß E zugewiesen wird. Die Klassifikation der Intention erfolgt dabei nicht diskriminativ, d.h., jede Intentionshypothese wird unabhängig von den übrigen Intentionshypothesen bewertet. Im Rahmen dieser Evaluierung werden die durch den Spracherkenner erzeugten Wortkettenhypothesen mit den Äußerungsbibliotheken aller Intentionen verglichen. In welcher Weise dies geschieht wird auf der syntaktisch-semantischen Ebene definiert.

- **Syntaktisch-semantische Ebene**
  Die syntaktisch-semantische Ebene teilt alle zur Intention gehörigen, potenziellen Äußerungen in zwei Phrasen auf: den *Operator* und den *Parameter*. Der Operator dient dazu, dem System mitzuteilen, welche Systemfunktion angesprochen werden sollen; der Parameter gibt an, inwiefern diese Systemfunktion manipuliert werden soll. Bei Operator und Parameter handelt es sich um syntaktisch und semantisch zusammengehörige Phrasen, die zu einer syntaktisch-semantisch korrekten Äußerung zusammengefügt werden.

  Operator und Parameter werden nun unterteilt in die so genannten *Basisphrasen*. Basisphrasen stellen die unterste Phrasenebene dar, d.h., sie werden nicht mehr in weitere Phrasen zerlegt, sondern nur in syntaktisch und semantisch verwandte Wörter (z.B. „ich möchte fahren“ oder „in die Barerstraße“). Bei dem Operator handelt es bereits um eine Basisphrase, weil das direkte Ansprechen einer Systemfunktion in der Regel mit einer einfachen Phrase formuliert werden kann. Die direkte Angabe eines Parameters kann allerdings erheblich komplexer ausfallen, da ein Parameter häufig dem Übermitteln einer Reihe verschiedener Attribute einer Systemfunktion entspricht. In solchen Fällen wird der Parameter in so genannte *Parameter-Phrasen* zerlegt. Ein Parameter kann also einer Reihe von Parameter-Basisphrasen entsprechen, während es sich bei dem Operator genau um eine Operator-Basisphrase handelt. Die Zusammensetzung einzelner Basisphrasen zu einer Äußerungsbibliothek einer Intention wird im nächsten Abschnitt im Detail geschildert.

- **Merkmalsebene**
  Die Merkmalsebene bildet die Schnittstelle der Intentionsmodelle zum Merkmalsvektor. Da es sich bei den Merkmalen um Wortbeobachtungen und deren Konfidenzmaße handelt, sehen die Intentionsmodelle pro potenzielles Wort einen Merkmalsknoten vor.

3.6.2 Realisierung der Intentionsmodelle durch Bayes'sche Netze

Die Strukturierung eines Intentionsmodells dient zur kompakten Repräsentation der Äußerungsbibliothek einer Intention, um von der Merkmalsebene direkt auf die Intention schließen zu können. Zur mathematischen Beschreibung werden, wie bereits in Kapitel 2 diskutiert, Bayes'sche Netze verwendet, deren Topologien und Wahrscheinlichkeiten im Folgenden erläutert werden.

Die Beschreibung der verwendeten Netzwerke folgt der im vorhergehenden Abschnitt vorgestellten Struktur. Zunächst wird die Intentionsebene und ihre Verknüpfung mit der Äußerungsebene geschildert. Abbildung 3.6 zeigt die Topologie des zugehörigen Bayes'schen Netzes.

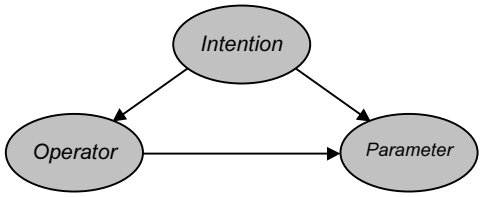


Abbildung 3.6: Topologie des Intentionsnetzes

Da dieses Netzwerk direkt Aussagen über die Intention gestatten soll, wird darauf als *Intentionsnetz* Bezug genommen. Die Intention wird im Rahmen des Intentionsnetzes mittels einer Boole'schen Zustandsvariable repräsentiert, die als Wurzelknoten keine direkte Abhängigkeiten von anderen Knoten besitzt. Die a-priori-Wahrscheinlichkeit $P(Intention)$ dieser diskreten Zustandsvariable wird neutral gewählt, sodass beide Zustände als gleich wahrscheinlich zu interpretieren sind. Operator und Parameter werden ebenfalls durch jeweils eine Boole'sche Zustandsvariable modelliert.

Der Tatsache, dass eine mit einer bestimmten Intention korrespondierende Äußerung in der Regel aus einem Operator und einem Parameter besteht, wird durch die Struktur und durch die Wahl der bedingten Wahrscheinlichkeiten $P(Operator|Intention)$ und $P(Parameter|Operator, Intention)$ Rechnung getragen. Im Rahmen der verwendeten Domäne zeigte sich, dass bereits auf Grund einer Parameterbeobachtung in den meisten Fällen direkt auf die Intention geschlossen werden kann. Der Parameter wurde deshalb stärker gewichtet und die bedingten Wahrscheinlichkeiten derart gewählt, dass eine vollständige Beobachtung des Parameters einer vollständigen Beobachtung einer bestimmten Intention entspricht. Darüber hinaus wurde die Modellierung derart ergänzt, dass eine vollständige Beobachtung von Operator und syntaktisch-semantisch verwandtem Parameter einer vollständigen Beobachtung einer mit der entsprechenden Intention korrespondierenden Äußerung entspricht.

Gleichung (3-1) stellt die Zusammenhänge als logischen Ausdruck dar:

$$Intention = (Operator \wedge Parameter) \vee Parameter \tag{3-1}$$

Der in Klammern gesetzte logische Ausdruck wäre im Falle einer vollständigen Beobachtung des Parameters überflüssig. Beobachtungen im vorgestellten Verfahren sind jedoch nie vollständig, sie werden immer als unsichere Information behandelt, da die Einzelworte der zu analysierenden Wortkettenhypothesen durch den Spracherkenner mit Konfidenzmaßen gewichtet werden, die nie-

mals dem maximal möglichen Wert entsprechen. Die Klassifikation der Intention basiert also größtenteils auf dem Parameter; der Operator dient der Konsolidierung des Klassifikationsergebnisses.

Die Modellierung der Verknüpfung von Äußerungs- und Merkmalsebene erfolgt in ähnlicher Weise. In Analogie zur Intentionsebene wird die Beobachtung der Operator-Basisphrase durch eine diskrete Boole'sche Zustandsvariable repräsentiert. In dem zugehörigen Bayes'schen Netz aus Abbildung 3.7 handelt es sich dabei um den Wurzelknoten, von dem alle übrigen Knoten des Netzes statistisch abhängig sind. Jedes Wort, d.h. jedes Merkmal der Operator-Basisphrase wird durch eine Boole'sche Zustandsvariable dargestellt, der ein Vokabular zugeordnet ist, das genau diesem Wort entspricht oder syntaktisch-semantisch adäquate Synonyme enthält.

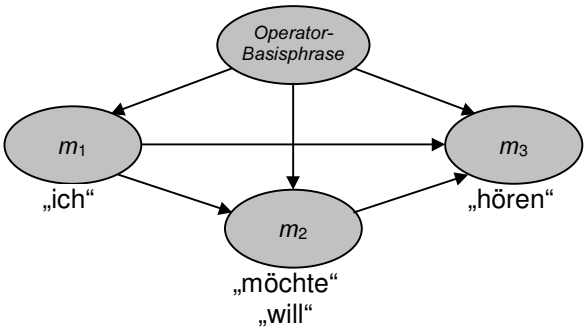


Abbildung 3.7: Topologie für eine Operator-Basisphrase bestehend aus drei Wörtern

Eine Operator-Basisphrase gilt dann als vollständig beobachtet, wenn für alle Merkmalsknoten je ein Wort vollständig beobachtet wurde. Dies wird wieder durch die Topologie des Netzes und durch die Wahl der bedingten Wahrscheinlichkeiten entsprechend einer logischen UND-Funktion gewährleistet. Die nachfolgende Gleichung stellt dies allgemein für eine Operator-Basisphrase mit n Merkmalen bzw. Worten dar:

$$Operator - Basisphrase = m_1 \wedge m_2 \wedge \ldots \wedge m_n \tag{3-2}$$

Bei der Topologie des Netzes handelt es sich wieder um eine geeignete Struktur zur Repräsentation einer logischen UND-Funktion. Jeder Knoten muss mit den übrigen Knoten durch eine Kante verbunden sein, ohne dass zyklische Strukturen innerhalb des Netzwerks entstehen.

Die Modellierung des Parameters weicht von der Modellierung eines Operators in verschiedenen Punkten ab. Die Tatsache, dass ein Parameter aus mehreren Parameter-Basisphrasen bestehen kann, wird durch eine Zwischenebene berücksichtigt. Folgende logische Gleichung stellt den Zusammenhang für einen aus m Parameter-Basisphrasen bestehenden Parameter mathematisch dar:

$$Parameter - Phrase = Parameter - Basisphrase1 \wedge \ldots \wedge Parameter - Basisphrasem \tag{3-3}$$

Auch in diesem Fall wird die unsichere Beobachtung der Parameter-Basisphrasen probabilistisch beschrieben; die logische Gleichung dient nur der besseren Orientierung. Die korrespondierende Topologie des Netzes für einen aus drei Parameter-Basisphrasen bestehenden Parameter wird in folgender Abbildung gezeigt:

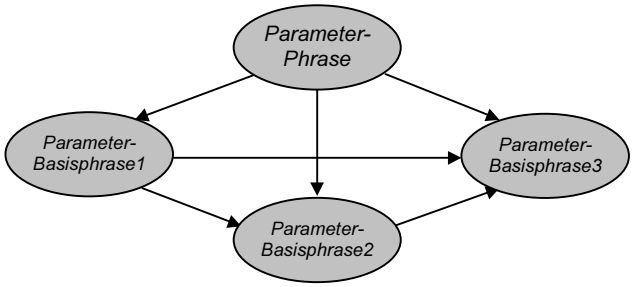


Abbildung 3.8: Topologie für eine Parameter-Phrase bestehend aus drei Parametern

Die Anzahl der maximal zulässigen Parameter-Basisphrasen eines Parameters wurde auf drei begrenzt, da die Anzahl der pro Äußerung maximal ansprechbaren Systemfunktionen maximal drei Funktionen umfassen darf. Dies ist die einzige Vorgabe an den Benutzer. Der Benutzer erhält auf diese Weise ein sehr einfaches mentales Modell von der Eingabeschnittstelle, da er sich darüber im Klaren ist, wie komplex typische Äußerungen maximal sein dürfen. Auf der technischen Seite bedeutet dies erheblich höhere Erkennungsraten, da diese Vorgabe den Benutzer davon abhält, alle seine Intentionen innerhalb einer einzigen Äußerung zu artikulieren. Somit halten sich zu komplexe Satzkonstruktionen und Widersprüche innerhalb einer Äußerung in Grenzen.

Die Modelle der Parameter-Basisphrasen unterscheiden sich in Struktur und Wahrscheinlichkeiten von den Modellen der Operator-Basisphrasen. Im Gegensatz zur Operator-Basisphrase werden die Worte einer Phrase nicht identisch gewichtet, sondern zunächst in *Schlüsselworte* und *optionale Worte* klassifiziert. Bei den Schlüsselworten handelt es sich um die Worte, die den Hauptteil der Information der Phrase tragen. Ein Mensch könnte anhand dieser Schlüsselworte auf die Aussage der Phrase schließen, während die optionalen Worte dazu dienen, die Schlüsselworte zu einer syntaktisch und semantisch korrekten Phrase zu vervollständigen.

Der Grund für diese unterschiedliche Gewichtung liegt in der Tatsache, dass gerade relativ kurze Wörter, wie zum Beispiel Artikel, durch den Spracherkenner unzureichend robust klassifiziert werden und somit im Falle einer Gleichberechtigung aller Worte einer Phrase negative Auswirkungen auf die Klassifikation der gesamten Parameter-Basisphrase haben. Die Schlüsselworte bestehen in den meisten Fällen aus einer größeren Anzahl von Buchstaben, wodurch eine robustere Analyse des akustischen Signals durch den Spracherkenner möglich ist.

Ein weiterer Vorteil für die Einteilung der Wörter in optionale Worte und Schlüsselworte liegt in der größeren Flexibilität bei der Modellierung einer Phrase. Auf diese Weise können mehrere Phrasen, die sich nur in einigen Wörtern unterscheiden, durch dasselbe Netzwerk repräsentiert werden. Die Phrasen „Lied Nummer vier“ und „Lied vier“ zum Beispiel lassen sich mit dem optionalen Wort „Nummer“ und den Schlüsselworten „Lied“ und „vier“ durch das Bayes’sche Netz aus Abbildung 3.9 modellieren. Da es sich bei den Schlüsselworten und bei den optionalen Worten um Merkmale handelt, werden deren Knoten mit m_S bzw. m_O bezeichnet.

Durch die mit der Einteilung aller Wörter in optionale Worte und Schlüsselworte verbundene Flexibilität lässt sich somit die Anzahl der benötigten Netze und somit die Komplexität des Gesamtsystems reduzieren.

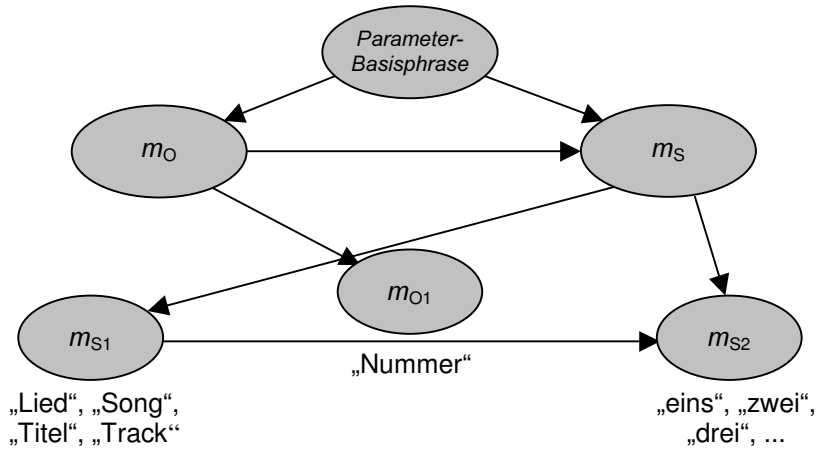


Abbildung 3.9: Topologie eines Bayes'schen Netzes zur Modellierung einer Parameter-Basisphrase

Beobachtungen von Schlüsselworten lassen sich durch eine Beobachtung syntaktisch-semantisch korrekter optionaler Worte noch untermauern. Ohne die optionalen Worte würde eine Äußerung zur Steuerung einer Systemfunktion einer Kommandosprache entsprechen. Dadurch ist *Insense* in der Lage, sowohl natürlichsprachliche Sätze als auch Kommandosprache zu interpretieren.

Die Knoten des in Abbildung 3.9 dargestellten Netzwerks visualisieren diskrete Boole'sche Zustandvariablen. Der Wurzelknoten modelliert die Beobachtung der Parameter-Basisphrase, bestehend aus der Beobachtung der Schlüsselworte und der optionalen Worte. Die bedingten Wahrscheinlichkeiten werden wieder entsprechend folgender logischer Gleichung bestimmt:

$$\textit{Parameter-Basisphrase} = (m_O \wedge m_S) \vee m_S \tag{3-4}$$

Die Beobachtung aller für eine Phrase relevanten Schlüsselworte wird wieder mittels einer UND-Modellierung der bedingten Wahrscheinlichkeiten erreicht:

$$m_S = m_{S1} \wedge m_{S2} \wedge \ldots \wedge m_{Sr} \tag{3-5}$$

Gleichung (3-5) gilt allgemein für eine Phrase mit r Schlüsselworten. Das Netzwerk aus Abbildung 3.9 enthält zwei Schlüsselworte, um die Intention eindeutig zu beschrieben. Diesen beiden Merkmalsknoten ist ein Vokabular mit den relevanten Synonymen bzw. Parameterwerten zuzuordnen.

Optionale Worte werden anhand folgender ODER-Modellierung berücksichtigt:

$$m_O = m_{O1} \vee m_{O2} \vee \ldots \vee m_{Op} \tag{3-6}$$

Jedes optionale Wort geht demnach unabhängig von der Beobachtung der anderen optionalen Wörter in die Klassifikation mit ein. Dies ermöglicht größere Flexibilität bei der Repräsentation einer möglichst großen Anzahl verschiedener Phrasen durch ein Modell. An einem später diskutierten Beispiel wird dies deutlich werden.

Zusätzlich zu den erwähnten Informationen zur Definition eines Parameter-Basisphrasen-Modells wird definiert, in welcher Reihenfolge die Schlüsselwörter und die optionalen Wörter in einer Äußerung vorkommen dürfen. Dies ermöglicht eine korrekte syntaktische Modellierung.

Abbildung 3.10 gibt einen Überblick über die Struktur eines Intentionsmodells:

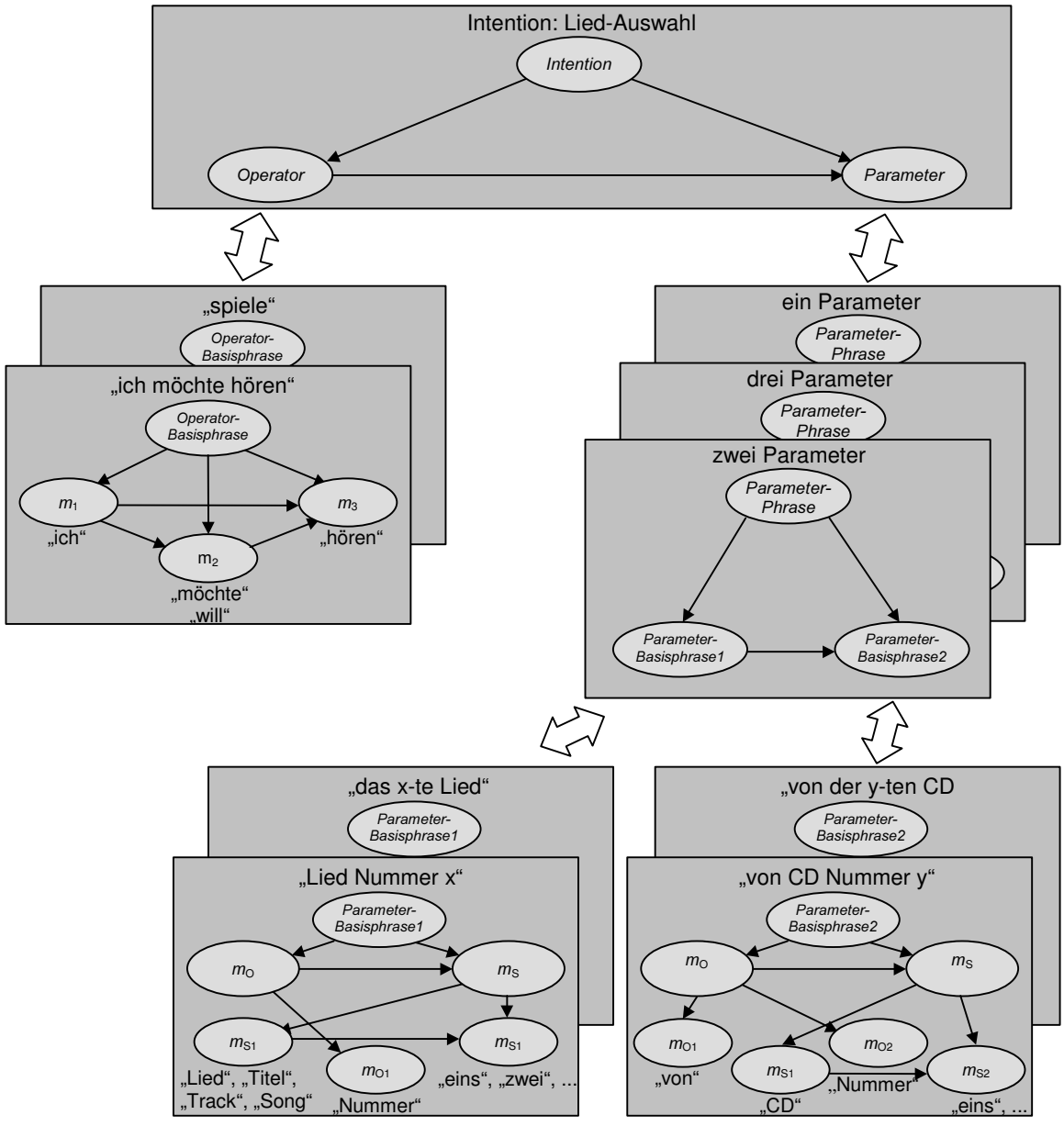


Abbildung 3.10: Beispiel für ein Intentionsmodell von *Insense*

Bei der Intention in Abbildung 3.10 handelt es sich um die Wahl eines bestimmten Liedes von CD, zusätzlich kann der Benutzer die CD-Nummer (CD-Wechsler) und die gewünschte Lautstärke angeben. Die oberste Ebene stellt die Beobachtung einer mit der entsprechenden Intention korrespondierenden Äußerung in Abhängigkeit von Parameter und Operator dar. Für jede potenzielle Operator-Basisphrase wird ein Bayes'sches Netz in der beschriebenen Weise definiert. Entsprechendes gilt für den Parameter mit der Ausnahme, dass die Intentionsebene mit den Basisphrasen über eine zusätzliche Ebene kommuniziert. Für jede Intentionshypothese der Intentionsbibliothek ist ein derartig strukturiertes Intentionsmodell zu definieren.

Welche Vielzahl von potenziellen sprachlichen Äußerungen zur Auswahl eines Liedes durch die in Abbildung 3.10 dargestellten Netzwerke modelliert wird, ist in Abbildung 3.11 dargestellt.

Abbildung 3.11 zeigt die Liste dieser Äußerungen, bestehend aus zwei Operator-Basisphrasen und bis zu drei Parametern, die jeweils durch zwei Parameter-Basisphrasen definiert werden.

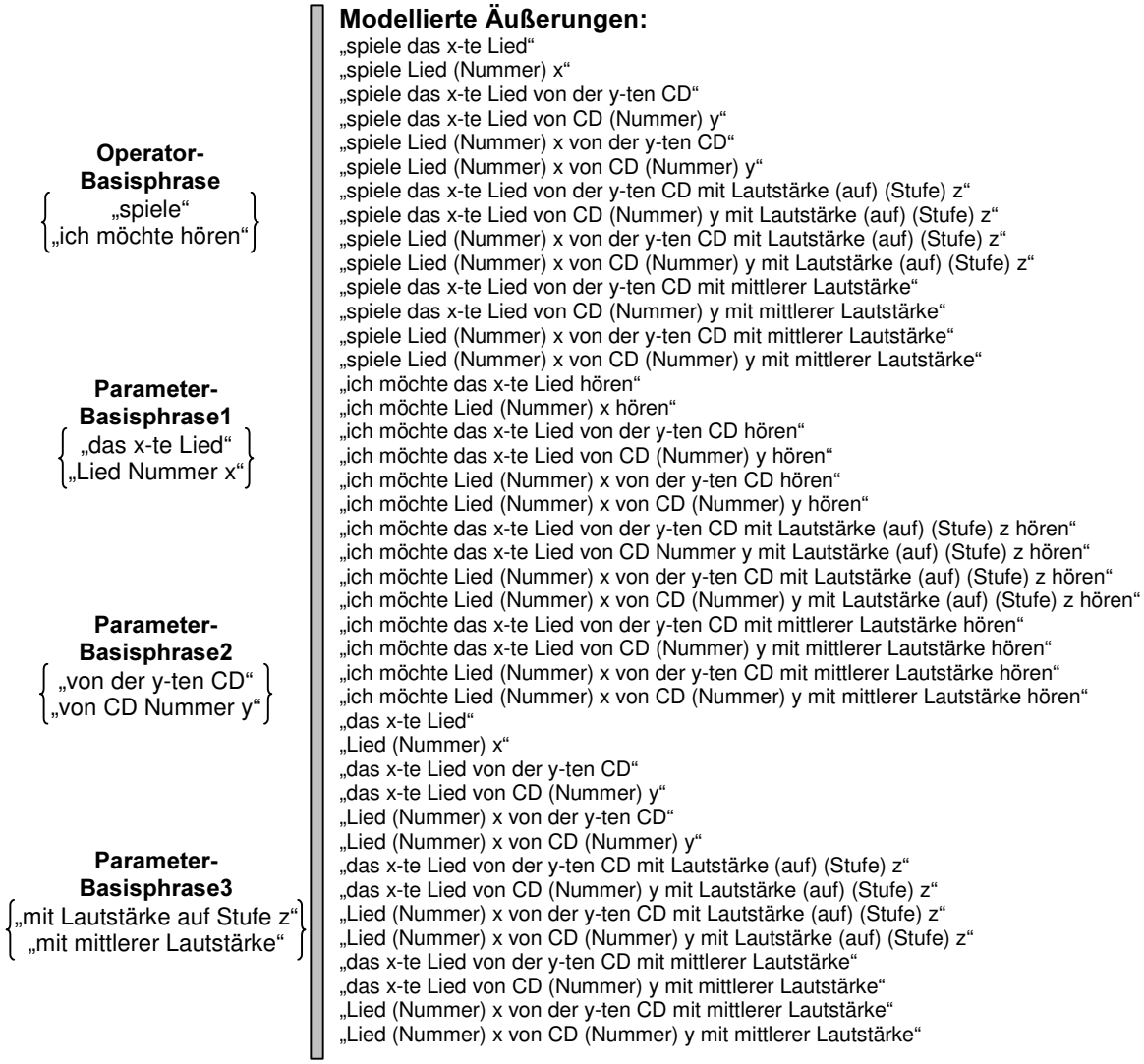


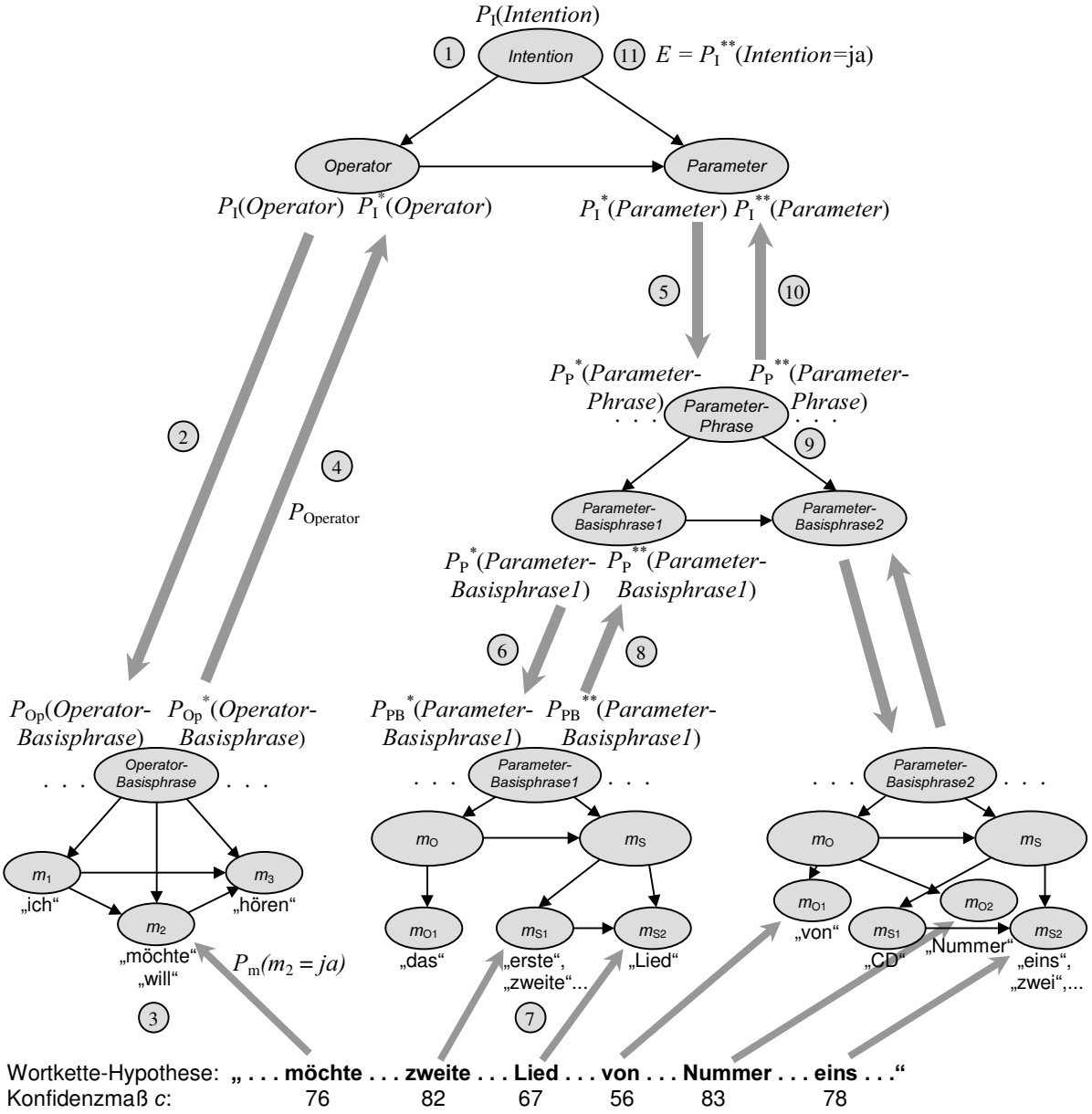
Abbildung 3.11: Liste der Äußerungen, die durch das in Abbildung 3.10 dargestellte Intentionsmodell abgedeckt werden

Wie man anhand von Abbildung 3.11 erkennen kann, ist das Intentionsmodell in der Lage, syntaktisch-semantisch korrekte Äußerungen bis hin zu kommandosprachlichen Satzfragmenten zu berücksichtigen. In Klammern dargestellt sind die optionalen Worte, die die Berücksichtigung einer noch größeren, nicht explizit aufgelisteten Anzahl von Äußerungen gestatten. Die modellierten Äußerungen ergeben sich aus dem Kombinieren der Operator- und Parameter-Basisphrasen. Prinzipiell gibt es keine Einschränkungen bezüglich der Satzstrukturen. Auch Nebensätze sind möglich.

Je mehr Informationen für die Klassifikation zur Verfügung stehen, desto mehr Informationen werden genutzt. Im Allgemeinen ist deshalb die Klassifikation der Intention aus einer natürlichsprachlichen Äußerung robuster als die Klassifikation auf Basis knapper Kommandosprache. Darauf wird in einem späteren Kapitel im Rahmen der Diskussion von Untersuchungsergebnissen näher eingegangen.

3.7 Intentionsbasierte Interpretation

Die intentionsbasierte Interpretation besteht aus einer syntaktisch-semantischen Evaluierung der Merkmale bzw. der Wortkettenhypothesen. Zur Erläuterung dieser Komponente ist in Abbildung 3.12 ein Intentionsmodell zur Analyse der am unteren Bildrand angegebenen Wortkettenhypothese dargestellt. Der Index der verwendeten Wahrscheinlichkeiten gibt an, auf welches Netzwerk Bezug genommen wird. $P_{\mathrm{I}}(\cdot)$ steht für das Intentionsnetz, $P_{\mathrm{Op}}(\cdot)$ für ein Operatornetz, $P_{\mathrm{P}}(\cdot)$ für ein Parameternetz und $P_{\mathrm{PB}}(\cdot)$ bezieht sich schließlich auf ein Parameter-Basisphrasen-Netzwerk.



Gesprochene Äußerung:
„ähmm ich möchte jetzt gerne mal das ähmm zweite Lied von der CD Nummer ähmm eins hören“

Abbildung 3.12: Darstellung der intentionsbasierten Interpretation einer spontansprachlichen Äußerung

Die für die intentionsbasierte Interpretation erforderlichen Überlegungen werden in den Schritten ① bis ⑪ vorgestellt:

① Zunächst wird dem Intentionsknoten eine neutrale Wahrscheinlichkeitsverteilung als a-priori-Wahrscheinlichkeit $P_{\mathrm{I}}(Intention)$ zugewiesen, sodass beide Zustände gleich wahrscheinlich sind:

$$(0{,}5;\ 0{,}5) \quad \mapsto \quad P_{\mathrm{I}}(Intention) \tag{3-7}$$

Die Wahrscheinlichkeit des ja-Zustands repräsentiert die quantitative Annahme, dass es sich bei der aktuellen Intention um die Benutzerintention handelt. Der minimale Wert für diesen Zustand entspricht 0,5, da Beobachtungen von Wörtern lediglich die Annahme einer Intention stützen können. Der Zustand *nein* entspricht nicht der Annahme, dass es sich um die falsche Intention handelt. Dieser Zustand bildet lediglich im Rahmen der Wahrscheinlichkeitstheorie den Gegenpol zum ja-Zustand.

② Anhand des Intentionsnetzes wird die Randwahrscheinlichkeit für den Operatorknoten $P_{\mathrm{I}}(Operator)$ berechnet und den Wurzelknoten aller Operator-Basisphrase-Netze als a-priori-Wahrscheinlichkeit zugewiesen. Für jedes Operatornetz gilt somit:

$$P_{\mathrm{I}}(Operator) \quad \mapsto \quad P_{\mathrm{Op}}(Operator - Basisphrase) \tag{3-8}$$

Dieser Schritt ist nötig, um die Operator-Basisphrase-Netze mit dem Intentionsnetz zu synchronisieren.

③ Nun werden die einzelnen Wörter der zu evaluierenden Wortkettenhypothese mit den Operator-Basisphrasen verglichen, d.h., die Wortkettenhypothese wird nach Wörtern aus den Vokabularen, die den einzelnen Merkmalsknoten zugewiesen wurden, untersucht. Dieser Vorgang wird als *Parsing* bezeichnet. Der Wortkettenausschnitt in Abbildung 3.12 zeigt lediglich die Worte, die für die betrachtete Intention relevant sind; die übrigen Worte wurden zur besseren Übersicht nicht dargestellt.

Im Falle einer Übereinstimmung von Merkmal und einem Vokabulareintrag eines Merkmalknotens muss die Beobachtung des entsprechenden Wortes auf den korrespondierenden Knoten des Operator-Basisphrasen-Netzes abgebildet werden. In Abbildung 3.12 trifft dies zuerst auf das Wort „möchte“ zu, das vom Spracherkenner mit einer quantitativen Konfidenz von 76 klassifiziert wurde. Für das Konfidenzmaß *c* steht dem Spracherkenner ein Wertebereich von 0 bis 100 zur Verfügung, wobei alle Werte zwischen den beiden Extremwerten auf Unsicherheit bei der Klassifikation hinweisen.

Unsicherheiten bei der Beobachtung von Wörtern müssen entsprechend als neue, unsichere Information in das Basisphrasennetz eingespeist werden. Da Bayes'sche Netze im Allgemeinen zwar Unsicherheit modellieren können, der Theorie entsprechend jedoch ausschließlich sichere Informationen in das Netz gebracht werden können, wurden Routinen entwickelt, um unsichere Informationen aus Quellen außerhalb des Netzes verarbeiten zu können.

Wie jedes probabilistisches Netz entspricht ein Bayes'sches Netz einer Repräsentation der Verbundwahrscheinlichkeit über alle Zustandsvariablen. Im Falle einer sicheren Information bezüglich einer bestimmten Zustandsvariablen werden in der Regel alle Einträge der Verbundwahrscheinlichkeit, die auf Grund der Kenntnis des Zustands dieser Variable unmöglich sind, auf Null gesetzt. Die resultierende Verbundwahrscheinlichkeit repräsentiert schließlich diese sichere Information bezüglich einer Zustandsvariable.

Entsprechend dem Bayes'schen Theorem lässt sich die Verbundwahrscheinlichkeit eines Netzwerks, bestehend aus dem Basisphrasenknoten als Wurzelknoten und einer Reihe von Merkmalsknoten, anhand folgender Gleichung berechnen:

$$P(Basisphrase, m_1, m_2, ...) = P(Basisphrase, m_1, m_2, ... \mid m_x) \cdot P(m_x) \tag{3-9}$$

Die unsichere Beobachtung eines Wortes, das durch den Knoten m_x repräsentiert wird, wird nun mittels einer Veränderung seiner Randwahrscheinlichkeit $P(m_x)$ modelliert:

$$P_{\mathrm{m}}(m_x) \quad \mapsto \quad P(m_x) \tag{3-10}$$

Interpretiert man die probabilistische Inferenz innerhalb eines Bayes'schen Netzes als die Berechnung der quantitativen Annahme über das Stattfinden eines bestimmten Ereignisses, so ist Gleichung (3-10) als Änderung dieser Annahme zu interpretieren. Entscheidend ist dabei, dass die Informationsquelle, die zu dieser Änderung führt, in diesem Fall nicht im Netzwerk modelliert ist. Es handelt sich also um eine Informationsquelle „von außen", die zu der Aussage $P_{\mathrm{m}}(m_x)$ führt; in diesem Fall wird der Spracherkenner als externe Informationsquelle interpretiert.

Die Änderung der Randwahrscheinlichkeit der Zustandsvariable m_x bringt eine neue Verbundwahrscheinlichkeit mit sich:

$$P^*(Basisphrase, m_1, m_2, ...) = P(Basisphrase, m_1, m_2, ... \mid m_x) \cdot P_{\mathrm{m}}(m_x) \tag{3-11}$$

Die Division von Gleichung (3-11) durch Gleichung (3-9) führt zu folgender Gleichung:

$$P^*(Basisphrase, m_1, m_2, ...) = P(Basisphrase, m_1, m_2, ...) \cdot \frac{P_{\mathrm{m}}(m_x)}{P(m_x)} \tag{3-12}$$

Die Nomenklatur $P^*(\cdot)$ bezeichnet im Folgenden die Wahrscheinlichkeiten, die unsicheren Wortbeobachtungen Rechnung tragen. Bei Gleichung (3-12) handelt es sich um die Rechenvorschrift zur Berücksichtigung unsicherer, externer Informationen. Hiervon wird für die Abbildung unsicherer Wortbeobachtungen auf entsprechenden Knoten eines Bayes'schen Netzes Gebrauch gemacht.

Bei der Randwahrscheinlichkeit eines Merkmalsknotens handelt es sich um die Annahme, dass ein Wort aus dessen Vokabular Teil der zu evaluierenden Wortkettenhypothese ist. Die Randwahrscheinlichkeit reflektiert dabei ebenfalls Beobachtungen anderer Wörter derselben Phrase in der Weise, dass alle Wörter einer Phrase syntaktisch und semantisch miteinander verwandt sind. Diese syntaktisch-semantischen Beziehungen zwischen den einzelnen Wörtern wird durch die UND-Modellierung der Basisphrasennetze erreicht. Je mehr Wörter einer Phrase bereits beobachtet wur-

den, desto höher die Erwartung, weitere Wörter dieser Phrase zu beobachten. Diesem Umstand ist bei der Abbildung unsicherer Wort-Beobachtungen auf die Basisphrasen-Netzwerks Rechnung zu tragen, um eine syntaktisch-semantische Gewichtung der Beobachtung zu ermöglichen. Dabei gilt es, beide Informationen, die quantitative Erwartung eines Wortes und die unsichere Information des Spracherkenners, sinnvoll miteinander zu verknüpfen. Abbildung 3.13 verdeutlicht diese Verknüpfung grafisch.

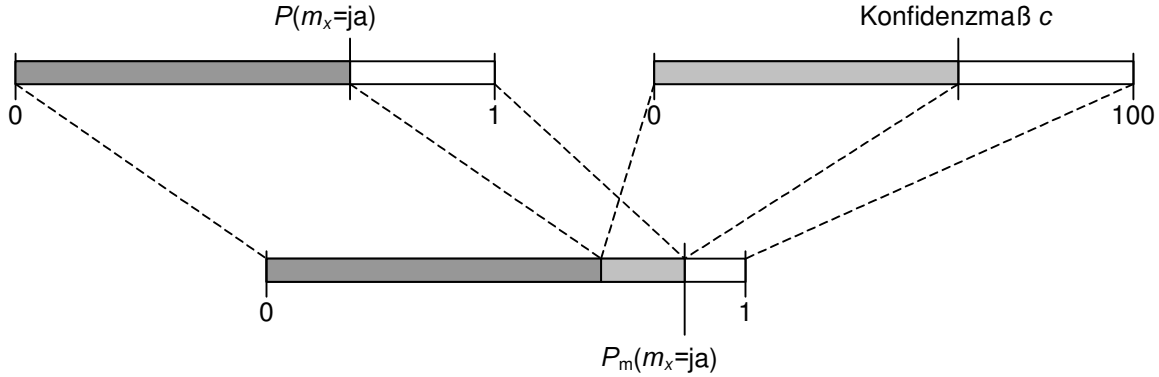


Abbildung 3.13: Fusion der Konfidenzmaße syntaktisch-semantischer und akustischer Informationen

Zunächst wird die Randwahrscheinlichkeit der Zustandsvariable, auf die eine Beobachtung abgebildet werden soll, berechnet. In Abbildung 3.13 wird diese Zustandsvariable allgemein mit m_x bezeichnet. Von Interesse ist nur der Wahrscheinlichkeitswert des Zustands, der die Annahme einer Beobachtung des entsprechenden Wortes modelliert, d.h. $P(m_x=\text{ja})$. Dieser Wert impliziert vorherige Beobachtung syntaktisch-semantisch verwandter Worte der gleichen Phrase und entspricht aus Sicht des Basisphrasennetzes der unsicheren Information aus einer internen Informationsquelle. Die unsichere Information der externen Informationsquelle, d.h., das erkannte Wort und sein Konfidenzmaß c, dienen nun dazu, die Unsicherheit bezüglich der internen Information zu verringern. Aus diesem Grund wird das Konfidenzmaß derart auf das Intervall 1-$P(m_x=\text{ja})$ projiziert, dass ein minimales Konfidenzmaß c (Wert 0) keiner Änderung der Unsicherheit und ein maximaler Wert (100) einer sicheren Beobachtung entspricht. Unter Berücksichtigung des Wertebereichs des Konfidenzmaßes c von 0 bis 100 entspricht dies folgender Gleichung:

$$\begin{aligned} P_{\text{m}}(m_x = \text{ja}) &= P(m_x = \text{ja}) + \frac{c}{100}(1 - P(m_x = \text{ja})) \\ &= P(m_x = \text{ja}) + \frac{c}{100} P(m_x = \text{nein}) \end{aligned} \tag{3-13}$$

Die durch eine Beobachtung entstehende neue Randwahrscheinlichkeit $P_{\text{m}}(m_x)$ wird anhand von Gleichung (3-12) in das entsprechende Basisphrasen-Netz eingespeist. Merkmalsknoten, deren Wörter nicht Teil der Wortkettenhypothesen sind, werden nicht manipuliert.

Für das Parsing von Wortkettenhypothesen nach Wörtern von Operator-Basisphrasen wird die Reihenfolge der Wörter aus Komplexitätsgründen nicht berücksichtigt. Mittels eines Operator-Basisphrasen-Netzes, das den Operator der Phrase „Ich möchte in die Barerstraße“ modelliert, kann auch die Phrase „In die Barerstraße möchte ich“ berücksichtigt werden. Für das Parsing nach Wörtern von Parameter-Basisphrasen ist die Reihenfolge der Wörter von großer Bedeutung und wird deshalb berücksichtigt. Darauf wird an der entsprechenden Stelle näher eingegangen.

④ Für jede Operator-Basisphrase wird die Randwahrscheinlichkeit des Wurzelknotens $P_{\mathrm{Op}}{}^{*}(Operator-Basisphrase)$ aus ihrer Verbundwahrscheinlichkeit berechnet. Der Wert des ja-Zustands reflektiert die Annahme, dass diese Phrase Teil der Wortkettenhypothese ist. Die Operator-Basisphrase, die am besten durch die Wortkettenhypothese abgedeckt wird, zeichnet sich durch den größten Wahrscheinlichkeitswert $P_{\mathrm{Op}}{}^{*}(Operator-Basisphrase=\mathrm{ja})$ aus. Das Maximum unter allen *x* Operator-Basisphrasen wird ermittelt:

$$P_{\mathrm{Operator}}=\max\{P_{\mathrm{Op}}{}^{*}(Operator-Basisphrase\ x=\mathrm{ja})\} \tag{3-14}$$

Der Wert P_{Operator} wird nun dem Operatorknoten des Intentionsnetzes entsprechend Gleichung (3-12) als externe Information zugewiesen:

$$P_{\mathrm{Operator}} \quad \mapsto \quad P_{\mathrm{I}}{}^{*}(Operator=\mathrm{ja}) \tag{3-15}$$

Die Beobachtung einer Operator-Basisphrase erhöht somit die Annahme, einen syntaktisch und semantisch verwandten Parameter zu beobachten, d.h., Beobachtungen bezüglich eines Parameter werden im Folgenden stärker gewichtet.

⑤ Analog zu Schritt ② wird die Randwahrscheinlichkeit des Parameterknotens $P_{\mathrm{I}}{}^{*}(Parameter)$ den Wurzelknoten aller Parameter-Phrasen-Netze als a-priori-Wahrscheinlichkeit zugewiesen:

$$P_{\mathrm{I}}{}^{*}(Parameter) \quad \mapsto \quad P_{\mathrm{P}}{}^{*}(Parameter-Phrase) \tag{3-16}$$

Somit werden alle Parameter-Phrasen-Netze auf die bisherigen Beobachtungen eines syntaktisch-semantisch adäquaten Operators abgestimmt.

⑥ In Analogie zu Schritt ⑤ wird die Randwahrscheinlichkeit des ersten Parameter-Basisphrasen-Knotens $P_{\mathrm{P}}{}^{*}(Parameter-Basisphrase1)$ den Wurzelknoten aller entsprechenden Parameter-Basisphrasen-Netze zugewiesen:

$$P_{\mathrm{P}}{}^{*}(Parameter-Basisphrase1) \quad \mapsto \quad P_{\mathrm{PB}}{}^{*}(Parameter-Basisphrase) \tag{3-17}$$

Nun können auch auf der Parameter-Basisphrasen-Ebene vorhergehende Operator-Beobachtungen berücksichtigt werden.

⑦ Das Abbilden von Wortbeobachtungen auf Parameter-Basisphrasen-Netze unterscheidet sich von der Abbildung auf Operatornetze in verschiedenen Aspekten. Die Parameter-Basisphrasen einer Parameter-Phrase werden nicht einzeln, sondern in ihrem Zusammenspiel betrachtet. Hierzu werden für jede Parameter-Phrase alle potenziellen Parameter-Äußerungen gebildet, indem aus den korrespondierenden Parameter-Basisphrasen alle möglichen Kombinationen gebildet werden. Der Vorteil wird anhand von Abbildung 3.14 dargestellt. Die Abbildung zeigt eine Äußerung, die aus einer Operator-Phrase und zwei Parameter-Basisphrasen besteht. Darunter ist ein typisches Ergebnis der Merkmalsextraktion, eine Wortkettenhypothese, die nur einige durch den Spracherkenner korrekt klassifizierte Worte enthält, zu sehen. Am unteren Rand werden zwei durch ein Intentionsmodell repräsentierte Äußerungen gezeigt. Die linke Äußerung besteht aus zwei Parameter-Basisphrasen,

während die rechte Äußerung lediglich aus einer besteht. Bei dem Versuch, die einzelnen Wörter der Wortkettenhypothese mit den modellierten Äußerungen in Bezug zu setzen, können vier Merkmale auf die linke Äußerung in der korrekten Weise abgebildet werden. Auf die rechte Äußerung werden nur drei Wörter abgebildet, die zudem auf eine inhaltlich falsche Aussage schließen lassen und somit die Ermittlung der wahren Benutzerintention verhindern.

Die Kombination mehrerer Parameter-Basisphrasen vor dem eigentlichen Evaluierungsprozess ermöglicht eine intelligentere Merkmalsselektion und somit eine robustere Interpretation einer Äußerung. Im linken Fall wird zwar die erste Parameter-Basisphrase schlecht durch die Wortkette abgedeckt, dafür werden die Schlüsselworte der folgenden Basisphrase exakt beobachtet. Die intentionsbasierte Interpretation gibt der linken Äußerungshypothese auf Grund einer Gewichtung der beiden syntaktisch-semantisch verwandten Parameter-Basisphrasen gegenüber der rechten Variante den Vorrang. Entscheidend ist dabei die aus Sicht des Intentionsmodells größere Ordnung der Wortkettenhypothese bezüglich der ersten Alternative, da das Wort „3“ syntaktisch-semantisch sehr gut passt; für die andere Alternative bleibt dieses Wort unberücksichtigt.

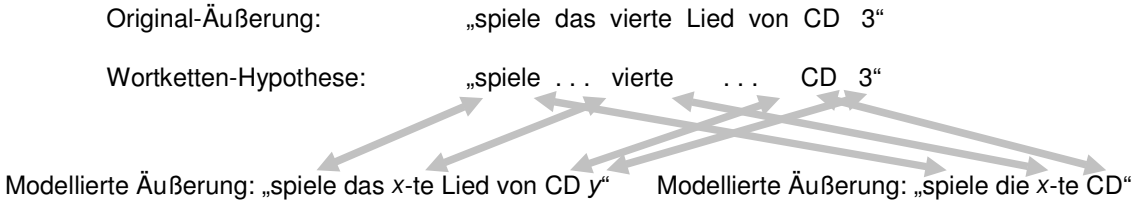


Abbildung 3.14: Merkmalsselektion durch Kombinieren syntaktisch-semantisch verwandter Parameter-Basisphrasen

Der durch Kombination von Parameter-Basisphrasen entstandene Parameterraum bildet die Grundlage für das Parsing, das Absuchen der Wortkettenhypothese nach den Wörtern, die den einzelnen Merkmalsknoten der Basisphrasennetze zugeordnet sind. Zunächst wird die zu evaluierende Wortkettenhypothese bezüglich der Wörter, die den Schlüsselwortknoten zugeordnet sind, untersucht. Danach folgen die optionalen Worte. Bei jeder Abbildung ist die Reihenfolge einzuhalten.

Abbildung 3.15 zeigt die Parsingbereiche für die einzelnen Worte einer aus zwei Parameter-Basisphrasen bestehenden Parameter-Phrase. Zunächst folgt das Parsing für das erste Schlüsselwort der ersten Basisphrase über die gesamte Äußerung. Das Wort „Lied“ wurde gefunden und markiert somit den Startbereich für das Parsing des zweiten Schlüsselwortes. Die Wörter „Lied“ und „4“ wurden gefunden und markieren den Anfangs- und den Endbereich für die Suche nach dem optionalen Wort „Nummer“, da dieses Wort in diesem Fall zwischen den beiden Schlüsselworten zu finden ist. Der Startpunkt für das Parsing nach Wörtern der zweiten Basisphrase ist durch die Beobachtung des letzten Wortes der vorhergehenden Basisphrase gegeben. Die weiteren Parsingschritte gestalten sich in Analogie zu der ersten Basisphrase.

Das Abbilden einer adäquaten Wortbeobachtung auf ein Parameter-Basisphrasen-Netz wird entsprechend der Gleichungen (3-12) und (3-13) vollzogen. Die Wahrscheinlichkeiten, die sowohl Beobachtungen bezüglich der Operator-Basisphrase als auch aller in Betracht zu ziehenden Parameter-Basisphrasen quantitativ modellieren, werden im Folgenden mit $P^{**}(\cdot)$ beschrieben.

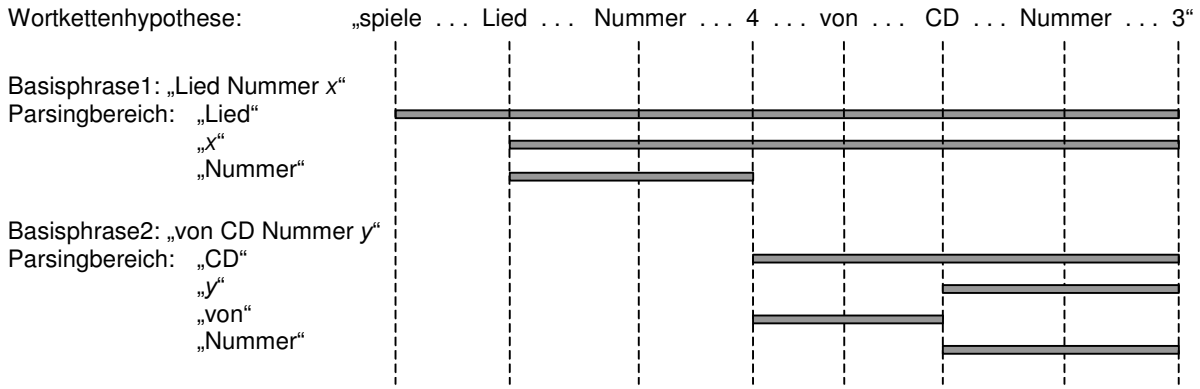


Abbildung 3.15: Parsingbereiche für die einzelnen Wortbeobachtungen

⑧ Für jede der Basisphrasen-Kombinationen ist folgendes Procedere notwendig. Nach dem Abbilden der Wortkettenhypothese auf die erste Basisphrase wird die Randwahrscheinlichkeit $P_{\mathrm{PB}}^{**}(Parameter - Basisphrase1)$ dem korrespondierenden Knoten des Parameter-Phrasen-Netzes zugewiesen:

$$P_{\mathrm{BP}}^{**}(Parameter - Basisphrase1) \quad \mapsto \quad P_{\mathrm{P}}^{**}(Parameter - Basisphrase\,1) \tag{3-18}$$

Analog zu den Schritten ⑥ und ⑦ werden die übrigen Basisphrasen der Parameter-Phrase entsprechend behandelt.

⑨ Für jede Parameter-Phrase wird die wahrscheinlichste Basisphrasenkombination bestimmt. Hierzu dient der Wert der Randwahrscheinlichkeit $P_{\mathrm{P}}^{**}(Parameter - Phrase = \mathrm{ja})$, die quantitative Annahme, dass der betrachtete Parameter Teil der Wortkettenhypothese ist.

Für jede Parameter-Phrase wird ein quantitatives Evaluierungsmaß berechnet, wobei für jede Parameter-Phrase lediglich die beste Basisphrasen-Kombination berücksichtigt wird. Zunächst wird ein Maß für die Vollständigkeit einer Phrasenbeobachtung bestimmt:

$$C_{\mathrm{Parameter}} = \frac{P_{\mathrm{P}}^{**}(Parameter - Phrase = \mathrm{ja}) - P_{\mathrm{P}}(Parameter - Phrase = \mathrm{ja})}{1 - P_{\mathrm{P}}(Parameter - Phrase = \mathrm{ja})} \cdot \frac{n_{\mathrm{P}}}{3} \tag{3-19}$$

Gleichung (3-19) setzt den Einfluss der Wortbeobachtungen auf das Intentionsmodell mit dem maximal möglichen Einfluss in Bezug. Der Wert $C_{\mathrm{Parameter}}$ repräsentiert demnach die Vollständigkeit einer Phrasenbeobachtung, die sowohl den einzelnen Wortbeobachtungen sowie den syntaktisch-semantischen Beziehungen zwischen den Wort-Beobachtungen Rechnung trägt. Da Parameter-Phrasen aus unterschiedlich vielen Parameter-Basisphrasen zusammengesetzt sein können, ist eine Normierung nötig. Hierzu wird die Anzahl der Parameter der betrachteten Parameter-Phrase n_{P} durch die maximal zulässige Anzahl von Parametern ($n_{\max} = 3$) dividiert.

Um eine Parameter-Phrase quantitativ zu evaluieren, wird der $C_{\mathrm{Parameter}}$ auf den Wurzelknoten des entsprechenden Parameter-Phrasen-Netzes abgebildet. $C_{\mathrm{Parameter}}$ wird zunächst auf den Wahrscheinlichkeitsbereich dieses Wurzelknotens $1 - P_{\mathrm{P}}^{**}(Parameter - Phrase = \mathrm{ja})$ projiziert. Dieser Bereich entspricht dem Wertebereich, der für Wortbeobachtungen zur Verfügung steht. Ein Dämpfungsfaktor schränkt die syntaktisch-semantische Gewichtung ein, da diese gerade bei Phrasen mit mehreren

Parametern stark ansteigt. Der Dämpfungsfaktor wird aus der Anzahl der beobachteten Parameter n_{iP} und der Anzahl der Parameter n_{P} der betrachteten Parameter-Phrase bestimmt. Gleichung (3-20) fasst dies zusammen:

$$\begin{aligned}\Delta P &= (1 - P_{\mathrm{P}}(Parameter - Phrase = \mathrm{ja})) \cdot C \cdot \sqrt{\frac{n_{\mathrm{iP}}}{n_{\mathrm{P}}}} = \\ &= \frac{1}{3}(P_{\mathrm{P}}^{**}(Parameter - Phrase = \mathrm{ja}) - P_{\mathrm{P}}(Parameter - Phrase = \mathrm{ja}))\sqrt{n_{\mathrm{iP}} n_{\mathrm{P}}}\end{aligned} \tag{3-20}$$

Schließlich wird ΔP direkt auf den Wurzelknoten abgebildet:

$$E_{\mathrm{P}} = P_{\mathrm{P}}(Parameter - Phrase = \mathrm{ja}) + \Delta P \tag{3-21}$$

Dieses Evaluierungsmaß E_{P} beschreibt die Beobachtung der betrachteten Parameter-Basisphrasen-Kombination quantitativ.

⑩ Die Randwahrscheinlichkeit $P_{\mathrm{P}}^{**}(Parameter - Phrase)$ der Parameter-Basisphrasen-Kombination mit dem maximalen Evaluierungsmaß E_{P} wird dem Parameterknoten des Intentions-Netzes zugewiesen:

$$P_{\mathrm{P}}^{**}(Parameter - Phrase) \quad \mapsto \quad P_{\mathrm{I}}^{**}(Parameter) \tag{3-22}$$

⑪ Anhand des Intentionsknotens kann direkt der Evaluierungswert E zur Beschreibung der Beobachtung der betrachteten Intention berechnet werden:

$$E = P_{\mathrm{I}}^{**}(Intention = \mathrm{ja})$$

Dieses Maß reflektiert die anhand einer bestimmten Wortkettenhypothese am sichersten beobachtete Operator-Basisphrase, die am sichersten beobachtete Parameter-Basisphrasen-Kombination sowie deren semantisch-syntaktischen Beziehungen zueinander auf Wort- und Phrasenebene.

Die erläuterten Schritte ① bis ⑪ sind für alle Intentionsmodelle und für alle Wortkettenhypothesen durchzuführen. Die Konstellation Intentionsmodell/Wortkettenhypothese mit dem maximalen Evaluierungsmaß E repräsentiert aus Sicht von *Insense* die Benutzerintention. Aus der besten Parameter-Basisphrasen-Kombination dieser Konstellation können dann direkt Systemaktionen erzeugt werden, die schließlich die Applikation in den gewünschten Zustand versetzen.

Abbildung 3.16 zeigt die Evaluierungsmaße E für die in Abbildung 3.4 dargestellten 20 Wortkettenhypothesen für alle Intentionshypothesen der Intentionsbibliothek. Die korrekte Intention „Anruf“ wurde für alle Wortkettenhypothesen als die wahrscheinlichste Intentionshypothese bewertet. *Insense* kann daraus direkt die Systemaktion *Anruf(Hunsinger)* ableiten und somit adäquat auf die Benutzereingabe reagieren. Durch die unpräzise Formulierung der spontansprachlichen Äußerung wird nicht nur die tatsächliche Benutzerintention mit hohen Evaluierungsmaßen bewertet, auch falschen Intentionshypothesen wurden hohe Evaluierungsmaße zugewiesen, da einige vom Benutzer verwendete Worte nicht Teil des Spracherkennervokabulars waren und somit Teile der Äußerung eventuell auf Worte abgebildet wurden, die anderen Intentionen zugeordnet werden konnten. Durch

die syntaktisch-semantische Gewichtung von Beobachtungen und die Merkmalsselektion des intentionsbasierten Ansatzes konnte die Benutzereingabe dennoch korrekt interpretiert werden.

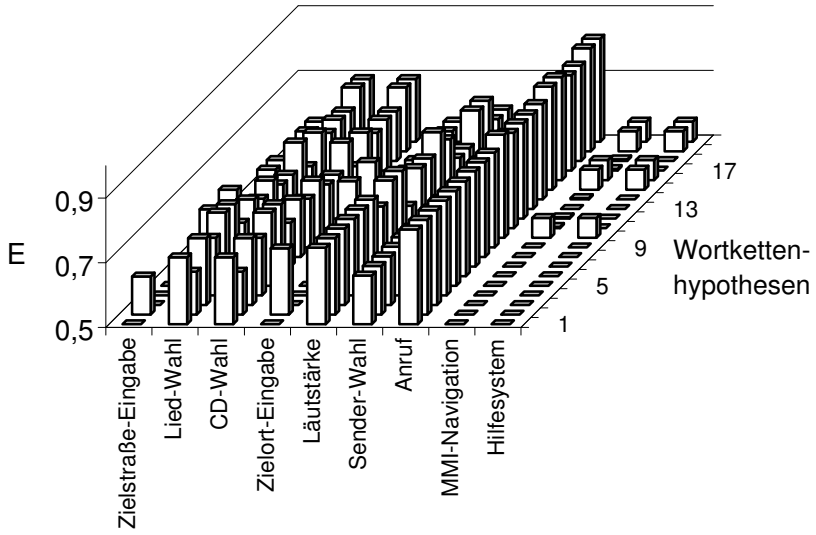


Abbildung 3.16: Evaluierungsmaße aller Intentionshypothesen für die in Abbildung 3.4 dargestellten 20 Wortkettenhypothesen

3.8 Ergebnisse und Diskussion

In diesem Kapitel werden einige Aspekte von *Insense* näher diskutiert, sowie die Ergebnisse verschiedener Untersuchungen vorgestellt. Zunächst wird auf die zur Evaluierung des Systems verwendeten Testäußerungen näher eingegangen.

3.8.1 Testäußerungen

Das sprachverstehende System wurde mit 250 Testäußerungen evaluiert. Diese Äußerungen wurden in drei Gruppen eingeteilt, um die Leistung des sprachverstehenden Systems abhängig von der Qualität der zu analysierenden Äußerung zu diskutieren. Die Äußerungen dieser drei Gruppen zeichnen sich durch folgende Eigenschaften aus:

- **Von *Insense* vollständig berücksichtigte Äußerungen**
  Diese Äußerungen verursachen keine Out-of-Vocabulary-Probleme (OoV) für den Spracherkenner und wurden syntaktisch-semantisch korrekt formuliert und sauber ausgesprochen. Beispiele hierfür sind folgende Äußerungen:

  „Bringe mich in die Barerstraße nach München“

  „Ich würde gerne Lied zwei von der dritten CD hören“

  Diese Art Äußerung kann vor allem bei Benutzern mit Berührungsängsten bezüglich neuer Technologien beobachtet werden. Der Benutzer ist sich zwar der Tatsache bewusst, mit einer

Maschine zu kommunizieren, verhält sich aber bei der Interaktion mit der Maschine ähnlich wie bei der Artikulation gegenüber einem Menschen anderer Muttersprache, d.h., die Formulierungen sind syntaktisch-semantisch korrekt und betont klar ausgesprochen. Dieser Äußerungstyp wird in den Untersuchungen durch 100 Testäußerungen berücksichtigt.

- **Kommandosprachliche Äußerungen**
  Diese Art der Artikulation ist für den Menschen unnatürlich, wird aber gerade bei guter Kenntnis der Applikation häufig verwendet; sie orientiert sich ausschließlich an der Architektur des zu bedienenden Systems. Typische kommandosprachliche Äußerungen lauten zum Beispiel:

  „Zieleingabe Barerstraße, München“

  „CD drei, Lied zwei“

  Für die Evaluierung wurden 50 kommandosprachliche Äußerungen verwendet.

- **Spontansprachliche Äußerungen**
  Spontansprachliche Äußerungen können sich unter Umständen durch syntaktisch-semantisch unsaubere Formulierungen und eventuell undeutlicher Aussprache auszeichnen. Dies kann darauf zurückzuführen sein, dass sich der Benutzer Gedanken über seine eigene Intention macht oder seine Absicht ändert, während er bereits spricht. Situationsbedingte Ablenkung kann ebenfalls als Ursache beobachtet werden. Dialekte, weggelassene Wortendungen führen in der Regel zu OoV-Phänomenen, die durch die sprachverstehenden Komponenten kompensiert werden können. Spontansprachliche Äußerungen könnten folgendermaßen aussehen:

  „Bring‘ mich doch bitte mal nach ähhh in die ähmmm Barerstraße, und zwar in München natürlich.“

  „Jetzt möcht‘ ich mal Lied Nummer ähmm welches Lied war das nochmal, Lied zwei von der ähmm dritten CD hören.“

  Da spontansprachliche Äußerungen dem zwischenmenschlichen Dialog am nächsten kommen, hat eine fehlerfreie Interpretation gerade dieser Äußerungen höchste Priorität auf dem Gebiet der Spracherkennung und des Sprachverstehens. Allerdings stellt die Interpretation spontansprachlicher Äußerungen aus erwähnten Gründen auch die größte Herausforderung dar. Die Testdaten beinhalten 100 spontansprachliche Äußerungen.

Die folgende Abbildung 3.17 zeigt, durch wie viele Testäußerungen die sprachliche Manipulation der einzelnen Systemfunktionen abgedeckt ist. Während die Systemfunktion „Telefon“ nur einen Anruf einer Person zulässt, kann die Manipulation anderer Systemfunktionen durch Angabe mehrerer Parameter näher spezifiziert werden. Die Wahl eines bestimmten Liedes kann zum Beispiel durch die Angabe der CD-Nummer und der gewünschten Lautstärke ergänzt werden. Somit lassen sich die, abhängig von der Systemfunktion, sehr unterschiedlichen Zahlen von Testäußerungen erklären.

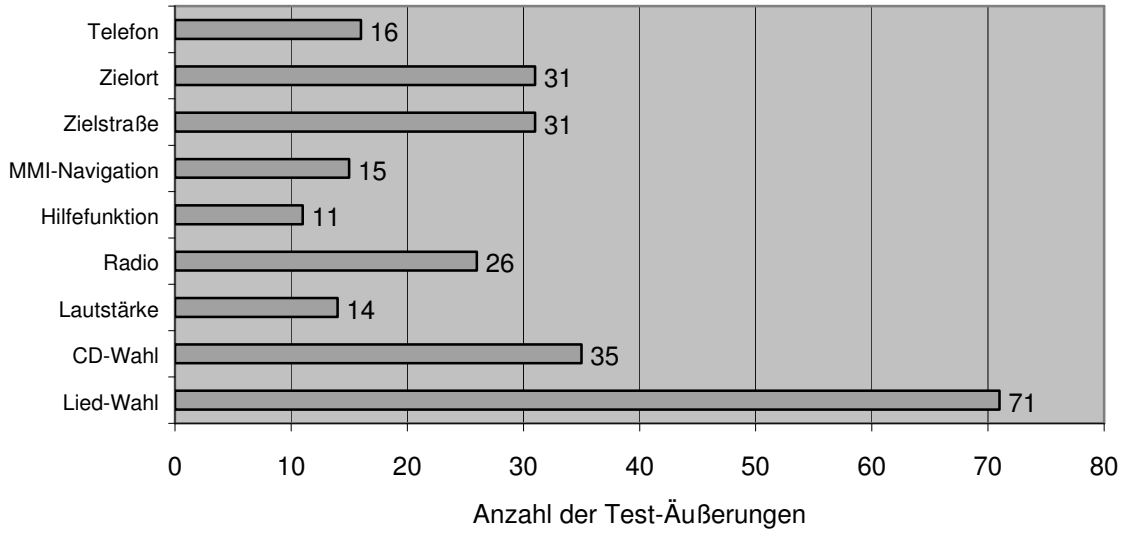


Abbildung 3.17: Testkorpus zur Evaluierung von *Insense*

3.8.2 Evaluierung von *Insense*

Das sprachverstehende System *Insense* wurde anhand der im vorherigen Abschnitt erwähnten 250 Testäußerungen quantitativ evaluiert. Bei der Erstellung von Erkennungsraten wird jeweils zwischen der Leistung des Gesamtsystems und der Leistung der syntaktisch-semantischen Evaluierung (SSE) unterschieden, um dokumentieren zu können, ob eine Fehlklassifikation der Benutzerintention auf Erkennungsfehler des Spracherkenners oder auf einen Fehler der syntaktisch-semantischen Evaluierung zurückzuführen ist. Eine Fehlklassifikation der Intention ist dann auf den Spracherkenner zurückzuführen, wenn die generierten Wortkettenhypothesen keine der für die syntaktisch-semantische Evaluierung relevanten Schlüsselworte beinhalten.

Wenn zum Beispiel der Benutzer das fünfte Lied hören möchte, keine der Wortkettenhypothesen aber die entsprechende Liednummer aufweist, kann auch die syntaktisch-semantische Analyse nicht die Benutzerintention korrekt klassifizieren. Fehler dieser Art wurden demnach als Spracherkennungsfehler behandelt, die sich in der Erkennungsrate für das Gesamtsystem niederschlagen.

Um die Klassifikationsleistung der eigentlichen Innovationen, der Intentionsmodelle und der syntaktisch-semantische Evaluierung zu dokumentieren, wurden als Fehlklassifikation ausschließlich jene Klassifikationsfehler gewertet, die eindeutig nicht auf den Spracherkenner zurückzuführen sind, sondern auf die sprachverstehenden Komponenten. Bei Fehlern der verstehenden Komponente weisen die zu evaluierenden Wortkettenhypothesen alle relevanten Schlüsselworte auf, dennoch konnte die Benutzerintention nicht korrekt ermittelt werden.

Um die prinzipielle Funktionsweise der syntaktisch-semantischen Evaluierung überprüfen zu können, wurde eine fehlerfreie Klassifikationsleistung des Spracherkenners simuliert, indem Wortkettenhypothesen generiert wurden, die einer syntaktisch-semantisch korrekten Aussage entsprechen. Die Konfidenzmaße der einzelnen Merkmale wurden hierzu auf 75,0 gesetzt. Bei allen getesteten Wortketten war die syntaktisch-semantische Komponente in der Lage, die Intention fehlerfrei zu

ermitteln, d.h., im Falle eines idealisierten Spracherkenners beträgt die Klassifikationsleistung von *Insense* 100 Prozent. Dieses Ergebnis lässt darauf schließen, dass der intentionsbasierte Ansatz unter idealen Bedingungen fehlerfrei arbeitet und somit sowohl die Generierung der Intentionsmodelle als auch die syntaktisch-semantische Interpretation von Wortketten keine prinzipiellen Aspekte des Sprachverstehens unberücksichtigt lässt.

Unter realen Bedingungen erzeugt der Spracherkenner allerdings sehr selten syntaktisch-semantisch korrekte Wortketten; zudem beinhalten Wortketten in der Regel viele Worte, die nicht Teil der gesprochenen Äußerung sind und somit erheblich zu Fehlklassifikationen beitragen. Da die Qualität der Wortketten in erster Linie mit der Art, sich dem Spracherkenner gegenüber zu artikulieren, korreliert, wurde die Klassifikationsleistung des Gesamtsystems und der syntaktisch-semantischen Evaluierung (SSE) anhand der im vorhergehenden Kapitel erläuterten Äußerungsarten evaluiert. Abbildung 3.18 zeigt diese Erkennungsraten für fehlerfrei formuliert und ausgesprochene, kommandosprachliche sowie spontansprachliche Äußerungen. Zusätzlich wird die Gesamterkennungsleistung des Systems angegeben, d.h. die Erkennungsleistung für die gesamten 250 Testäußerungen. Zunächst werden die Erkennungsraten für die drei Testkorpi einzeln diskutiert.

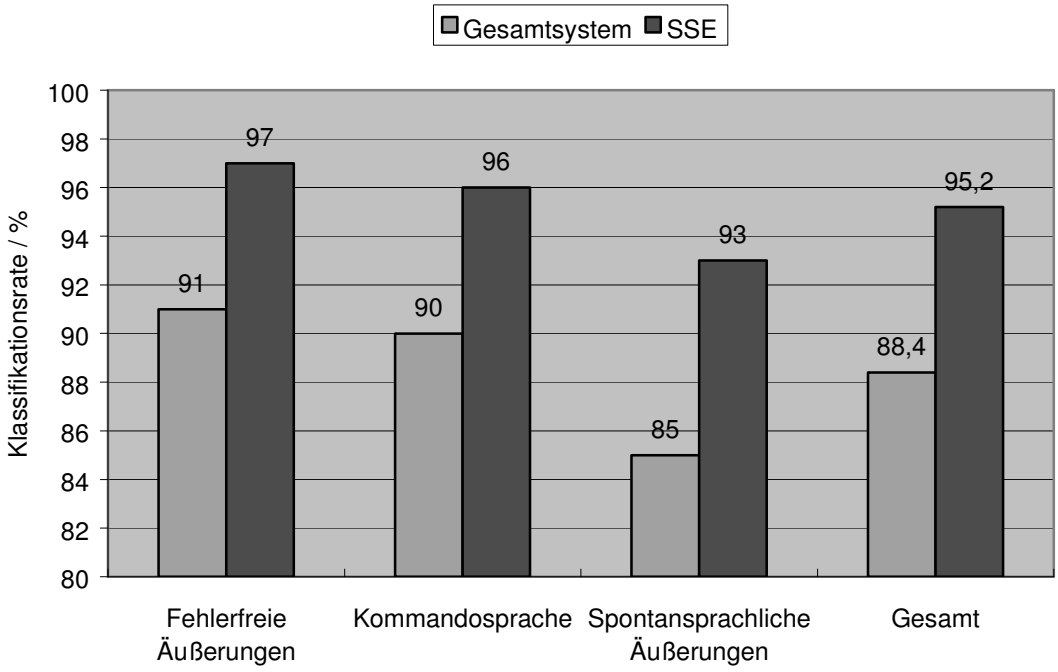


Abbildung 3.18: Klassifikationsrate von *Insense* in Abhängigkeit von der Qualität der Testäußerungen

Wie der Abbildung zu entnehmen ist, liegen die Klassifikationsraten des Gesamtsystems niedriger als die Erkennungsleistung der syntaktisch-semantischen Komponenten. Diese Differenz ist ausschließlich auf Fehlklassifikationen des Spracherkenners zurückzuführen. Erwartungsgemäß ist diese Differenz bei Spontansprache ausgeprägter als bei den übrigen Äußerungen, da in diesem Fall häufiger mit OoV-Phänomenen zu rechnen ist.

Die Abhängigkeit der Erkennungsleistung der syntaktisch-semantischen Komponenten von der Qualität der Äußerungen ist geringer, weil die Nutzung syntaktisch-semantischer Beziehungen zwi-

schen Wörtern und Phrasen den Einfluss falsch erkannter Worte kompensiert und somit zu einer robusten Interpretation einer Äußerung beitragen. Da Syntax und Semantik durch die Intentionsmodelle erfasst werden, spricht dies generell für die Verwendung des intentionsbasierten Ansatzes für das Sprachverstehen. Die Ergebnisse belegen darüber hinaus, dass es sich bei der Merkmalsselektion um einen sehr mächtigen Aspekt des intentionsbasierten Ansatzes handelt. Dieser Mechanismus gewährleistet eine gezielte Suche relevanter Merkmale auch in fehlerbehafteten Wortketten. Ein gravierender Einfluss des Out-of-Vocabulary-Phänomens kann auf diese Weise weitestgehend vermieden werden.

Gemittelt über alle Testäußerungen ist festzuhalten, dass die Erkennungsraten der intentionsbasierten Interpretation von Benutzeräußerungen auf sehr hohem Niveau liegen. Eine höhere Robustheit ist in erster Linie durch die Verwendung eines leistungsfähigeren Spracherkenners für die Merkmalsextraktion möglich.

3.8.3 Untersuchungen und Diskussion

Neben der Evaluierung von *Insense* anhand von Klassifikationsraten wurden zusätzliche Untersuchungen durchgeführt, um die intentionsbasierte Interpretation zu diskutieren. Die Ergebnisse einiger Analysen werden in diesem Abschnitt behandelt.

Für die Systemkomponente, die auf Basis der von *Insense* ermittelten Intention entsprechende Dialogschritte einleitet, ist die Kenntnis des maximalen Evaluierungsmaßes von großer Bedeutung, da im Falle einer Unterschreitung eines Schwellwertes von einer Fehlklassifikation auszugehen ist. Wäre in dieser Situation die Systemfunktion, die der Benutzer per Sprache manipulieren wollte, bekannt, so könnte der Dialog entsprechend gezielt formuliert werden und das System die eigentliche Intention auf sehr effiziente und für den Benutzer komfortable Weise in Erfahrung bringen. Hierzu wurde untersucht, mit welcher Klassifikationsleistung *Insense* die vom Benutzer angesprochenen Systemfunktionen erkennt. Tabelle 3-1 zeigt die Ergebnisse für alle drei Testkorpi.

Weitgehend unabhängig von der Qualität der 250 Testäußerungen liegen die Klassifikationsraten auf einem sehr hohen Niveau. Selbst bei Spontansprache kann ein Dialogmodell im Falle einer Fehlklassifikation der Benutzerintention äußerst zuverlässig auf die von *Insense* interpretierte Systemfunktion zurückgreifen. Diese Möglichkeit ist für einen praktischen Einsatz des sprachverstehenden Systems sehr vorteilhaft.

	Fehlerfreie Äußerungen	Kommandosprache	Spontansprachliche Äußerungen
Klassifikationsrate / %	97,0	98,0	98,0

Tabelle 3-1: Klassifikationsrate der angesprochenen Systemfunktionen in Abhängigkeit von der Qualität der Testäußerung

Zur Interpretation von Äußerungen wertet *Insense* nicht nur die aus Sicht der Spracherkennung wahrscheinlichste Wortkettenhypothese aus, sondern die n besten Hypothesen. Um einen möglichst guten Kompromiss zwischen Klassifikationsleistung und Rechenkomplexität zu finden, wurden die Fehlerraten für Klassifikationsprozesse mit bis zu zwanzig evaluierten Wortkettenhypothesen be-

stimmt. Grundlage hierfür war der Testkorpus mit 100 spontansprachlichen Äußerungen. Abbildung 3.19 zeigt die Ergebnisse.

Die Evaluierung nur auf Basis der aus Sicht des Spracherkenners besten Wortkette führt zu einer sehr hohen Fehlerrate von 25 Prozent. Das Hinzuziehen der zweitbesten Hypothese senkt die Fehlerrate bereits auf 19 Prozent ab. Dies deutet darauf hin, dass die durch Signalanalyse bestimmte zweitbeste Hypothese besser auf den Intentionsraum abgebildet werden könnte als die beste Wortkette. Dies kann als klares Indiz dafür gewertet werden, dass ein *verstehender* Ansatz einem rein *erkennenden* Verfahren überlegen ist. Mit Einbeziehen weiterer Wortkettenhypothesen pendeln sich die Fehlerraten zwischen 14 und 16 Prozent ein.

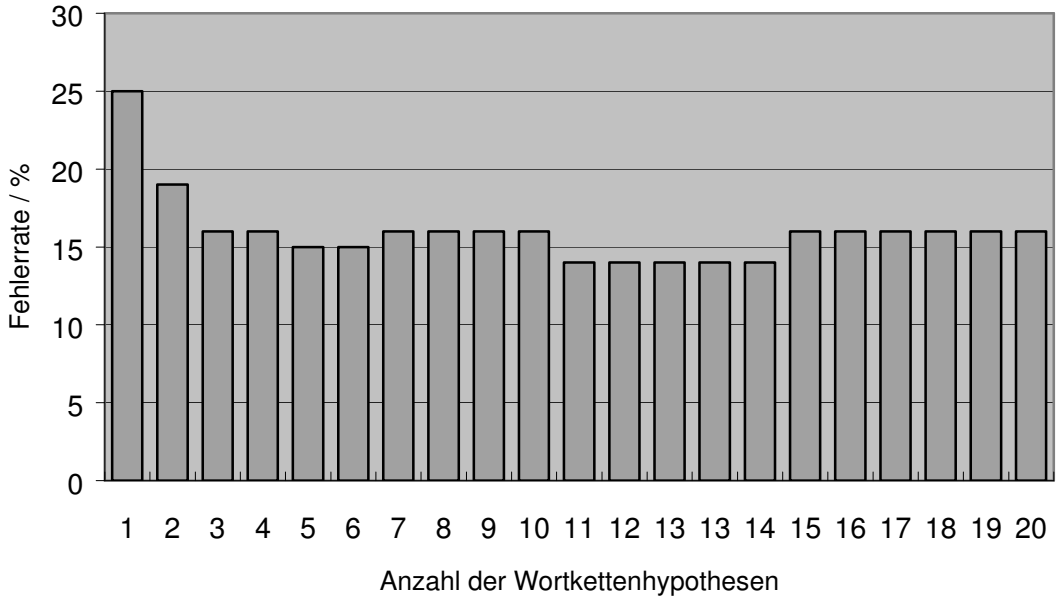


Abbildung 3.19: Zusammenhang zwischen Fehlerrate und Anzahl der evaluierten Wortkettenhypothesen

Der Rechenaufwand für eine intentionsbasierte Interpretation einer Äußerung ist linear abhängig von der Anzahl der evaluierten Wortkettenhypothesen. Im *Insense*-Demonstrator wurden als guter Kompromiss zwischen Klassifikationsleistung und Rechenkomplexität stets die fünf besten Wortkettenhypothesen als Merkmalsvektor herangezogen.

3.9 Implementierung von *Insense*

Dieses Kapitel beschreibt die Implementierung des sprachverstehenden Systems *Insense,* die Realisierung der einzelnen Systemkomponenten sowie die entwickelte Benutzungsoberfläche.

Insense wurde in Analogie zu der in Kapitel 3.2 allgemein diskutierten Systemarchitektur implementiert. Kernkomponente ist die syntaktisch-semantische Evaluierung. Mit dem Start dieses Programmteils werden zunächst die Dateien der Intentionsbibliothek und der Intentionsmodelle eingelesen und die Intentionsmodelle entsprechend initialisiert. Da diese syntaktisch-semantische Komponente sowohl vom Spracherkenner als auch von der *Insense*-Benutzungsoberfläche gestartet

werden kann, arbeitet sie im Zusammenspiel mit den übrigen erwähnten Komponenten als Client. Entsprechend fungieren die Spracherkenneranbindung sowie die *Insense*-Benutzungsoberfläche als Server, die mit der Komponente zur syntaktisch-semantischen Komponente über TCP/IP-Sockets kommunizieren. Abbildung 3.20 stellt dies schematisch dar.

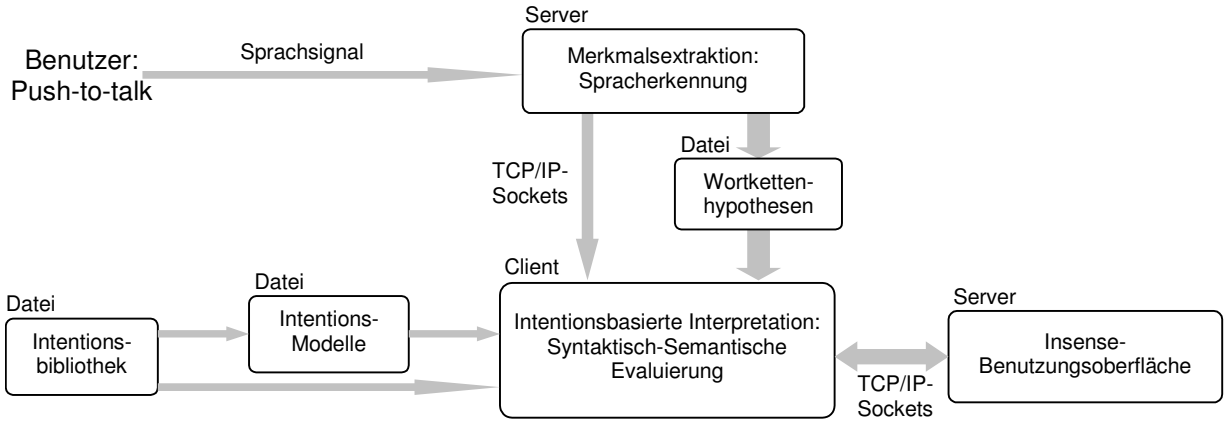


Abbildung 3.20: Systemarchitektur der Implementierung von *Insense*

Die Intentionsbibliothek besteht aus dem in Kapitel 3.4 vorgestellten Intentionsraum. Sie kann kontextabhängig dynamisch eingeschränkt bzw. erweitert werden, um die Klassifikationsrate zu erhöhen und um die benötigte Rechenleistung zu minimieren. Im Rahmen der Implementierung entspricht die Intentionsbibliothek einer Datei mit allen potenziellen Intentionen und Verweisen auf Dateien, deren Informationen zur Intentionsmodellierung benötigt werden.

Für alle Einträge der Intentionsbibliothek werden die Intentionsmodelle aus den einzelnen Basisphrasen-Modellen zusammengesetzt. Für die Definition eines Intentionsmodells wird eine Reihe von Dateien verwendet, die alle Kombinationen der Operator-Phrasen und Parameter-Phrasen enthalten und auf die entsprechenden Basisphrasen verweisen. Die Basisphrasen-Bibliothek hält für jede Basisphrase die Topologie des Bayes'schen Netzes, die bedingten Wahrscheinlichkeiten sowie das Vokabular der einzelnen Wortknoten bereit. Abbildung 3.21 zeigt dies.

Durch die Wahl unterschiedlicher a-priori-Wahrscheinlichkeiten für die Wurzelknoten der Intentionsnetze kann bei der folgenden Evaluierung dem aktuellen Kontext Rechnung getragen werden.

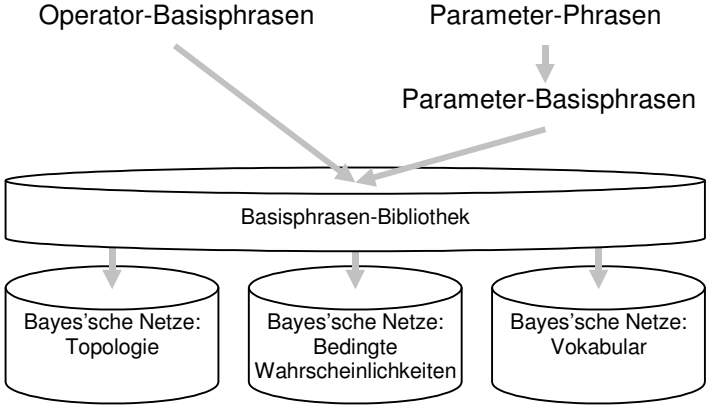
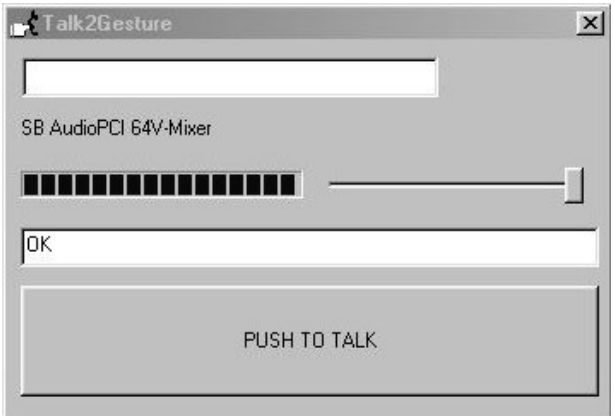


Abbildung 3.21: Dateien zur Definition eines Intentionsmodells

Zur Merkmalsextraktion wurde das Spracherkennungssystem ASR 1600–API von Lernout and Hauspie [L&H98] eingesetzt. Abbildung 3.22a) zeigt die Benutzungsoberfläche der Spracherkenneranbindung [Neu01], realisiert unter Windows 98 mit Visual C++ [Str92]. Das Vokabular des Spracherkenners besteht aus 286 Elementen; der Erkenner wird ohne Sprachmodell [L&H98] betrieben, d.h., Folgewahrscheinlichkeiten zwischen Worten werden vom Erkenner nicht berücksichtigt, da die entsprechenden Statistiken kaum in der Lage sind, spontansprachliche Äußerungen zu modellieren.

Das Hauptbedienelement ist die *Push-to-talk*-Taste, durch deren Betätigung das Sprachsignal aufgenommen und die Analyse des Signals auf akustischer Ebene gestartet wird. Nach Beendigung des Erkennungsprozesses werden die n besten Wortkettenhypothesen in eine Datei geschrieben und die Komponente zur syntaktisch-semantischen Evaluierung durch eine Nachricht über einen TCP/IP-Socket davon in Kenntnis gesetzt. Die Zwischenspeicherung der Wortkettenhypothesen wird dafür benötigt, um eventuell weitere Evaluierungsprozesse auf Basis derselben Wortketten durchführen zu können. Eine zeitliche Verzögerung durch die Zwischenspeicherung ist für den Benutzer nicht wahrnehmbar. Abbildung 3.22b) zeigt eine derartige Datei mit den fünf, aus der Sicht des Spracherkenners, wahrscheinlichsten Wortkettenhypothesen mit den Konfidenzmaßen auf Wortebene.

Nachdem die wahrscheinlichste Intention klassifiziert wurde, werden die korrespondierenden Systemaktionen zur Visualisierung an die Benutzungsoberfläche gesendet. Die Benutzungsoberfläche kann ebenfalls einen Evaluierungsprozess triggern, sodass verschiedene Parametereinstellungen direkt anhand derselben Wortkettenhypothesen evaluierbar sind.



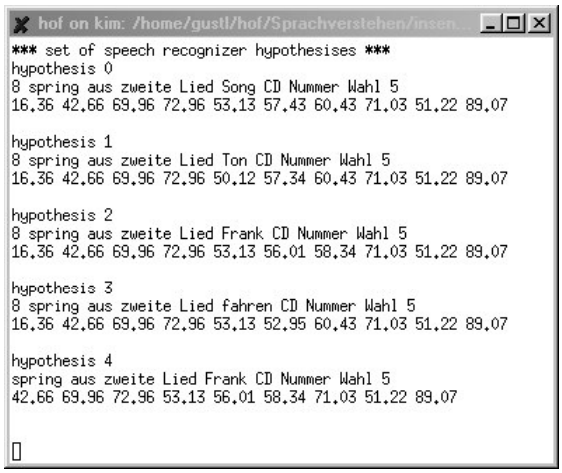


Abbildung 3.22: a) Benutzungsoberfläche des Spracherkenners,
b) Datei mit den n besten Wortkettenhypothesen

Die syntaktisch-semantische Komponente erhält von der Merkmalsextraktion die Nachricht über neu vorliegende Wortkettenhypothesen und liest diese Informationen aus der entsprechenden Datei ein. Nach der intentionsbasierten, syntaktisch-semantischen Evaluierung der Wortkettenhypothesen wird die wahrscheinlichste Folge von Aktionen und Parametern auf der *Insense*-Benutzungsoberfläche (Abbildung 3.23) ausgegeben. Sie dient in erster Linie dazu, die Klassifikationsergebnisse darzustellen und näher zu dokumentieren, aber auch zum Einstellen diverser Systemparameter. Darüber hinaus können die Evaluierungsmaße aller Intentionen für alle Wortkettenhypo-

thesen durch ein Histogramm visualisiert werden. Abbildung 3.24 zeigt ein Beispiel für eine derartige Darstellung.

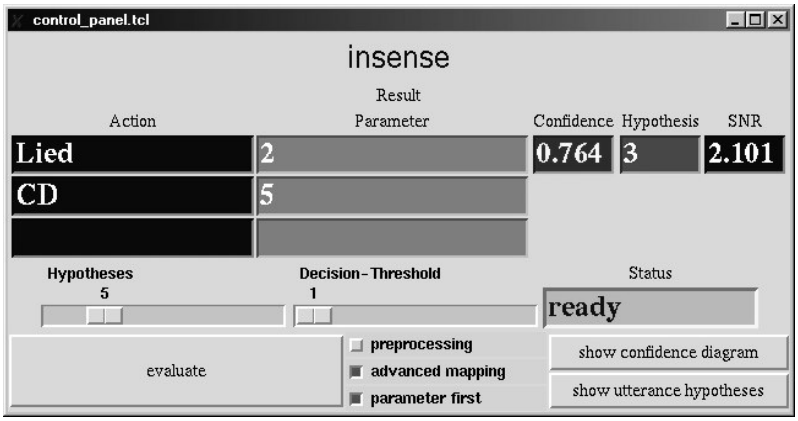


Abbildung 3.23: Benutzungsoberfläche von *Insense*

Die gesamte syntaktisch-semantische Komponente von *Insense* wurde in C/C++ [Str92] unter Linux implementiert. Als Inferenzverfahren der Bayes'schen Netze wurde das Junction-Tree-Verfahren von Jenson [Jen92][Jin99] implementiert.

Die Benutzungsoberfläche wurde in Tcl/Tk [Wel99] als Server realisiert; dieses Skript kommuniziert mit dem Hauptprogramm über bidirektionale TCP/IP-Sockets.

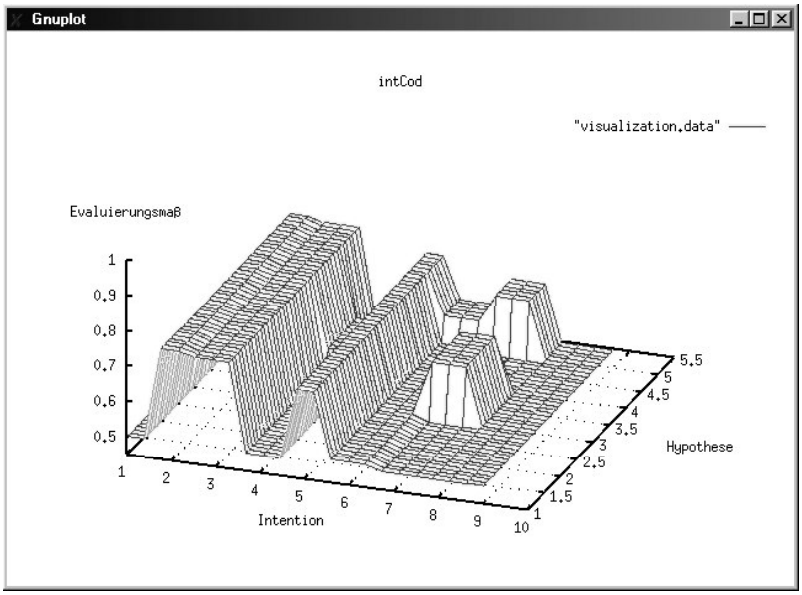


Abbildung 3.24: Visualisierung aller Evaluierungsmaße

Insense wurde nicht nur als Stand-alone-Demonstrator implementiert, sondern spielt im Rahmen des adaptiven, multimodalen Systems *NINIA* (*N*atürliche *In*teraktion *i*m *A*utomobil) eine tragende Rolle. Bei der Domäne handelt es sich um die informationstechnischen Einrichtungen eines Automobils, die mittels spontansprachlicher Äußerungen, dynamischer Handgesten [Mor00][Zob02] und haptischer Eingaben bedient werden können. Systemmittelpunkt ist ein auf Handgestenbedienung optimiertes Bedienkonzept [Gei02], das dem Benutzer durch die Schnittstelle *GeCom* zugänglich

gemacht wird. Das adaptive Hilfesystem *GHelp* [Nie01][Nie02] gewährleistet eine intuitive Interaktion mittels dynamischer Handgesten.

Um ein möglichst realistisches Szenario zu schaffen, wurde das Gesamtsystem in einen Fahrsimulator, bestehend aus einem Fahrzeug der Luxusklasse und diversen Simulationsmöglichkeiten, integriert.

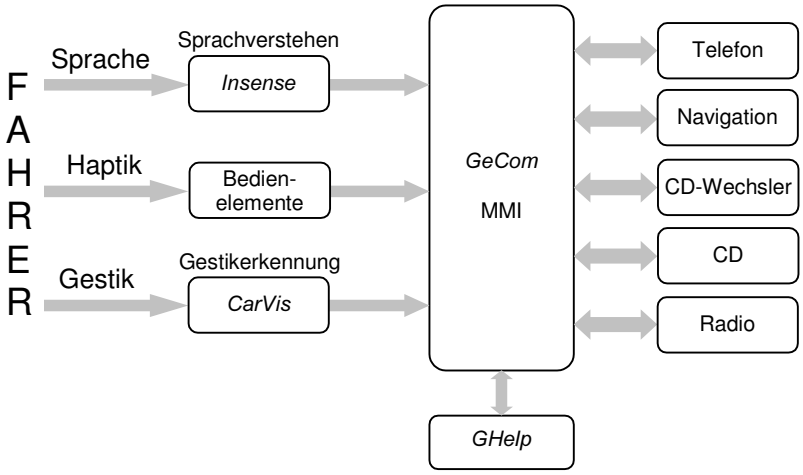


Abbildung 3.25: *Insense* im Rahmen des Systems *NINIA*

Zum Betrieb von *Insense* sind zwei Mikrofone vorgesehen; eines auf dem Armaturenbrett, direkt vor dem Fahrer, ein weiteres in der Nähe des Rückspiegels, direkt auf den Fahrer gerichtet. Das Summensignal dient dann als Eingangssignal für *Insense* um den Einfluss der Blickrichtung auf die Erkennungsleistung zu minimieren, d.h., der Fahrer kann sowohl auf die Fahrbahn als auch auf das zentrale Display blicken, während er dem Fahrzeug spontansprachliche Anweisungen gibt.

Ein Taster am Lenkrad dient als Push-to-Talk-Taste, um die Aufnahme und Analyse des Sprachsignals zu triggern. Im Rahmen dieses *NINIA*-Demonstators werden nur die Klassifikationsergebnisse an Gecom weitergeleitet, deren Evaluierungsmaße einen bestimmten Schwellwert überschreiten. Dies führt zu einem sehr robusten Einsatz von Sprache im Fahrzeug unter realen Bedingungen.

4

Intentionsbasierte Interpretation unvollständiger Aktionssequenzen: Planerkennung

Die Umsetzung komplexer, höherwertiger Intentionen erfordert häufig eine Reihe von Benutzeraktionen. In diesem Kapitel wird die Anpassung des intentionsbasierten Ansatzes für die Interpretation von unvollständigen Aktionssequenzen beschrieben. Bei dem dabei entwickelten System handelt es sich um einen Planerkenner.

4.1 Grundidee

Bei komplexen Softwaresystemen lässt sich die Benutzerintention nicht durch eine Einzelaktion vermitteln, sondern nur durch eine Abfolge von Aktionen, die im folgenden als *Plan* bezeichnet wird. Bei einem Plan handelt es sich somit um eine Reihe von Benutzeraktionen zum Erlangen eines gewünschten Ziels. Hätte eine Software-Applikation die Fähigkeit, anhand der bereits getätigten Benutzeraktionen Aussagen über die Intention des Benutzers zu treffen, so könnte das Erreichen dieses Ziels durch Adaption der Dialogführung erheblich komfortabler und effizienter gestaltet werden. Voraussetzung für eine derartige Adaption ist die Interpretation aller bisherigen Benutzeraktionen.

In diesem Kapitel wird ein Verfahren zur Interpretation und zum *Verstehen* einer Aktionssequenz vorgestellt, ein so genannter *Planerkenner*. Abbildung 4.1 zeigt den grundlegenden Gedanken der Planerkennung. Der Benutzer hat eine bestimmte Intention, für die er sich einen Plan überlegt, eine Aktionssequenz bestehend aus *m* Einzelaktionen. Schritt für Schritt teilt der Benutzer diese Aktionen der Applikation mit, wobei jede dieser Eingaben dem Planerkenner als Eingangsinformationen zur Verfügung stehen. Ziel der Planerkennung ist die Klassifikation des Plans auf Grund aller bisher beobachteten Aktionen möglichst lange, bevor der Benutzer sein Ziel erreicht hat.

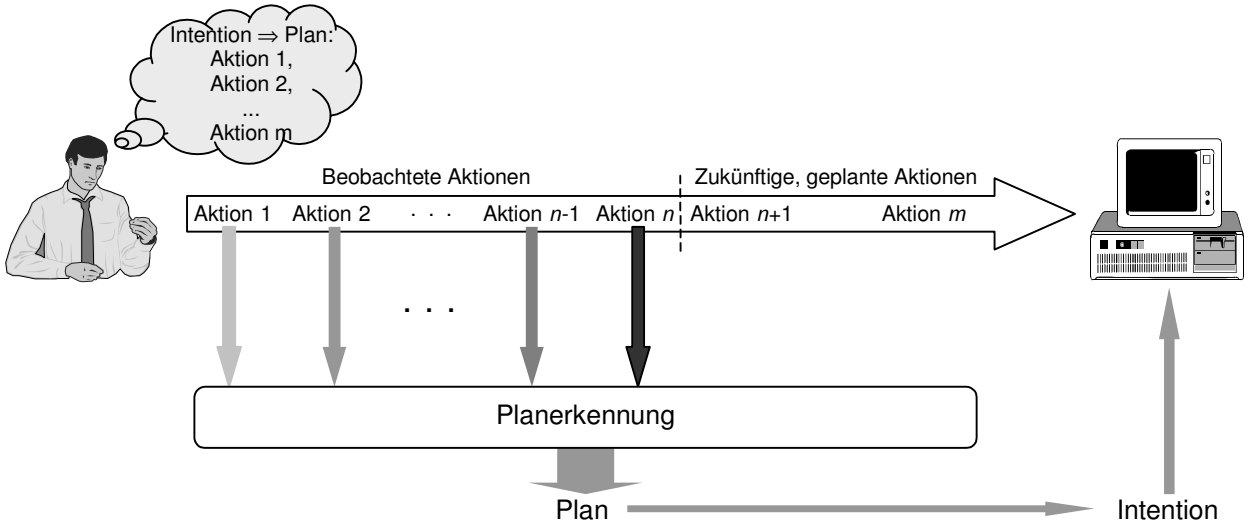


Abbildung 4.1: Grundidee der Planerkennung

Die Motivation bei der Umsetzung des intentionsbasierten Ansatzes für die Planerkennung war die Entwicklung eines weitestgehend applikationsunabhängigen Verfahrens, um mit gewissen Einschränkungen eine Portierung des Verfahrens auf beliebige Anwendungen zu gewährleisten. Aus diesem Grund wurde der Planerkenner für eine Umgebung entwickelt, die zum einen Pläne unterschiedlichster Komplexität und zum anderen nahezu beliebig viele Lösungswege für ein Ziel erlaubt. Diesen Aspekten wurde eine Unix-Shell mit einer festgelegten Verzeichnisstruktur und einem definiertem Befehlsvokabular gerecht.

Um zunächst einen allgemeinen Eindruck typischen Benutzerverhaltens in der Unix-Umgebung zu erhalten, wurde eine Voruntersuchung durchgeführt, in deren Rahmen zwölf Versuchspersonen mit unterschiedlichsten Unix-Erfahrungen innerhalb einer definierten Verzeichnisstruktur Aufgaben zu erledigen hatten. Abbildung 4.2 zeigt die Aktionssequenzen bzw. die Pläne zweier Versuchspersonen (VP1 und VP2) für identische Aufgabenstellungen. Zum besseren Vergleich der beiden listenartig dargestellten Pläne wurden in der mittleren Spalte verschiedene Einzelaktionen zu Teilplänen zusammengefasst und interpretiert. Zunächst ist auffällig, dass VP2 für das gleiche Ergebnis mehr als doppelt so viele Eingaben benötigt wie VP1. VP1 scheint sowohl mit Unix als auch mit der Verzeichnisstruktur vertraut zu sein, da sie die betreffenden Dateien zielsicher findet und die richtigen Unix-Kommandos problemlos anwendet. VP2 hat sowohl in der Kenntnis der Verzeichnisstruktur als auch in der Handhabung verschiedener Befehle Defizite. Zudem wurde die Aufgabe nicht vollständig gelöst, da die Datei *belief.c* nicht als relevant für die Lösung der Aufgabe erkannt worden war.

Bei den dargestellten Beispielen handelt es sich um zwei Extremfälle. Die Aktionssequenzen der übrigen Versuchspersonen liegen zwischen der Zielsicherheit von VP1 und der geringen Effizienz der Handlungen von VP2. Alle Pläne lassen sich in der in Abbildung 4.2 gezeigten Form interpretieren, was darauf schließen lässt, dass Benutzer, trotz zum Teil erheblich unterschiedlicher Aktionssequenzen, bei der Lösung einer Unix-Aufgabe ähnliche Strategien verfolgen. Der Vergleich der Testpläne zeigt in vielen Fällen Gemeinsamkeiten in der Wahl der Teilpläne, wobei sich diese in den Aktionsfolgen sehr deutlich unterscheiden können.

Aufgabe: **„Entpacken Sie alle komprimierten Dateien der Verzeichnisbäume *Code* und *Texte*, die älter sind als ein Jahr, drucken und löschen Sie diese Dateien“**

Versuchsperson 1 (VP1)	Interpretation	Versuchsperson 2 (VP2)
ls –lp Code/C-Code cd Code/C-Code	Gehe in das Verzeichnis C-Code	ls cd Code ls cd C-Code
	Informationen über C-Code	ls –al
gunzip belief.c.Z lp belief.c rm belief.c	Entpacke, drucke und lösche belief.c	
cd ../Tcl-Code	Gehe in das Verzeichnis Tcl-Code	cd .. cd Tcl-Code
ls	Informationen über Tcl-Code	ls –al
gunzip vis.tcl.gz visbay.tcl.Z lp vis.tcl visbay.tcl rm vis.tcl visbay.tcl	Entpacke, drucke und lösche vis.tcl.gz und visbay.tclZ	compress vis.tcl.gz unzip vis.tcl.gz man gunzip gunzip vis.tcl.gz lpstat –t lpr vis.tcl rm vis.tcl gunzip visbay.tcl.Z lpr visbay.tcl rm visbay.tcl
cd ls –lp Texte/Dokumente	Informationen über Dokumente	cd .. ls- al cd Tcl-Code ls –al cd .. ls –al cd .. ls –al cd Texte ls –al cd Dokumente ls -al
ls –lp Texte/Dokumente	Informationen über Papers	cd .. ls –al cd Papers ls –al
gunzip Texte/Papers/98mue1.ps.gz lp Texte/Papers/98mue1.ps rm Texte/Papares/98mue1.ps	Entpacke, drucke und lösche 98mue1.ps.gz	gunzip 98mue1.ps.gz lpr 98mue1.ps cd .. ls -al

Abbildung 4.2: Pläne zweier Versuchspersonen für eine bestimmte Aufgabenstellung

In diesem Zusammenhang ist insbesondere der Vergleich beider Pläne mit der optimalen bzw. kürzesten Lösung einer Unix-Aufgabe interessant. Im Durchschnitt entsprechen nur 38,6 Prozent der Aktionen eines Plans den optimalen Befehlen, d.h. der effizientesten Lösung. Somit bestehen Pläne aus ca. 2,6-mal mehr Benutzereingaben als nötig. Dieser Überschuss an Aktionen ist auf folgende Aspekte zurückzuführen:

- Häufig werden falsche, aber mit dem korrekten Befehl verwandte Kommandos angewandt
- Anstatt des effizientesten Befehls werden oft synonyme Aktionssequenzen eingegeben
- Benutzer holen häufig Informationen über die Syntax von Kommandos, die Verzeichnisstruktur und über Dateien ein

- Tippfehler, vor allem bei Verzeichnis- und Dateinamen

Diese Einflüsse wirken sich so stark auf die Pläne aus, dass dieses Phänomen im Planerkennungsprozess berücksichtigt werden sollte.

Nach Analyse der aus der Voruntersuchung resultierenden Pläne ergeben sich folgende Anforderungen an einen Planerkenner für komplexe Domänen:

- Benutzer verhalten sich sehr häufig suboptimal und benötigen, gemessen an der effizientesten Lösung, unnötige Teilpläne und Aktionen. Diese Verhaltensmuster erweisen sich dennoch als charakteristisch für bestimmte Ziele und sollten für eine benutzeradäquate Planerkennung herangezogen werden.

- Der Planerkenner muss dem breiten Spektrum an potenziellen Plänen für das Erreichen eines Ziels Rechnung tragen und deshalb unterschiedlichste Lösungsstrategien modellieren.

- Da ein Benutzer mehrere Ziele gleichzeitig verfolgen könnte, und somit diese Pläne ineinander verzahnt sein könnten, müssen alle Planhypothesen parallel betrachtet werden.

Die Grundidee des hier vorgestellten Planerkenners besteht in der Auswertung möglichst aller beobachteten Benutzeraktionen, d.h. sowohl optimaler als auch nicht optimaler Eingaben. Dies gewährleistet eine benutzeradäquate Planerkennung, da selbst typische Bedienungsfehler zur Klassifikation der Intention herangezogen werden. Somit können schon frühzeitig Aussagen über die Intention des Benutzers getroffen werden.

Unter Unix sind Pläne sehr stark von der aktuellen Verzeichnis- und Dateistruktur bestimmt. Hat eine Intentionshypothese zum Beispiel zum Ziel, alle Dateien eines bestimmten Verzeichnisses mit einem bestimmten Format zu drucken, so hängt die Anzahl der relevanten Dateien von der Verzeichnisstruktur und den Dateiattributen ab. Durch Anwendung eines Unix-Befehls können aber schnell andere Dateien im Sinne der Intentionshypothese relevant werden. Für ein statistisches Verfahren stellen verändernde Randbedingungen ein Problem dar, da die Verzeichnis- und Dateistruktur, auf deren Basis die bedingten Wahrscheinlichkeiten berechnet wurden, eventuell in dieser Form gar nicht mehr existiert. Zudem bereiten neue Intentionshypothesen das Problem, dass neue Pläne bzw. Trainingsdaten generiert werden müssten. Vor allem für die Berücksichtigung von nicht optimalen Plänen sind neue Daten notwendig. Daraus ergibt sich eine weitere Anforderung an den hier vorgestellten Planerkenner: Die Intentionsmodelle sollten aus Modulen bestehen, die so zusammengesetzt werden können, dass sie einfach neuen Verzeichnis- und Dateistrukturen angepasst werden können. Darüber hinaus soll eine Erweiterung der Intentionsbibliothek ohne erneute Akquisition von Trainingsdaten möglich sein.

Als grundlegender Ansatz zur benutzeradäquaten Planerkennung ist eine intentionsbasierte Interpretation der Aktionssequenzen ideal geeignet, da die Intentionsmodelle mit ihrer Möglichkeit der Merkmalsselektion und der unterschiedlichen Gewichtung von Merkmalsbeobachtungen alle Voraussetzungen erfüllen. Die Fähigkeit der Bayes'schen Netze, unvollständige und unscharfe Informationen zu verarbeiten, kann direkt für die Klassifikation unvollständiger Aktionssequenzen genutzt werden. Zudem können die Intentionshypothesen quantitativ evaluiert werden, was für das auf die

Ergebnisse der Planerkennung zugreifende Dialogmodell in der Regel von großer Relevanz ist. Die Realisierung des Planerkenners orientiert sich an der in Kapitel 2 vorgestellten ersten Ausprägung des intentionsbasierten Ansatzes.

In den folgenden Abschnitten wird die Umsetzung des intentionsbasierten Ansatzes für eine benutzeradäquate Planerkennung mit dem Namen *AMPlan* (*A*daptive *M*ensch-Maschine-Interaktion mittels *Plan*erkennung) vorgestellt. Dabei handelt es sich um eine maßgebliche Weiterentwicklung des in [Hof01b] vorgestellten Planerkenners.

4.2 Stand der Technik

Bei der Planerkennung wird zwischen der *keyhole-recognition* und der *intended-recognition* unterschieden [Coh81]. Die erste Art Planerkenner analysiert die Aktionen eines Benutzers, ohne dass sich dieser einer Beobachtung bewusst ist. Bei der zweiten Art weiß der Benutzer von der Planerkennung und handelt kooperativ. Da es sich bei dem hier vorgestellten System um einen *keyhole*-Erkenner handelt, wird nur auf Arbeiten zu diesem Thema eingegangen.

Zu Beginn der Forschungsaktivitäten auf dem Gebiet der Planerkennung wurden zunächst verstärkt regelbasierte Ansätze untersucht [Kau86]. Charniak und Goldman [Cha93] untersuchten erstmals Bayes'sche Netze für die Planerkennung. Dabei handelte es sich allerdings nicht um einen Ansatz zur Ermittlung der Benutzerintention, sondern um ein Verfahren zum Verstehen von Geschichten. Hierfür wurde ein Regelsatz entworfen, der abhängig von beobachteten Sätzen die Netzstruktur aufbaut und verändert und somit Aussagen über den vermittelten Inhalt zulässt.

Huber [Hub93][Hub94] entwickelte ebenfalls einen Formalismus zur dynamischen Generierung eines Bayes'schen Netzes zur Planerkennung. Dieses Verfahren sieht ein Netz für alle Planhypothesen vor und unterscheidet sich deshalb grundlegend von dem hier vorgestellten System. Darauf aufbauend entwickelte Pynadath [Pyn95][Pyn99] einen Ansatz zur Vorhersage von Fahrmanövern in Verkehrsituationen. Albrecht [Alb97] nutzte Bayes'sche Netze, um Benutzeraktionen im einem Computerspiel abzuschätzen.

Ein von Lesh [Les99] entwickeltes Verfahren ist in der Lage, Unix-Pläne zu klassifizieren. Da es sich aber um einen graphenbasierten Ansatz handelt, der einen bestimmten Regelsatz vorsieht, um mit jeder neuen Beobachtung Planhypothesen ausschließen zu können, ist dieser Algorithmus nicht mit *AMPlan* vergleichbar.

Das hier vorgestellte System *AMPlan* zeichnet sich dadurch aus, dass es in der Lage ist, neben optimalen Aktionssequenzen auch nicht optimales Benutzerverhalten konsequent für die Planerkennung heranzuziehen. Dass diese Fähigkeit für eine benutzeradäquate Planerkennung unumgänglich ist, zeigten die bereits diskutierten Voruntersuchungen, da die Versuchspersonen im Durchschnitt 2,6-mal mehr Befehle eingaben als unbedingt nötig. Hinweise auf Planerkenner, die suboptimales Agieren derart konsequent in den Planerkennungsprozess mit einbeziehen, wurden in der Literatur nicht gefunden, sodass *AMPlan* einen innovativen Beitrag zu diesem Forschungsthema leistet.

4.3 Systemarchitektur

Die Architektur des intentionsbasierten Ansatzes wurde für die Verwendung zur Planerkennung in die in Abbildung 4.3 dargestellte Form gebracht. Die Intentionsbibliothek umfasst alle von der Planerkennung in Betracht zu ziehenden Intentionshypothesen. Für jedes mögliche Ziel wird ein Intentionsmodell auf Basis eines Bayes'schen Netzes geschaffen, um Beobachtungen von Benutzeraktionen mit dieser Intentionshypothese in Bezug setzen zu können.

Das Benutzerverhalten wird durch die Beobachtungsfolge **o** dokumentiert. In dem hier vorgestellten System handelt es sich dabei um einzelne Aktionen; die Planerkennung selbst wertet allerdings Aktionssequenzen aus, indem vorhergehende Beobachtungen nicht verworfen werden.

Jede Benutzeraktion wird im Rahmen der Merkmalsextraktion in die für die Planerkennung interessanten Aspekte zerlegt. Diese sind neben dem eingegebenen Kommando die Optionen, die Parameter sowie das aktuelle Verzeichnis.

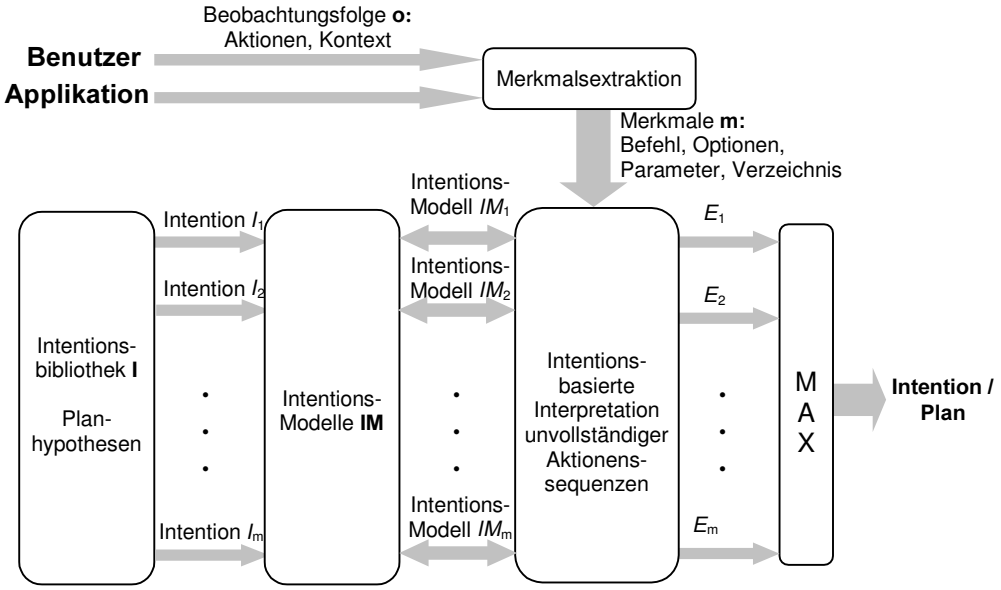


Abbildung 4.3: Systemarchitektur des intentionsbasierten Ansatzes für die Planerkennung

Bei der Komponente zur intentionsbasierten Interpretation von Aktionssequenzen handelt es sich um den eigentlichen Prozess der Planerkennung. Gestartet wird dieser Prozess bei jeder neu beobachteten Benutzeraktion, wobei die vorhergehenden Beobachtungsfolgen mit einbezogen werden. Diese Informationen dienen als Grundlage für die Bewertung aller Intentionshypothesen, indem sie auf die entsprechenden Intentionsmodelle abgebildet und mit einem quantitativen Evaluierungsmaß bewertet werden. Die Intention mit dem maximalen Evaluierungswert stellt schließlich das Ergebnis der Planerkennung dar. Entscheidend ist, dass es sich bei diesem Ergebnis nur um eine Momentaufnahme handelt, die bei einer neu eingegebenen Aktion mittels eines neuen Planerkennungsprozesses wieder zu überdenken ist.

4.4 Intentionsbibliothek

Eine Betrachtung der Intentionshypothesen ist in der Unix-Domäne nur in Hinblick auf die zugrunde liegende Datei- und Verzeichnisstruktur sinnvoll. Im Rahmen von *AMPlan* wurde die in Abbildung 4.4 dargestellte Verzeichnis- und Dateistruktur als Ausgangssituation verwendet.

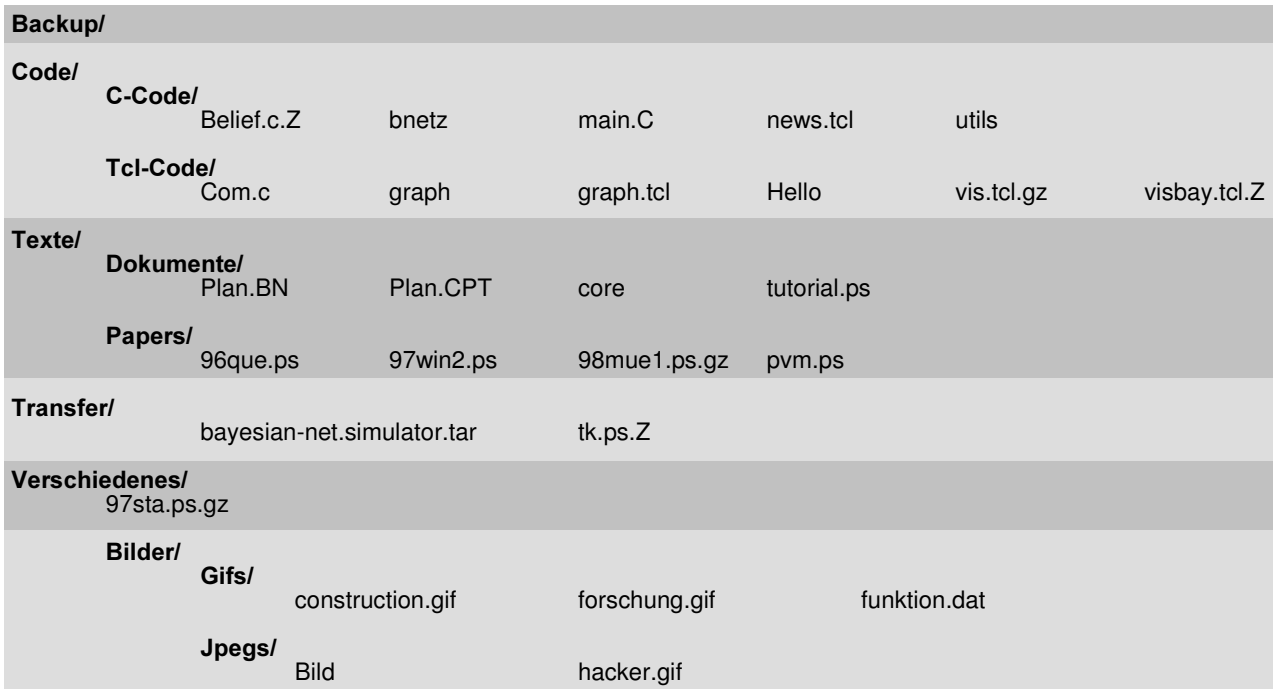


Abbildung 4.4: Verzeichnis- und Dateistruktur für *AMPlan*

Jede Datei ist durch die Attribute Verzeichnis, Dateigröße, Zeitpunkt der letzten Änderung und Format charakterisiert. Die dargestellten Dateien und Verzeichnisse können durch eine Reihe von Standard-Unix-Befehlen beliebig angesprochen und manipuliert werden.

I_1: „Komprimiere alle Dateien (>10k) des Unterverzeichnisse *C-Code* und kopiere diese in das Verzeichnis *Backup*“

I_2: „Erzeuge das Verzeichnis *Zip* im Verzeichnis *Backup* und verschiebe alle komprimierten Dateien in das neu angelegte Verzeichnis“

I_3: „Lösche die ASCII-Datei, die das Wort *output* enthält“

I_4: „Dekomprimiere alle komprimierten Dateien des Verzeichnisses *Papers*, drucke und komprimiere die Postscript-Dateien unter diesen Dateien, erzeuge das Verzeichnis *Verschiedenes/Archiv* und verschiebe diese Dateien in das neu angelegt Verzeichnis“

I_5: „Drucke und komprimiere alle Dateien des Verzeichnisses *C-Code*, die größer als 5k sind“

I_6: „Komprimiere alle Dateien des Verzeichnisses *Papers*“

I_7: „Finde eine ASCII-Datei, die weniger als 100 Wörter beinhaltet“

I_8: „Erzeuge das Verzeichnis *Transfer/Software* und verschiebe alle Dateien des Verzeichnisses *Transfer* in das neue Verzeichnis“

I_9: „Drucke alle ASCII-Dateien des Verzeichnisses *Dokumente*, die nicht größer als 5k sind“

I_{10}: „Komprimiere alle Dateien (>100k), die seit mindestens einem Monat nicht geändert wurden“

Abbildung 4.5: Intentionsbibliothek von *AMPlan*

Als Überbegriff für Dateien und Verzeichnisse wird im Folgenden die Bezeichnung *Objekt* verwendet. Auf Kommandosequenzen, die inhaltlich einer Art Teilplan entsprechen, weil sie in ihrer Abfolge ein Teilziel erfüllen, wird als *Operator* Bezug genommen.

In der dargestellten Datei- und Verzeichnisstruktur sind durch Kombination von Operatoren nahezu unendlich viele Intentionen bzw. Plänen denkbar. Für die Intentionsbibliothek wurden zehn realistische Ziele ausgewählt, die zum Teil erhebliche Schnittmengen aufweisen. Abbildung 4.5 zeigt die Intentionsbibliothek von *AMPlan*.

4.5 Merkmalsextraktion

Die Merkmalsextraktion zerlegt die Beobachtungsfolge in alle für die Klassifikation relevanten Informationen (Abbildung 4.6). Die Beobachtungsfolge besteht zum einen aus der zuletzt getätigten Benutzereingabe sowie aus Kontextinformationen, bei denen es sich im Fall von *AMPlan* um das aktuelle Verzeichnis sowie um die Syntax des eingegebenen Befehls handelt. Zunächst wird anhand der Kommandoanalyse der eingegebene Unix-Befehl identifiziert. Mit dieser Information und dem Wissen typischer Parameterkonstellationen des erkannten Befehls werden die Parameter analysiert und in Optionen, allgemeine Parameter und Objekte zerlegt. In dem in Abbildung 4.6 dargestellten Beispiel handelt es sich um eine Option und zwei Objekte.

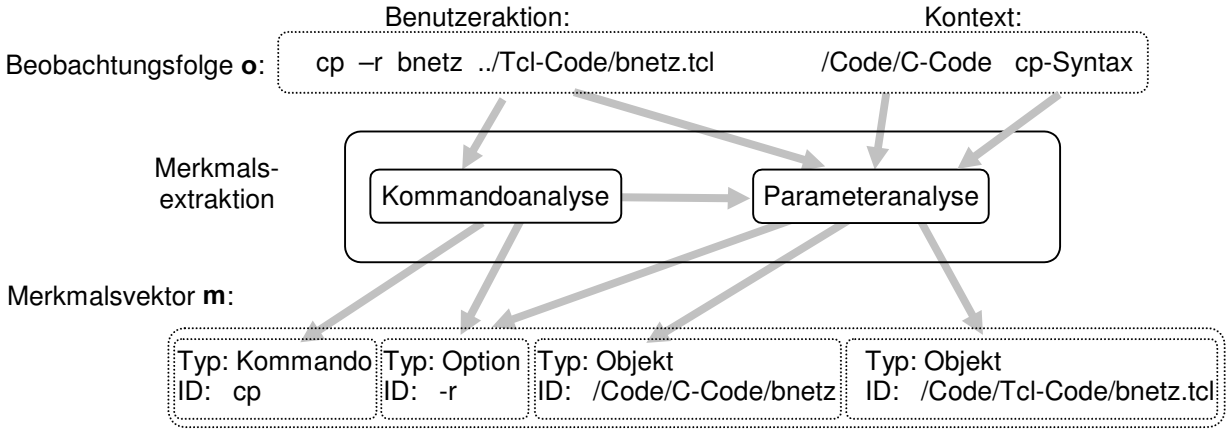


Abbildung 4.6: Analyse der Beobachtungsfolge durch die Merkmalsextraktion

Objekte müssen eindeutig identifizierbar definiert sein, um Fehlzuweisungen von Merkmalsbeobachtungen und von in den Intentionsmodellen repräsentierten Merkmalen auszuschließen. Hierfür werden die Objekte mit ihren absoluten Pfaden definiert, indem die Parameter eines eingegebenen Kommandos mit dem Wissen über Verzeichnisstruktur und Befehlssyntax analysiert werden.

Die Charakterisierung einer Aktion anhand mehrerer Merkmale ermöglicht eine effiziente Modellierung von Tippfehlern, da stets nur die Merkmale einer Aktion von Belang sind, die im Intentionsmodell berücksichtigt sind. Somit werden einfach die von Tippfehlern betroffenen Komponenten ignoriert, während die korrekten Merkmale derselben Aktion als Eingangsdaten für die Planerkennung dienen. Die ermöglicht die Merkmalsselektion des intentionsbasierten Ansatzes.

Beschrieben werden die Merkmale einer Aktion anhand des Typs und eines Identifikationsstrings:

- **Merkmalstyp**
  Es gibt fünf verschiedene Merkmalstypen: Kommando, Option, Objekt, Information und das aktuelle Verzeichnis. Je nach Merkmalstyp werden die Merkmale unterschiedlich interpretiert.
- **Identifikationsstring (ID)**
  Der Erkennungsstring dient der eindeutigen Beschreibung von Kommandos, Objekten und Optionen. Die Objektnamen werden durch den absoluten Pfad ergänzt, um eine eindeutige Identifikation der Merkmale zu gewährleisten.

Ein Merkmal besteht ausschließlich aus dem Merkmalstyp und einem ID-String. Je nach Eingabe kann eine Benutzeraktion durch eine unterschiedliche Anzahl von Einzelmerkmalen beschrieben werden. Diese werden dann durch den Merkmalsvektor **m** zusammengefasst.

4.6 Intentionsmodelle

Wie die in Kapitel 4.1 diskutierten Voruntersuchungen ergaben, gibt es unter Unix unzählige Möglichkeiten für die Umsetzung eines Ziels. Deshalb müssen die Intentionsmodelle von *AMPlan* eine möglichst große Bandbreite an potenziellen Aktionssequenzen abdecken. Darüber hinaus wird die Planerkennungs-Logik nahezu komplett durch die Struktur der Bayes'schen Netze realisiert.

4.6.1 Struktur der Intentionsmodelle

Aufgabe eines Intentionsmodells ist die Repräsentation aller für diese Intention charakteristischen Aktionen sowie deren syntaktisch-semantische Beziehungen zueinander. Für die grundlegende Struktur eines Intentionsmodells wird ein typischer Plan als Reihe von Operatoren definiert, die auf die für diese Intention relevanten Objekte angewendet werden. Gleichung (4-1) stellt dies dar:

$$Plan = \textbf{Operator}(\textbf{Objekt}) \tag{4-1}$$

Diese Interpretation eines Plans bestimmt die Struktur eines *AMPlan*-Intentionsmodells. Abbildung 4.7 gibt einen Überblick über die Struktur.

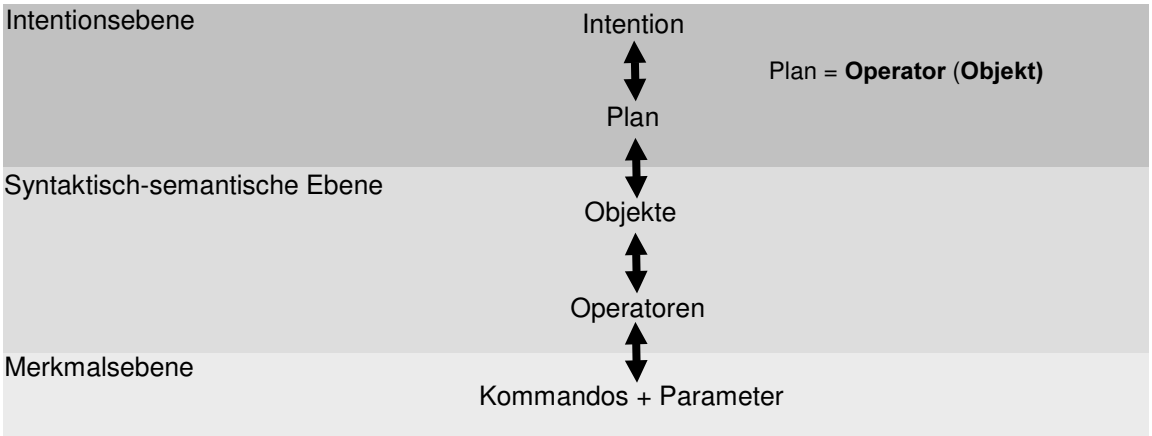


Abbildung 4.7: Allgemeine Struktur eines *AMPlan*-Intentionsmodells

Analog zu Abbildung 2.5 erfolgt die Beschreibung eines *AMPlan*-Intentionsmodells anhand von drei Ebenen:

- **Intentionsebene**
  Für die erfolgreiche Umsetzung einer Intention ist eine Abfolge von Benutzeraktionen, ein Plan, notwendig. Da das Erkennen eines Plans äquivalent mit der Klassifikation der Intention ist, sind aus Sicht des Verfahrens die Begriffe *Intention* und *Plan* synonym. Wahrscheinlichkeitsbasierte Aussagen über Planhypothesen sind identisch mit Aussagen über Intentionshypothesen.

- **Syntaktisch-semantische Ebene**
  Ein Plan sieht die Manipulation von Objekten durch Anwendung von Operatoren vor. Auf der syntaktisch-semantischen Ebene wird zunächst festgelegt, welche Objekte für die Intentionshypothese von Belang sind. Für jedes dieser Objekte wird definiert, welche Operatoren darauf anzuwenden sind. Darüber hinaus werden noch die einzelnen Operatoren derart beschrieben, dass sie sowohl optimale als auch suboptimale Benutzeraktionen modellieren. Für die Operatoren wird hierfür auf Operator-Subnetze zurückgegriffen, die eine flexible Anpassung eines Intentionsmodells an unterschiedliche Verzeichnisstrukturen ermöglichen. Diese Operator-Module werden schließlich durch Merkmalsknoten realisiert.

- **Merkmalsebene**
  Die Merkmalsknoten der Operator-Module modellieren die Beobachtungen von Unix-Kommandos, Optionen, Objekten, eingeholter Informationen und Verzeichnissen. Über diese Knoten greift die intentionsbasierte Interpretation auf das Intentionsmodell zu.

Im nächsten Abschnitt wird die konkrete Umsetzung eines *AMPlan*-Intentionsmodells auf Basis eines Bayes'schen Netzes vorgestellt.

4.6.2 Realisierung der Intentionsmodelle

Für jede Intentionshypothese wird ein Intentionsmodell in Form eines Bayes'schen Netzes bereitgestellt. Da die Generierung eines Intentionsmodells für beliebige Intentionshypothesen möglich sein soll, wurden allgemeingültige Regeln für die Ermittlung der Netzwerktopologie entwickelt.

Der erste Schritt besteht darin, die im Sinne der Intentionshypothese relevanten Objekte und Operatoren zu ermitteln. Abbildung 4.8 zeigt dies schematisch sowie anhand eines konkreten Beispiels. Zunächst werden die Attribute der Objekte ermittelt, auf die sich die Intentionshypothese bezieht. Mit dem zusätzlichen Wissen über die Verzeichnisstruktur und über die Attribute aller Objekte des Dateisystems können auf diese Weise die Objekte bestimmt werden, die für die Intentionshypothese von Belang sind. Die relevanten Operatoren können der Intentionshypothese direkt entnommen werden.

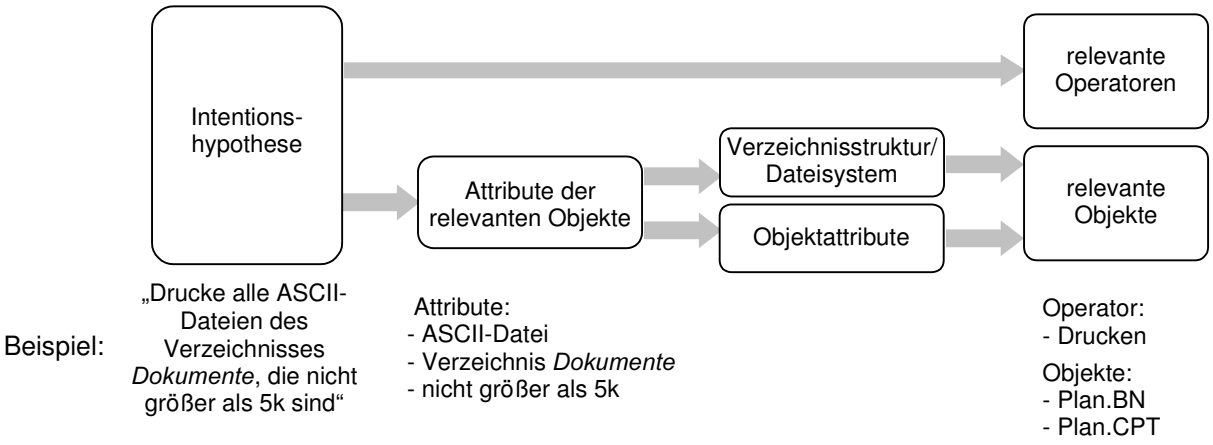


Abbildung 4.8: Der erste Schritt bei der Realisierung eines Intentionsmodells besteht aus der Ermittlung der relevanten Objekte und Operatoren

Nachdem alle Voraussetzungen für die Strukturierung des Bayes'schen Netzes erfüllt sind, werden die Objekte im Netz so dargestellt, dass jedes dieser Objekte in einer bestimmten Art und Weise zu manipulieren ist, um den Plan als vollständig beobachtet zu werten. Um dies qualitativ und quantitativ zu modellieren, ist für die Intentionsebene die in Abbildung 4.9 für drei Objekte dargestellte Topologie eines Bayes'schen Netzes vorgesehen. Alle Zustandsvariablen haben dabei einen Boole'schen Zustandsraum.

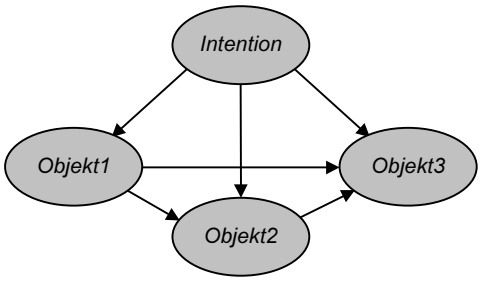


Abbildung 4.9: Ausschnitt eines Intentionsmodells für die Manipulation von drei Objekten

Jedes zu manipulierende Objekt wird durch einen Knoten (*Objektknoten*) repräsentiert, der mit dem Wurzelknoten, dem so genannten *Intentionsknoten,* mit einer Kante verbunden wird. Der Intentionsknoten spielt für die Evaluierung von Aktionssequenzen die zentrale Rolle, da der Evaluierungswert einer Intentionshypothese direkt daraus abgeleitet werden kann.

Da ein Plan nur dann vollständig beobachtet wurde, wenn seine Objekte korrekt manipuliert wurden, werden die bedingten Wahrscheinlichkeiten der Objektknoten so gewählt, dass sie folgender logischen UND-Beziehung genügen:

$$Intention = Objekt1 \wedge Objekt2 \wedge ... \tag{4-2}$$

Nachdem die geforderten Objekte im Intentionsmodell berücksichtigt sind, wird festgelegt, welche Operatoren auf diese Objekte anzuwenden sind. Hierfür ist für jeden Objektknoten ein Teilnetz nötig, das sich in Topologie und Wahrscheinlichkeiten an dem in Abbildung 4.9 dargestellten Netz-

ausschnitt orientiert. Abbildung 4.10 stellt einen derartigen Netzausschnitt für die Manipulation eines Objekts durch zwei Operatoren dar.

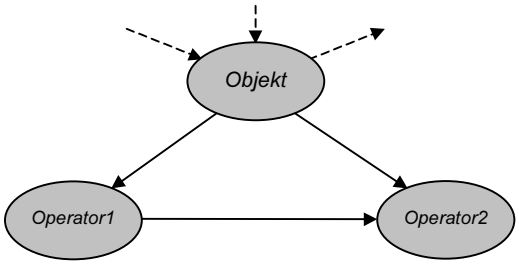


Abbildung 4.10 : Modellierung einer Manipulation eines Objektes anhand zweier Operatoren

Die bedingte Wahrscheinlichkeit des Objektknotens wurde bereits festgelegt, da es sich bei dieser Zustandsvariable um einen der in Abbildung 4.9 dargestellten Objektknoten handelt. Die bedingten Wahrscheinlichkeiten folgen wieder einer UND-Modellierung:

$$Objekt = Operator1 \wedge Operator2 \wedge ... \tag{4-3}$$

Die *Operatorknoten* stellen Aktionssequenzen dar, die für die Intentionshypothese notwendig bzw. charakteristisch sind. Um die Operatoren nun anhand der in Kapitel 4.5 vorgestellten Merkmale zu repräsentieren, werden Operator-Subnetze verwendet, die auf einfache Weise in das bisherige Intentionsmodell integriert werden können, da die Operatorknoten aus Abbildung 4.10 die Wurzelknoten der Operator-Subnetze bilden.

Für die Generierung der Operator-Subnetze wurden 12 Versuchspersonen gebeten, innerhalb einer definierten Datei- und Verzeichnisstruktur jeweils 20 Unix-Aufgaben zu lösen. Aus den resultierenden 240 Plänen wurden alle für eine korrekte Lösung der Aufgaben im Frage kommenden Operatoren ermittelt. Für jeden Operator wurde auf Basis der Testdaten ein Bayes'sches Netz geschaffen, das den optimalen sowie fehlerhaften, aber dennoch charakteristischen Aktionssequenzen zum Umsetzen des entsprechenden Teilplans Rechnung trägt. Insgesamt wurden 12 Operator-Subnetze erstellt. Abbildung 4.11 fasst dies zusammen.

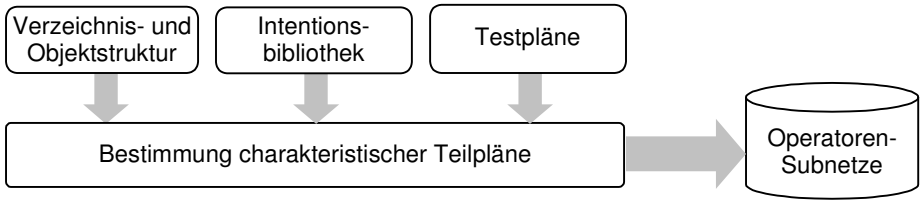


Abbildung 4.11: Generierung von Operator-Subnetzen zur Modellierung charakteristischer Teilpläne

Neben den optimalen Aktionen auch suboptimales Verhalten für die Klassifikation der Intention heranziehen zu können, erfordert eine spezifische Struktur der Operatornetze. Abbildung 4.12 gibt einen groben Überblick. Zunächst werden nur das Kommando und die Parameter bzw. Objekte modelliert, die zu einer perfekten Umsetzung des Operators führen. Die Analyse der aus der Voruntersuchung resultierenden Pläne lassen den Schluss zu, dass ein typischer Unix-Benutzer neben den optimalen Befehlen weitere Aktionen benötigt, um Informationen für die nächsten Eingaben zu sammeln. Derartige Aktionssequenzen sprechen für mangelndes Wissen über Kommandos und Ob-

jekte. Abbildung 4.12 teilt diese Aspekte in zwei Gruppen auf: Aktionen, die darauf zurückzuführen sind, dass der Benutzer das optimalen Kommando oder dessen Syntax nicht kennt, und Aktionen, die darauf zurückzuführen sind, dass der Benutzer die zu manipulierenden Objekte nicht kennt.

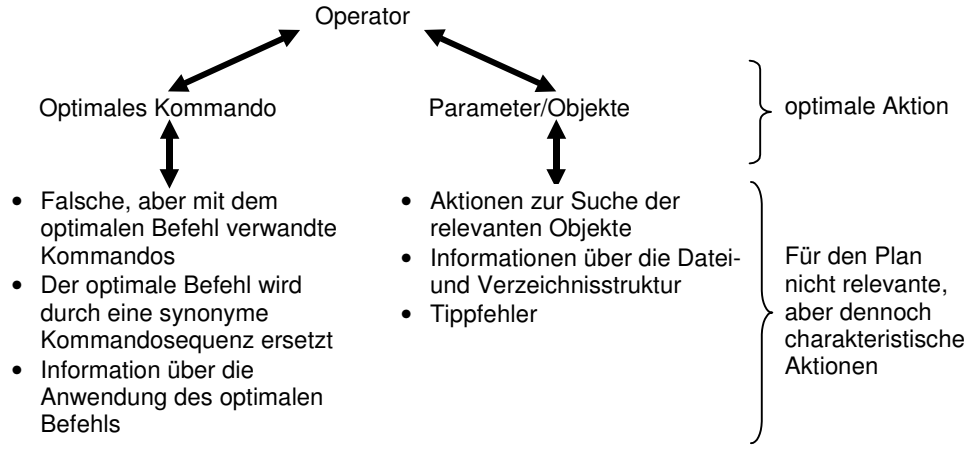


Abbildung 4.12: Grobe Struktur eines Operator-Subnetzes und Wirkung nicht relevanter, aber charakteristischer Aktionen

Die Topologie eines Operator-Subnetzes orientiert sich an der in Abbildung 4.12 dargestellten Struktur. Die Beobachtung eines Operators bedingt die Eingabe des im Sinne des Operators optimalen Unix-Kommandos sowie der relevanten Objekte. Auch darüber hinaus beobachtete, suboptimale Aktionen lassen einen Rückschluss auf die optimale Aktion und damit auf den Operator zu. Die optimale Aktion nimmt somit die Rolle eines Vermittlers zwischen dem Operator und den nicht optimalen Aktionssequenzen ein.

Wie die Strukturen und die Wahrscheinlichkeiten der Operator-Subnetze bestimmt werden, wird nun anhand eines konkreten Beispiels vorgestellt. Abbildung 4.13 zeigt ein Subnetz zur Modellierung des Operators *Objekt verschieben* anhand konkreter Objekte.

Alle Knoten dieses Netzes sind Darstellungen Boole'scher Zustandsvariablen. Bei dem Wurzelknoten handelt es sich um den Operatorknoten, der die Schnittstelle zwischen dem bisherigen Intentionsmodell und dem Operator-Subnetz bildet, d.h., die Operatorknoten des Intentionsmodells werden durch jeweils ein komplettes Operator-Subnetz erweitert. Die übrigen Knoten sind Merkmalsknoten, denen jeweils der Merkmalstyp und ein Identifikationsstring ID zugeordnet sind. Darüber sind Bedingungen an die Knoten geknüpft, um die durch die Netz-Topologie erzeugte Planerkennungslogik noch zu ergänzen.

Bei der intentionsbasierten Interpretation von Aktionssequenzen werden die Merkmale einer Aktion mit den Merkmalen, die im Intentionsmodell durch Merkmalsknoten modelliert werden, verglichen. Im Falle von Übereinstimmungen wird den betreffenden Merkmalsknoten der ja-Zustand zugewiesen, d.h., die Zustandsvariablen werden instanziiert. Durch die Bedingungen können Abhängigkeiten zwischen Merkmalen erzeugt werden. Die für Beschreibung der Bedingungen verwendete Syntax wird anhand zweier beliebiger Merkmalsknoten m_x und m_y kurz erläutert. Die Merkmalsknoten werden immer durch eine logische UND- oder ODER-Verknüpfungen verbunden. Ist ein Term von eckigen Klammern umschlossen, so bezieht sich dieser Termin stets auf die aktuelle Merkmalsbeobachtung, d.h., „$<m_x \vee m_y>$ bedeutet, dass ein Knoten nur dann instanziiert wird, wenn innerhalb

desselben Merkmalsvektors entweder der Knoten m_x oder m_y oder beide Knoten instanziiert werden. Ohne Klammern dürfen die Knoten bereits auf Grund vorheriger Aktionen instanziiert worden sein.

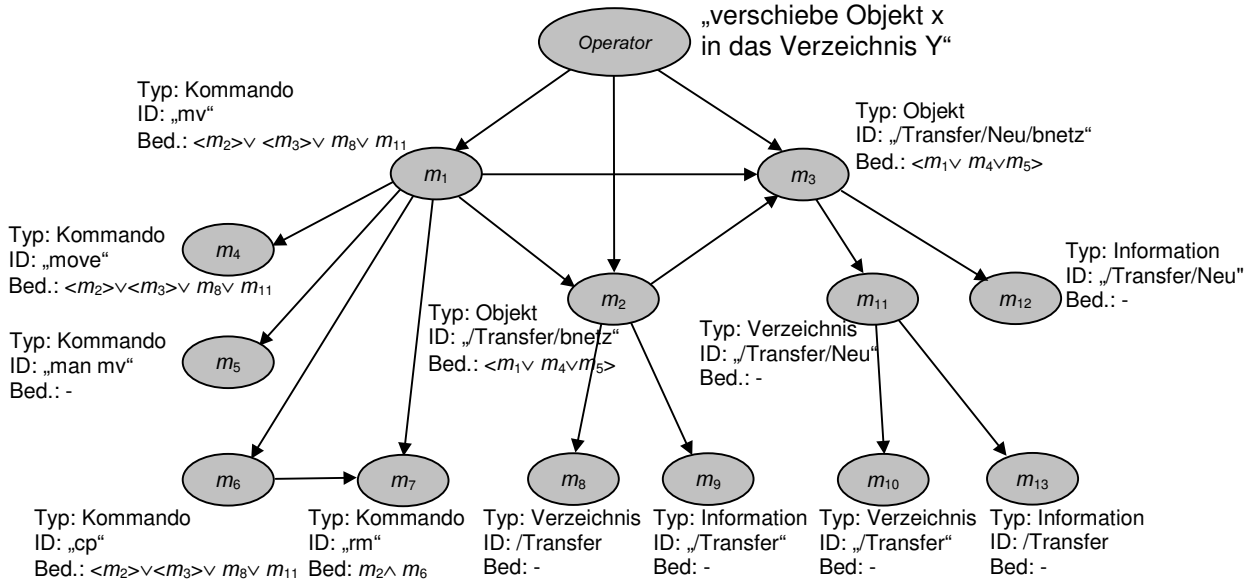


Abbildung 4.13: Subnetz zur Modellierung des Operators „verschiebe das Objekt *bnetz* des Verzeichnisses *Transfer* in das Verzeichnis *Transfer/Neu*“

Am Beispiel des Merkmalsknotens m_1 werden die Möglichkeiten, die sich aus den Bedingungen ergeben, kurz erläutert. Der logische Ausdruck „<m_2>∨ <m_3>∨ m_8∨ m_{11}“ knüpft die Instanziierung von m_1 an andere Merkmale. Bei Beobachtung des Befehls „mv“ wird m_1 nur dann instanziiert, wenn mindestens einer der Parameter (m_2 bzw. m_3) innerhalb derselben Benutzereingabe ebenfalls beobachtet wurde oder falls das aktuelle Verzeichnis einem der beiden Objektverzeichnisse entspricht (m_8 bzw. m_{11}) entspricht. Dadurch können zusätzliche syntaktisch-semantische Beziehungen zwischen verschiedenen Merkmalsknoten realisiert werden.

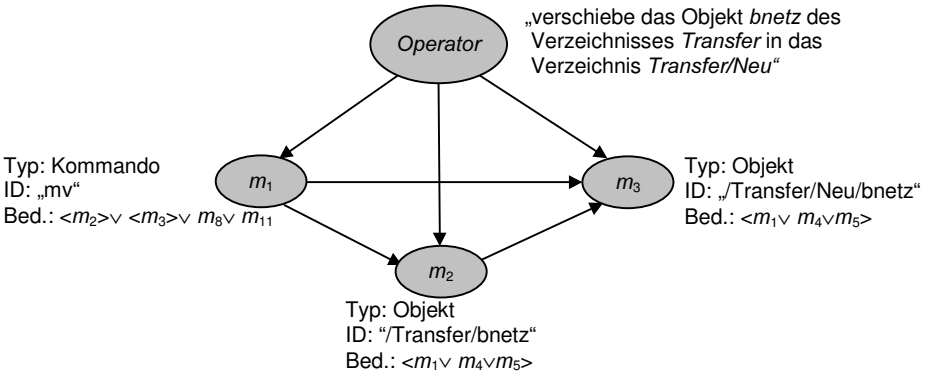


Abbildung 4.14: Operator-Netzausschnitt zur Modellierung einer optimalen Aktion

Um die Ideen hinter der Struktur und den Wahrscheinlichkeiten eines typischen Operator-Netzes verständlicher zu diskutieren, wird das Netz schrittweise aufgebaut. Wie bereits erwähnt wird zunächst die im Sinne des Operators optimale Benutzeraktion modelliert. Im dem gewählten Beispiel ist dies die Aktion „mv /Transfer/bnetz /Transfer/Neu“, deren Merkmale aus einem Unix-Kommando und zwei Parametern bestehen. Bei Beobachtung dieser drei Merkmale wird der Opera-

tor als beobachtet gewertet. Dies wird in dem in Abbildung 4.14 dargestellten Netzausschnitt modelliert.

Die bedingten Wahrscheinlichkeiten der Boole'schen Merkmalsknoten folgen einer UND-Logik. Zur eindeutigen Identifikation der Merkmale werden diese mit den diskutierten Attributen beschrieben.

Zur Modellierung charakteristischer Aktionen wird das Netz an den Merkmalsknoten um weitere Knoten und Pfeile erweitert. Wie anhand von Abbildung 4.12 diskutiert, wird dabei unterschieden, ob diese Aktionen dem Ausführen des optimalen Befehls oder dem Finden der relevanten Objekte dienen. Abbildung 4.15 stellt den Netzausschnitt dar, der die Aktionen berücksichtigt, die Rückschluss auf das optimale Kommando und damit auf den Operator zulassen.

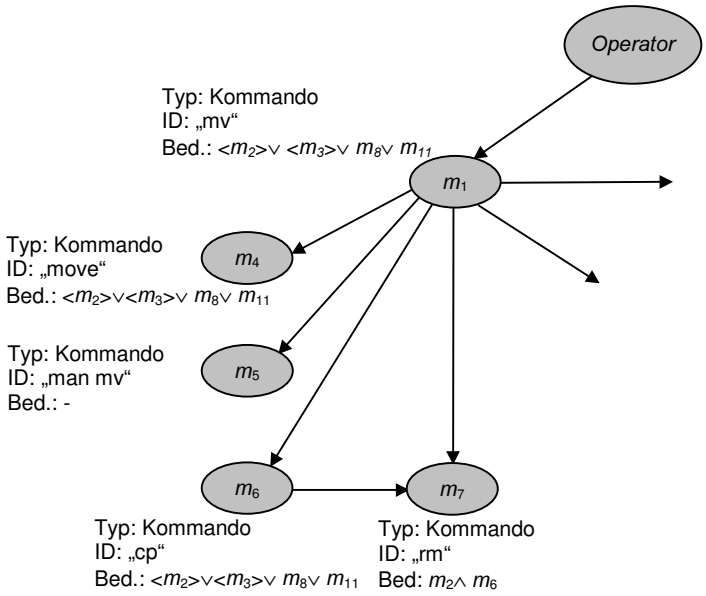


Abbildung 4.15: Operator-Netzausschnitt zur Modellierung fehlerbehafteter Aktionen

Jede Aktion wird als Kindknoten des optimalen Kommandos vernetzt. In Abbildung 4.15 trifft dies auf die Knoten m_4, m_5, m_6 und m_7 zu, die alle in Abhängigkeit von m_1 stehen. Knoten m_4 steht für die Eingabe „move“, die als Befehl nicht existiert, irrtümlich aber anstelle von „mv“ häufig eingegeben wird. Basis für die Berechnungen der bedingten Wahrscheinlichkeit dieser Knoten sind die durch die Voruntersuchung ermittelten 240 Pläne. Am Beispiel des Knotens m_4 wird gezeigt, wie dessen bedingte Wahrscheinlichkeit $P(m_4 \mid m_1)$ für den *Verschieben*-Operator bestimmt wird:

$$P(m_4 = \text{ja} \mid m_1 = \text{ja}) = \frac{n_{\text{Aktion}}}{n_{\text{Operator}}} \tag{4-4}$$

$$P(m_4 = \text{ja} \mid m_1 = \text{nein}) = \frac{n_{\text{Aktion}}}{n_{\neg\text{Operator}}} \tag{4-5}$$

n_{Aktion} ist die Häufigkeit des „move“-Befehls in den Aktionssequenzen des betrachteten *Verschieben*-Operators, n_{Operator} die Häufigkeit des *Verschieben*-Operators in den Testplänen. Bei $n_{\neg\text{Operator}}$ handelt es sich um die Zahl aller übrigen Operatoren. Die restlichen Wahrscheinlichkeitswerte kön-

nen durch Subtrahieren der Werte von Gleichung (4-4) und Gleichung (4-5) von Eins bestimmt werden.

Je charakteristischer der Befehl „move“ für den *Verschieben*-Operator ist, desto höher ist der Wert $P(m_4 = \text{ja} \mid m_1 = \text{ja})$ im Verhältnis zu $P(m_4 = \text{ja} \mid m_1 = \text{nein})$. Bei Eingabe des Befehls „move“ wird der Knoten m_4 in den ja-Zustand versetzt. Somit wirkt die Rückschlusswahrscheinlichkeit $P(m_1 \mid m_4 = \text{ja})$ auf den Knoten m_1 und stärkt die Annahme, dass der Benutzer das eigentliche Kommando „mv“ eingeben möchte. Diese Information wird schließlich durch das gesamte Bayes'sche Netz propagiert, sodass auch die Wahrscheinlichkeiten von Operator- und Intentionsknoten auf den neusten Stand gebracht werden. Die veränderte Wahrscheinlichkeit des Intentionsknotens reflektiert direkt die Annahme, dass die betrachtete Intentionshypothese der Benutzerintention gleicht.

Da nur Aktionen in ein Operator-Subnetz mit einbezogen werden sollten, die tatsächlich charakteristisch für diesen Operator sind, ist darauf zu achten, dass folgende Bedingung erfüllt ist:

$$P(m_4 = \text{ja} \mid m_1 = \text{ja}) > P(m_4 = \text{ja} \mid m_1 = \text{nein}) \tag{4-6}$$

Mit allen suboptimalen Aktionen ist analog zu verfahren, mit Ausnahme von Aktionssequenzen, die in ihrer Wirkung identisch mit optimalen Kommandos sind. In Abbildung 4.15 trifft dies auf die Merkmalsknoten m_6 und m_7 zu, die das optimale Kommando *mv* durch die synonyme Kommandosequenz *cp* und *rm* ersetzen. In diesem Fall werden die bedingten Wahrscheinlichkeiten beider Merkmalsknoten so gewählt, dass sie eine UND-Funktion realisieren.

Die Wirkung einer Merkmalsbeobachtung auf den Intentionsknoten wird dadurch ermöglicht, dass der Intentionsknoten und die Merkmalsknoten indirekt über ein Reihe anderer Knoten, wie zum Beispiel Operator- und Objektknoten, statistisch abhängig sind. Für die Integration suboptimaler Aktionen in ein Intentionsmodell wurde bewusst eine Topologie gewählt, die gezielt eine Besonderheit der Bayes'schen Netze nutzt: das Phänomen *d-separation* [Rus95] [Jen96]. Dieses Phänomen beschreibt die Auflösung statistischer Abhängigkeiten zweier Zustandsvariablen, falls die einzige Variable, über die die beiden Knoten indirekt in Verbindung stehen, instanziiert wird. Dies ermöglicht die Realisierung einer definierten Planerkennungslogik. Bezogen auf Abbildung 4.15 bedeutet dies, dass die Eingabe des Befehls „move“ keinen Einfluss auf das restliche Netz hat, wenn der Benutzer bereits vorher das Kommando „mv“ aufgerufen hat. Die Instanziierung von Knoten m_1 blockiert den Informationsfluss zwischen m_4 und den übrigen Knoten. Sobald ein Merkmal der optimalen Aktion zur Instanziierung seines Merkmalsknotens führt, sind alle suboptimalen Eingaben, die mit diesem Merkmal korreliert sind, trotz Instanziierung ihrer Knoten ohne Wirkung. Die zugrunde liegende Idee ist, dass ein Benutzer nach Erreichen eines Teilziels sich weiteren Teilzielen widmet und deshalb folgende Aktionen anderen Teilplänen zuzuordnen ist. Die Topologie des Intentionsmodells erlaubt eine analoge Logik auf Operator- und Objektebene. D.h., falls ein Objekt im Sinne der Intentionshypothese korrekt manipuliert wurde, sind nachfolgende Eingaben, die dieses Objekt bzw. diesen Teilplan betreffen, wirkungslos.

Aktionen, die Rückschluss auf das Objekt zulassen, sind in Abbildung 4.16 dargestellt. Um ein Objekt anzusprechen, muss der Benutzer das Objekt kennen und in die Situation kommen, dieses Ob-

jekt zu adressieren, d.h., diese Aktionen dienen vor allen dem Sammeln von Informationen über potenziell zu manipulierende Objekte sowie dem Aufsuchen des Objektverzeichnisses. Dies wirkt sich auch auf die Topologie des Operator-Subnetzes aus. Die Knoten m_{10} und m_{11} modellieren die Verzeichnisse */Transfer* bzw. das Zielverzeichnis */Transfer/Neu*. Ausgehend vom Objektknoten, dessen Objekt das Zielverzeichnis vorgibt, wird ein Knoten zur Modellierung des Zielverzeichnisses mit dem Objektknoten verbunden. Sobald der Benutzer das Zielverzeichnis betritt, erhöht sich die Annahme, dass er auf ein Objekt dieses Verzeichnisses zugreifen wird. Nicht nur mit dem Zielverzeichnis wird in dieser Weise verfahren, sondern mit allen Verzeichnissen, die der Benutzer beim Aufsuchen des Zielverzeichnisses betreten könnte. Dies bezieht sich vor allem auf alle dem Zielverzeichnis übergeordneten Verzeichnisse. In Abbildung 4.16 wird das Zielverzeichnis /Transfer/Neu durch den Knoten m_{11} dargestellt, der Knoten m_{10} modelliert das übergeordnete Verzeichnis */Transfer*. Im Unterschied zu dem in Abbildung 4.15 gezeigten Netzausschnitt modellieren Verzeichnisknoten keine Aktionen, sondern Verzeichnisse. Aktionen wurden im Rahmen der Testpläne in der Regel nicht rückgängig gemacht, während Verzeichniswechsel sehr häufig stattfinden und das Intentionsmodell darauf zu reagieren hat. Dies wird im Detail in Kapitel 4.7 erläutert.

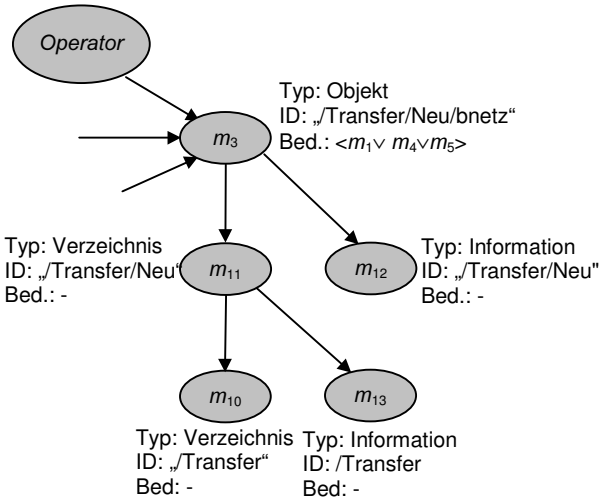


Abbildung 4.16: Operator-Netzausschnitt zur Modellierung fehlerbehafteter Aktionen

Die bedingten Wahrscheinlichkeiten der Verzeichnisknoten wurden anhand der 240 Testpläne auf Basis von Häufigkeiten bestimmt. Da Operator-Subnetze für beliebige Verzeichnisstrukturen generierbar sein sollten, wurde nur die Wahrscheinlichkeit $P(\mathit{Verzeichnis} \mid \mathit{Unterverzeichnis})$ bestimmt, die Auskunft darüber gestattet, wie wahrscheinlich das Interesse an einem Unterverzeichnis ausgehend vom aktuellen Verzeichnis ist. Diese Wahrscheinlichkeit wird in allen Operator-Subnetzen für jeden Verzeichnisknoten herangezogen. In der gleichen Weise wird die Wahrscheinlichkeit $P(\mathit{Verzeichnis} \mid \mathit{Objekt})$ bestimmt und eingesetzt. Diese Wahrscheinlichkeit gibt Auskunft darüber, wie wahrscheinlich das Benutzerinteresse am zu manipulierenden Objekt ist, wenn er sich bereits im Zielverzeichnis befindet.

Analog wird mit Aktionen verfahren, die Informationen über ein Verzeichnis und deren Dateien und Unterverzeichnisse geben sollen. Diese Aktionen werden durch Merkmalsknoten festgehalten, die direkt mit dem Objektknoten verbunden sind. Für den Merkmalstyp *Information* ist nur interessant, dass der Benutzer Informationen über ein bestimmtes Objekt einholt, und nicht, welche Befeh-

le er hierfür verwendet. Somit kann die Komplexität des Netzes erheblich reduziert werden, da die Vielzahl von Möglichkeiten, Informationen über ein Objekt einzuholen, viele Knoten erfordern würde.

Mit der Erstellung eines Intentionsmodells für jede Hypothese der Intentionsbibliothek wurde die Grundlage für die intentionsbasierte Klassifikation von Aktionssequenzen geschaffen. Diese intentionsbasierte Interpretation von Benutzeraktionen, die den eigentlichen Prozess der Planerkennung darstellt, ist Gegenstand des folgenden Abschnittes.

4.7 Intentionsbasierte Interpretation von Aktionssequenzen

Aufgabe der Planerkennung ist die Klassifikation von Aktionssequenzen mit dem Ziel, bereits zu einem frühen Zeitpunkt Aussagen über die Benutzerintention treffen zu können. Jede neu beobachtete Aktion wird auf alle Hypothesen der Intentionsbibliothek abgebildet, um anhand des quantitativen Evaluierungsmaßes die wahrscheinlichste Benutzerabsicht zu berechnen. Zunächst wird der Algorithmus zur Interpretation unvollständiger Aktionssequenzen allgemein beschrieben. Schließlich wird die Herangehensweise exemplarisch anhand des in Kapitel 4.6 entwickelten Intentionsmodells diskutiert.

Die Grundidee des Verfahrens besteht darin, in allen Intentionsmodellen Merkmalsknoten zu finden, deren Attribute sich mit den Merkmalen einer beobachteten Aktion decken. Die gefundenen Zustandsvariablen werden dann instanziiert, indem ihnen der ja-Zustand zugewiesen wird. Die Rückschlusswahrscheinlichkeiten des Intentionsknotens der Intentionsmodelle erlauben schließlich quantitative Aussagen darüber, welche Intention der Benutzer verfolgt.

Zunächst ist noch die Klärung der Nomenklatur des Merkmalsvektors und der einzelnen Merkmale nötig. Bei dem Merkmalsvektor werden drei Ausprägungen unterschieden: Der Vektor, der ausschließlich die aktuell beobachtete Benutzeraktion beschreibt, wird mit $\mathbf{m}_{t=0}$ symbolisiert. Der Zeitpunkt t=0 drückt aus, dass mit jeder neuen Beobachtung der Planerkennungsprozess erneut gestartet wird. Auf den Vektor, der ausschließlich die Merkmale vergangener Aktionen beinhaltet, wird mit $\mathbf{m}_{t<0}$ Bezug genommen. Die Vereinigungsmenge dieser Vektoren ist $\mathbf{m}_{t\leq 0}$, also der Vektor, der die Merkmale vergangener und der aktuellen Aktion zusammenfasst. Das k-te Merkmal des Vektors $\mathbf{m}_{t=0}$ wird mit $m_{t=0;k}$ beschrieben.

Zu Beginn wird den Intentionsknoten aller Intentionsmodelle die a-priori-Wahrscheinlichkeit $P(Intention) = (0{,}5;05)$ zugewiesen, um zu gewährleisten, dass alle Hypothesen gleichberechtigt sind. Die intentionsbasierte Interpretation unvollständiger Aktionssequenzen wird nun anhand von Abbildung 4.17 in den Schritten ① bis ⑩ behandelt:

① Der Merkmalsvektor $\mathbf{m}_{t=0}$ beschreibt die zuletzt beobachtete Benutzeraktion und erlaubt somit Aussagen über das verwendete Unix-Kommando und die adressierten Objekte bzw. Parameter. Da im Rahmen der Interpretation einer Aktionssequenz zwischen Kommandos zum Wechsel des aktuellen Verzeichnisses, Kommandos zum Sammeln von Informationen und den übrigen Unix-Befehlen unterschieden wird, wird an dieser Stelle abgefragt, um welchen Kommandotyp es sich

bei der letzen Eingabe handelt. Diese Information ist dem Merkmalstyp des Merkmals $m_{t=0;1}$ zu entnehmen.

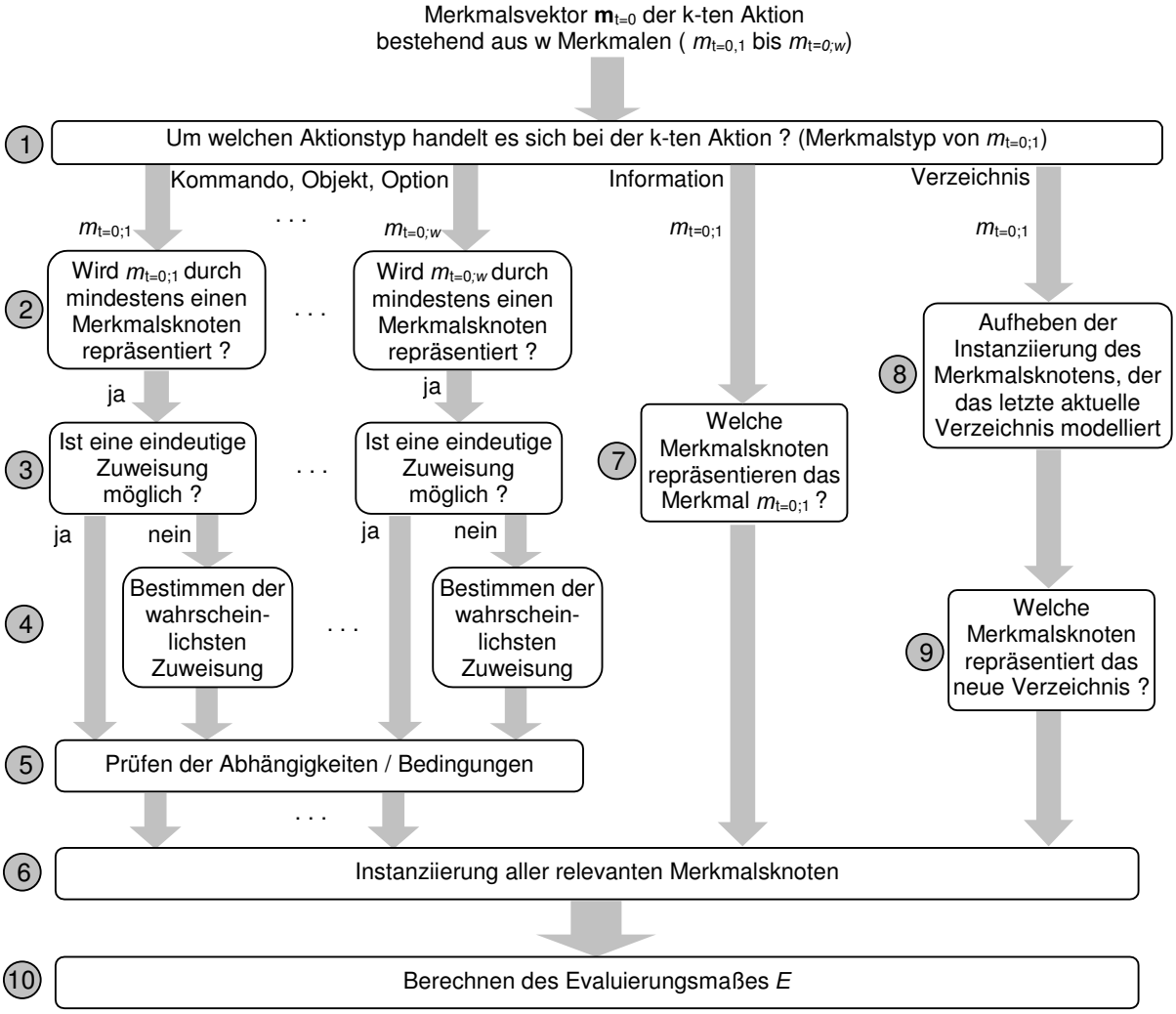


Abbildung 4.17: Ablauf der intentionsbasierten Interpretation unvollständiger Aktionssequenzen

② Falls die Merkmalsextraktion die letzte Benutzeraktion in der Kategorie Kommando, Option oder Objekt eingestuft hat, so werden für alle w Merkmale die Merkmalsknoten des Intentionsmodells gesucht, deren Zusatzinformationen mit diesen Merkmalen konsistent sind. Kommt kein Merkmalsknoten in Frage, so wird das betrachtete Merkmal in die weiteren Überlegungen nicht mit einbezogen.

③ Unter Umständen wird ein Merkmal von mehreren Merkmalsknoten desselben Intentionsmodells beschrieben. Trifft dies nicht zu, so wird direkt mit Schritt ⑤ fortgefahren. Gibt es mehrere Alternativen, sodass keine eindeutige Zuordnung erfolgen kann, so wird der wahrscheinlichste Knoten anhand von Schritt ④ bestimmt.

④ Um unter den Alternativen den Merkmalsknoten zu finden, der am wahrscheinlichsten dem beobachtete Merkmal entspricht, wird für alle in Frage kommenden Knoten m_x die Wahrscheinlichkeit $P(m_x = \text{ja} \mid \mathbf{m}_{t<0})$ berechnet. Der Vektor $\mathbf{m}_{t<0}$ beschreibt dabei alle bisher instanziierten Zustands-

variablen, d.h. die Merkmalsknoten, denen aufgrund bisheriger Benutzeraktionen feste Zustände zugewiesen wurden. Dem Knoten mit der höchsten Wahrscheinlichkeit $P(m_x = \text{ja} \mid \mathbf{m}_{t<0})$ wird schließlich das betrachtete Merkmal zugewiesen.

⑤ Nun ist zu prüfen, ob die Merkmalsknoten, die auf Grund von Übereinstimmung mit jeweils einem Merkmal instanziiert werden können, auch die hierfür nötigen Bedingungen erfüllen. Dazu werden für jedes Merkmal die in Kapitel 4.6.2 behandelten Bedingungen bzw. Abhängigkeiten von anderen Merkmalsknoten überprüft.

⑥ Nur wenn die an einem Merkmalsknoten geknüpften Abhängigkeiten erfüllt sind, wird der entsprechenden Zustandsvariable der Zustand „ja“ zugewiesen und dieses Merkmal somit in den Planerkennungsprozess mit eingebunden.

⑦ Handelt es sich bei der letzten Eingabe um einen Befehl zum Einholen von Informationen über ein Objekt, so werden alle Merkmalsknoten, die diese Aktion beschreiben, für die Instanziierung freigegeben.

⑧ Mit Aktionen zum Wechseln des aktuellen Verzeichnisses (*cd*-Kommando) wir in anderer Weise verfahren. Wie in Kapitel 4.6 erwähnt, wird das aktuelle Verzeichnis nicht anhand von Benutzeraktionen, sondern indirekt über die aktuelle Position in der Verzeichnisstruktur modelliert. Da immer nur ein Verzeichnis aktuell sein kann, dürfen nur die *Verzeichnis*-Knoten instanziiert sein, die dem aktuellen Verzeichnis entsprechen. Aus diesem Grund werden zunächst die Instanziierungen aller Knoten des Merkmalstyps *Verzeichnis* aufgehoben.

⑨ Im Intentionsmodell wird nach Merkmalsknoten gesucht, deren Attribute mit dem neuen Verzeichnis übereinstimmen. Alle Knoten, auf die dies zutrifft, werden instanziiert. Ähnlich wie die Knoten des Typs *Information* dürfen Verzeichnismerkmale auf mehrere Knoten abgebildet werden. Wird kein Knoten gefunden, so bleibt die letzte Benutzeraktion für das betrachtete Intentionsmodell ohne Relevanz.

⑩ Nachdem das Intentionsmodell nun den bisherigen sowie der zuletzt beobachteten Benutzeraktion Rechnung trägt, wird für diese Intention der Evaluierungswert *E* berechnet:

$$E = n_{\text{Objekte}} \cdot (P(\mathit{Intention} = \text{ja} \mid \mathbf{m}_{t\leq 0}) - P(\mathit{Intention} = \text{ja})) \tag{4-7}$$

Der Vergleich der Rückschlusswahrscheinlichkeit $P(\mathit{Intention} = \text{ja} \mid \mathbf{m}_{t\leq 0})$ mit der a-priori-Wahrscheinlichkeit $P(\mathit{Intention} = \text{ja})$ des Intentionsknotens zeigt, wie komplett die betrachtete Intentionshypothese durch die bisherige Aktionssequenz umgesetzt wurde. Da die Evaluierungswerte von Intentionshypothesen, die unter Umständen die Manipulation unterschiedlich vieler Objekte vorsehen, verglichen werden, wird die Anzahl der für die betrachtete Intentionshypothese relevanten Objekte n_{Objekte} hinzumultipliziert.

Das in den Schritten ①bis ⑩ beschriebene Vorgehen ist für jede neue Benutzeraktion und für jede Intentionshypothese durchzuführen. Die Intentionshypothese mit dem maximalen Evaluierungswert *E* ist das Ergebnis des Interpretationsprozesses.

Der vorgestellte Formalismus wird nun in Abbildung 4.18 am Beispiel eines konkreten Intentionsmodells anschaulicher dargestellt. Die Abbildung zeigt einen Ausschnitt des Intentionsmodells für die Intentionshypothese „Drucke und verschiebe alle Dateien des Verzeichnisses */Transfer* und verschiebe sie nach */Transfer/Neu*“. In diesem Beispiel sind zwei Objekte, die Dateien *bnetz* und *tk.ps* des Verzeichnisses */Transfer*, zu drucken und zu verschieben. Wurzelknoten des Netzes ist der Intentionsknoten, dessen Rückschlusswahrscheinlichkeit nach der intentionsbasierten Interpretation einer Aktionssequenz zur Evaluierung der Intentionshypothese herangezogen wird. Die Repräsentation der beiden relevanten Objekte geschieht in Analogie zu der in Abbildung 4.9 dargestellten Struktur. Für die Objekte sind jetzt die Operatoren zu definieren. Abbildung 4.18 zeigt dies am Beispiel des Objektes *bnetz*, dessen Vernetzung sich an der in Abbildung 4.10 vorgestellten Struktur orientiert. Für eine klarere Darstellung wurden die beiden Operatorknoten nach deren Aufgabe benannt (*Verschieben* und *Drucken*). Die Subnetzstruktur des *Verschieben*-Operators gleicht dem in Kapitel 4.6 exemplarisch entwickelten Operator-Subnetz. Der untere Bildrand zeigt einen Ausschnitt einer Aktionssequenz. Das Abbilden der Aktionen 1 bis 6 auf das Bayes’sche Netz wird nun geschildert.

Bei der ersten Eingabe handelt es sich um eine Aktion zum Betreten des Verzeichnisses */Transfer*. Die Merkmalsextraktion bestimmt die Merkmale dieser Aktion, in diesem Fall den Typ und einen ID-String. Diese Informationen gleichen den Attributen der Knoten m_8 und m_{10}, die deshalb beide instanziiert werden, d.h., ihnen wird der ja-Zustand zugewiesen. Diese Maßnahmen wirken sich über die Knoten m_2 und m_3 direkt auf den Operatorknoten bis hin zum Intentionsknoten aus. Die betrachtete Intentionshypothese wird schließlich auf Basis dieser ersten Beobachtung in der oben beschriebenen Weise bewertet.

Die zweite Aktion besteht im Sinne der Intentionshypothese aus einem falschen Kommando und einem richtigen Objekt. Da sich aber von dem Kommando „move“ auf den korrekten Befehl „mv“ schließen lässt, stärkt diese Eingabe dennoch die Annahme, dass die Benutzerintention der betrachteten Intentionshypothese gleicht. Das Kommando wird auf Knoten m_4 abgebildet, das Objekt auf Knoten m_2. Das Prüfen der Abhängigkeiten bestätigt, dass die Voraussetzungen für die Instanziierung beider Knoten erfüllt sind, somit wird ihnen der ja-Zustand zugewiesen. Dies schlägt sich entsprechend auf den Evaluierungswert E_2 nieder, der den ersten beiden Aktionen Rechnung trägt.

Für die Merkmale der dritten Aktion gibt es keinen korrespondierenden Merkmalsknoten, sodass das die Evaluierungswerte E_3 und E_2 identisch sind. Mit der vierten Aktion wird analog zu den vorhergehenden Beobachtungen verfahren.

Die fünfte Aktion bezweckt einen Verzeichniswechsel, der durch Instanziierung des Merkmalsknotens m_{11} im Bayes’schen Netz berücksichtigt wird. Die Instanziierung aller übrigen Merkmalsknoten, die ein konkretes Verzeichnis modellieren, wird rückgängig gemacht, d.h., Knoten m_8 und m_{10} beschreiben jetzt keine sicheren Beobachtungen mehr.

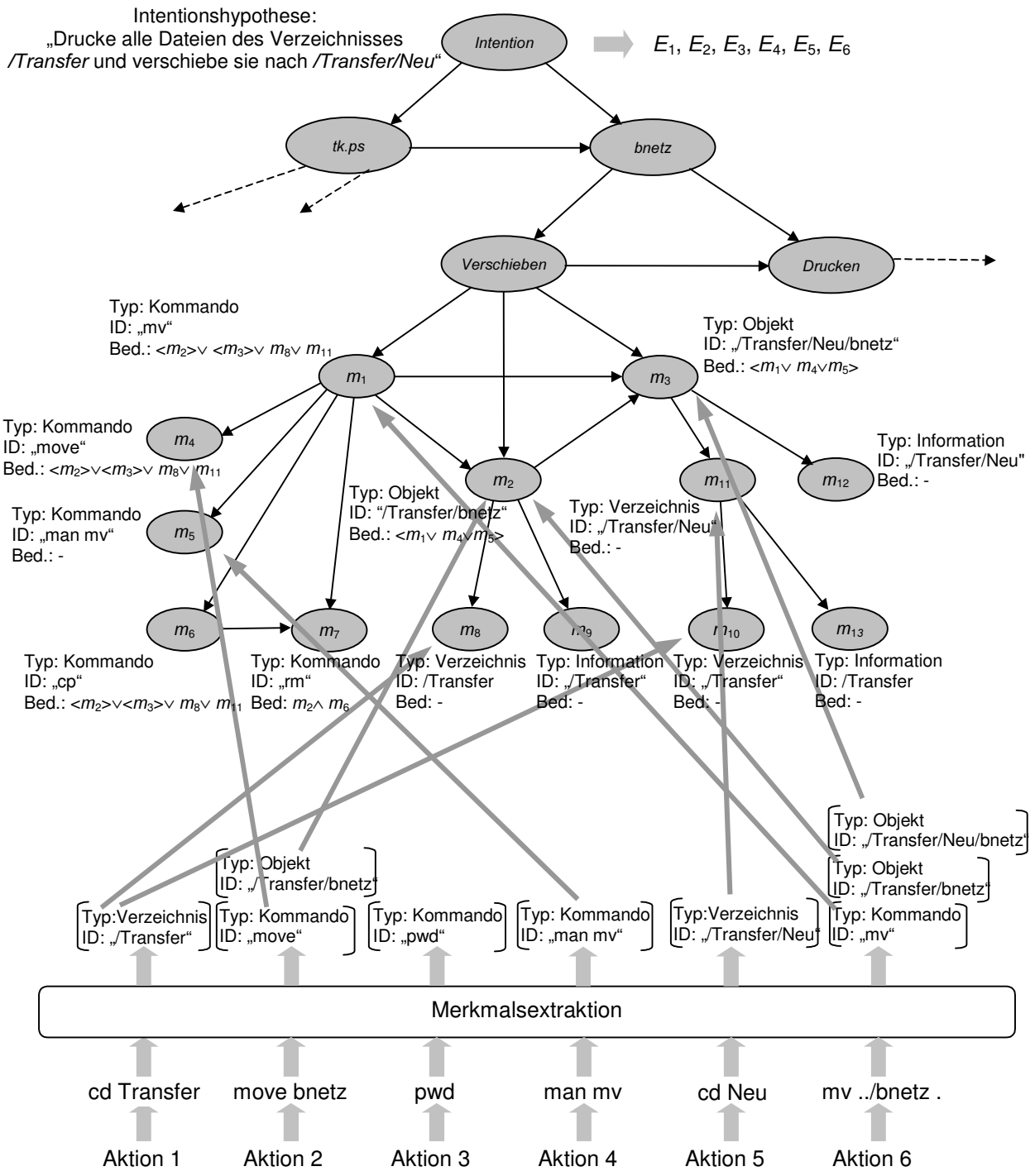


Abbildung 4.18: Beispiel für die intentionsbasierte Interpretation eine Aktionssequenz auf Basis eines Intentionsmodells

Mit der sechsten Aktion werden schließlich die Knoten m_1, m_2 und m_3 instanziiert, sodass der Operator *Verschieben* auf Grund der in Kapitel 4.6 vorgestellten UND-Modellierung als vollständig beobachtet gewertet wird. Entsprechend steigt der Evaluierungswert E_6.

4.8 Ergebnisse und Diskussion

Um die Klassifikationsleistung von *AMPlan* zu dokumentieren, wurden die zehn Intentionshypothesen bzw. Ziele der Intentionsbibliothek 16 Versuchspersonen als Aufgabe gegeben. Die resultierenden 160 Pläne wurden herangezogen, um Aussagen über die Erkennungsleistung des Planerkenners zu machen. Da es sich bei der Planerkennung um einen Mustererkennungsprozess handelt, der auf Basis unvollständiger Beobachtungsfolgen klassifiziert, sind Erkennungsraten nur in Abhängigkeit von der Vollständigkeit der Aktionssequenzen aussagekräftig. Abbildung 4.19 zeigt diese Klassifikationsraten.

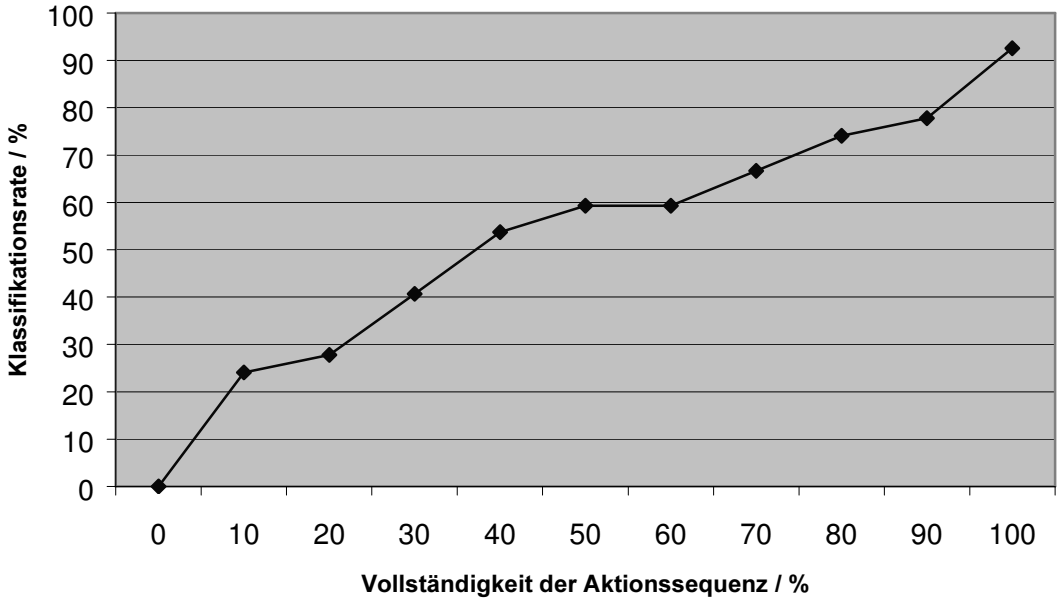


Abbildung 4.19: Klassifikationsrate in Abhängigkeit von der Vollständigkeit der Aktionssequenz

Bereits nach 10% der Aktion erhöht sich die Erkennungsrate deutlich. Dies ist darauf zurückzuführen, dass die ersten Aktionen in der Regel dazu dienen, ein bestimmtes Verzeichnis aufzusuchen. Somit kann bereits nach wenigen Aktionen eine Tendenz rein auf Basis suboptimaler Aktionen abgeschätzt werden.

Im weiteren Verlauf steigt die Klassifikationsleistung erwartungsgemäß an. Nach der Hälfte der Aktionssequenz (50 %) konnte die Benutzerintention zu 60% korrekt ermittelt werden. Dieses Ergebnis dokumentiert das Potenzial von *AMPlan*, als Basis für eine adaptive, zielgerichtete Dialogführung eingesetzt werden zu können, da bereits lange vor Abschluss der Aktionssequenz Aussagen über die Benutzerintention getroffen und eine Vielzahl falscher Intentionshypothesen ausgeschlossen werden können.

Zu bemerken ist, dass sich die Erkennungsraten aus Abbildung 4.19 auf die in Abbildung 4.5 vorgestellte Intentionsbibliothek mit zehn Einträgen bezieht. Würde man 100 Intentionshypothesen mit einbeziehen, so würden diese Erkennungsraten sicherlich sinken.

Dass die Erkennungsrate bei vollständiger Aktionssequenz (100 %) nur 92,6 Prozent beträgt und nicht 100 Prozent, ist darauf zurückzuführen, dass die Intentionsbibliothek zwei Hypothesen beinhaltet, deren Ziele Untermengen bzw. Teilziele anderer Hypothesen sind. Abbildung 4.20 dient der Diskussion dieses Phänomens. Dargestellt ist die „Intention X", die nur aus der Manipulation eines Objekts besteht. Dass dieser Plan bereits vollständig beobachtet wurde, soll durch den ausgefüllten Kreis verdeutlicht werden. Das Ziel von „Intention X" ist ein Teilziel von „Intention Y" und wurde auch für diese Intentionshypothese als komplett beobachtet gewertet. Darüber hinaus gibt es ein weiteres Ziel von „Intention Y", für das bisher keine Aktionen beobachtet wurden. Die Farben der Kreise visualisieren dies. Berechnet man nun die Evaluierungsmaße beider Intentionshypothesen durch Anwendung von Gleichung (4-8) und geht davon aus, dass „Intention X" vollständig ($P(\mathit{Intention} = \text{ja} \mid \mathbf{m}_{t\leq 0}) = 1$) und „Intention Y" zur Hälfte ($P(\mathit{Intention} = \text{ja} \mid \mathbf{m}_{t\leq 0}) = 0{,}75$) beobachtet wurde, so ergeben sich identische Werte. Für diese Konstellation von Intentionshypothesen ist der Planerkenner somit blind. Für die Klassifikation der Intention auf Basis der Testpläne stellt dies aber nur in Ausnahmefällen ein Problem dar. Dieses Phänomen könnte durch einfache Zusatzregeln behoben werden, die aber im Rahmen dieser Arbeit nicht untersucht wurden.

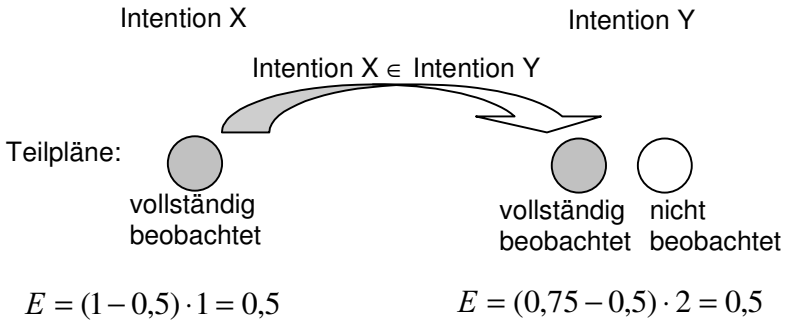


Abbildung 4.20: Falls ein Ziel einer Intentionshypothese einem Teilziel einer anderen Hypothese gleicht, kann dies zu identischen Evaluierungswerten führen

Um einen prinzipiellen Eindruck von den Evaluierungsmaßen zu vermitteln, sind in Abbildung 4.21 die Evaluierungswerte aller Intentionshypothesen in Abhängigkeit von einer Aktionssequenz, bestehend aus 20 Eingaben, dargestellt. Die zeitliche Reihenfolge der Aktionen orientiert sich in der Abbildung von links nach rechts. Wie das Histogramm zeigt, kann bereits nach wenigen Aktionen eine Reihe Intentionshypothesen als Benutzerintention ausgeschlossen werden. Nach Beobachtung der halben Aktionssequenz kristallisiert sich eine bestimmte Intention zunehmend als die wahrscheinliche Benutzerintention heraus.

Für die Klassifikation ist in erster Linie das Verhältnis der einzelnen Evaluierungswerte zueinander interessant. Die absoluten Evaluierungswerte sind hilfreich, um das „Vertrauen" in die Klassifikation quantitativ zu untermauern, d.h., der Evaluierungswert entspricht einem Konfidenzwert. Hier zahlt sich die Wahrscheinlichkeitstheorie als mathematisches Fundament des Verfahrens aus, da Erkennungsergebnisse einfach und eindeutig interpretierbar sind. Vor allem für eine auf *AMPlan* aufbauende Dialogmodellierung ist interessant, dass auf diese Weise mehrere ähnlich wahrscheinliche Intentionshypothesen berücksichtigt werden können, anstatt sich eventuell zu früh auf eine Hypothese festzulegen.

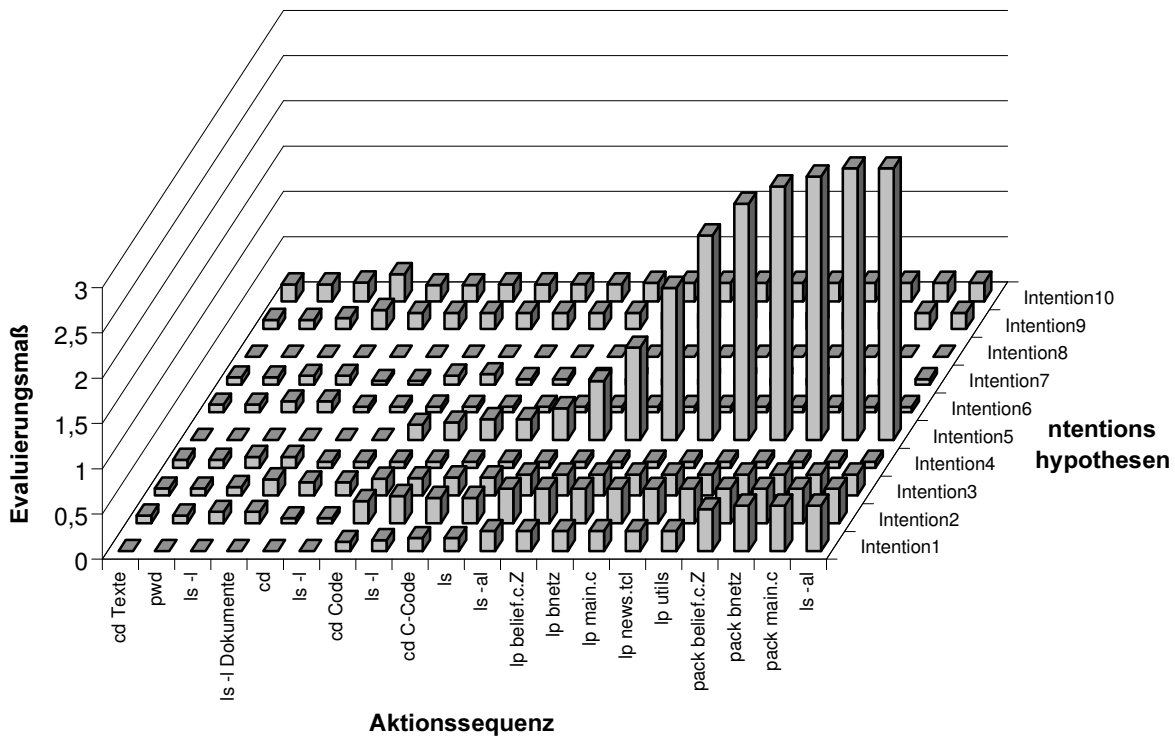


Abbildung 4.21: Evaluierungsmaße aller Intentionshypothesen in Abhängigkeit der Benutzeraktionen

Auf Grund der Merkmalsselektion werden ausschließlich die für jede Intentionshypothese relevanten Aktionen aus der Aktionsabfolge herausgesucht. Dies ermöglicht nicht nur einen robusten Umgang mit unbekannten Befehlen oder nicht modelliertem Fehlverhalten des Benutzers, sondern auch eine parallele Betrachtung mehrerer Benutzerintentionen. Der Benutzer könnte also zum Beispiel zwei Ziele gleichzeitig verfolgen, sodass sich beide Pläne zu einer großen Aktionssequenz verzahnen. Dies stellt für *AMPlan* kein Problem dar.

Die Tatsache, dass der in Kapitel 4.7 vorgestellte Algorithmus zur intentionsbasierten Interpretation von Aktionssequenzen nur aus wenigen Regeln besteht, ist auf die leistungsfähigen Intentionsmodelle zurückzuführen, die auf Grund ihrer Topologie einen Großteil der Planerkennungslogik realisieren. Hierfür wurde von den Möglichkeiten der Bayes'schen Netze, syntaktisch-semantische Zusammenhänge darzustellen, konsequent Gebrauch gemacht.

4.9 Implementierung von *AMPlan*

Die Implementierung des Planerkenners *AMPlan* wurde unter Linux in C/C++ [Str92] vorgenommen. Abbildung 4.22 zeigt die Systemarchitektur. Kernkomponente ist ein zentraler Server, der die Merkmalsextraktion sowie die intentionsbasierte Interpretation realisiert.

Bei Start von *AMPlan* werden zunächst sowohl alle in Betracht zu ziehenden Intentionshypothesen als auch deren Intentionsmodelle aus Dateien eingelesen. Für die Intentionsmodelle ist jeweils eine Datei mit der Topologie des Bayes'schen Netzes, eine Datei mit den bedingten Wahrscheinlichkeiten der Knoten und eine Datei mit den Attributen, die den Merkmalsknoten zugewiesen werden,

einzulesen. Für einen effizienten Einsatz der Bayes'schen Netze wurde der von Jenson [Jen92][Jin99] entwickelte Junction-Tree-Algorithmus implementiert.

Die Merkmalsextraktion liest bei Starten von *AMPlan* eine Datei mit allen zulässigen Unix-Kommandos und deren Syntax ein. Diese Informationen gewährleisten schließlich die korrekte Zerlegung einer Benutzeraktion in ihre Bestandteile.

Das Domänenmodell analysiert die aktuelle Verzeichnisstruktur und schickt die Informationen über Verzeichnisse und deren Dateien und Unterverzeichnisse an die Merkmalsextraktion. Das Domänenmodell ist als separater Prozess in C++ realisiert und kommuniziert mit dem zentralen Server als Client.

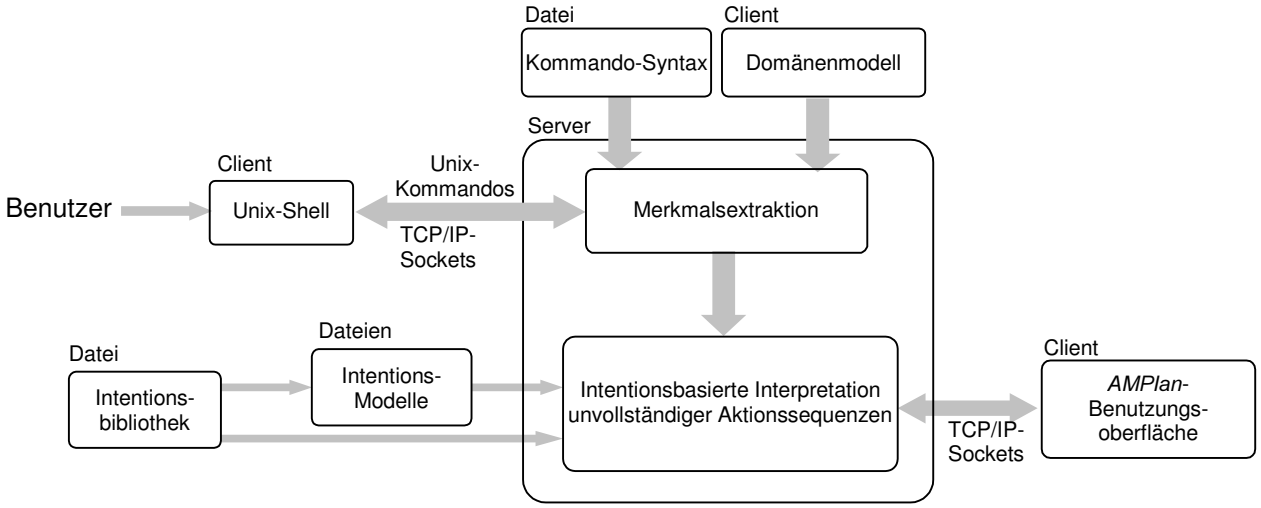


Abbildung 4.22: Systemarchitektur der *AMPlan*-Implementierung

Als Eingabeschnittstelle wurde ein Programm entwickelt, dass das Verhalten und die Funktionalität einer Standard-Unix-Shell exakt nachbildet. Eine „echte" Unix-Shell konnte auf Grund mangelnder Eingriffsmöglichkeiten, die für das Auslesen einer Benutzereingabe benötigt werden, nicht herangezogen werden. Dieser Prozess wurde als Client realisiert, der eine Benutzeraktion direkt an die Merkmalsextraktion weiterleitet.

Die Merkmalsextraktion hat das nötige Wissen, um eine eingegebene Aktion auf Plausibilität zu prüfen. Hierfür wird die Syntax des eingegebenen Befehls analysiert sowie eventuell angegebene Verzeichnis- und Dateinamen mit den vom Domänenmodell bereitgestellten Informationen verglichen. Handelt es sich um eine korrekte Eingabe, so wird diese im realen Dateisystem mit entsprechenden Bildschirmausgaben ausgeführt. Handelt es sich um eine fehlerhafte Eingabe, so reagiert das Programm mit den Unix-typischen Fehlermeldungen. Dennoch wird die Eingabe in diesem Fall für die Planerkennung herangezogen. Abbildung 4.23 zeigt die Eingabeschnittstelle.

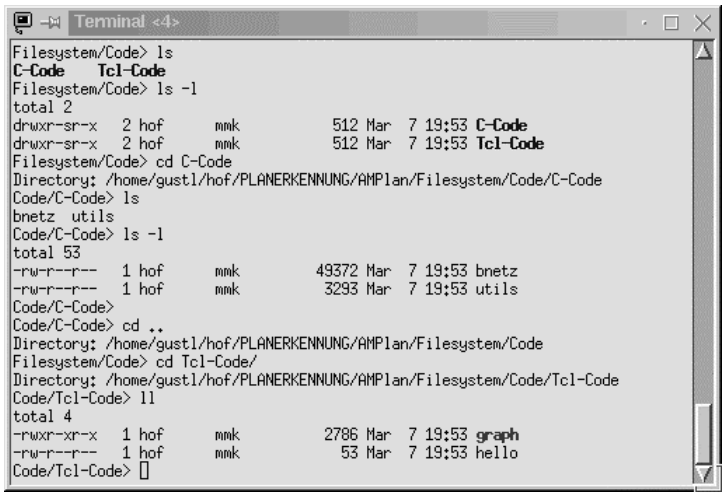


Abbildung 4.23: Das Verhalten der Eingabeschnittstelle ist identisch mit einer Unix-Shell

Die *AMPlan*-Benutzungsoberfläche wurde in Tcl/Tk [Wel99] als Client realisiert wurde. Dieser Client ermöglicht jederzeit Auskunft über die Evaluierungsmaße aller Intentionshypothesen der Intentionsbibliothek. Darüber hinaus können jedes Intentionsmodell visualisiert, die instanziierten Merkmalsknoten farblich hervorgehoben sowie die Randwahrscheinlichkeit jedes Knotens berechnet werden. Somit gewährleistet diese Komponente eine maximale Transparenz des Planerkenners. Abbildung 4.24 zeigt einen Screenshot dieser Komponente; dargestellt sind die Evaluierungsmaße aller zehn Hypothesen in Histogrammform.

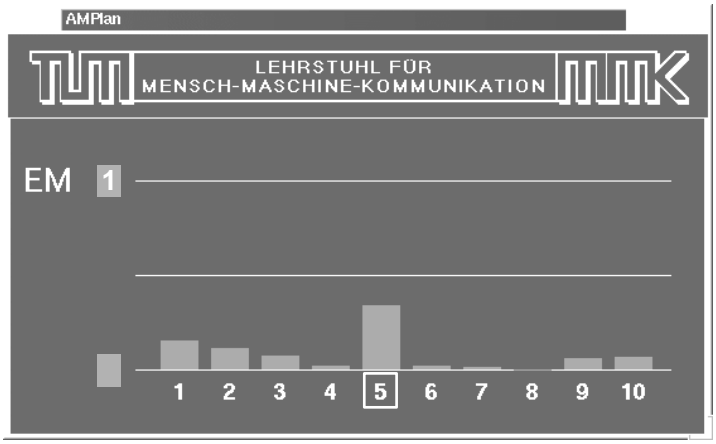


Abbildung 4.24: Die *AMPlan*-Benutzungsschnittstelle erlaubt Auskunft über Evaluierungsmaße der Intentionshypothesen in Form eines Histogramms

Die Kommunikation zwischen dem zentralen Server und seinen Clients geschieht ausnahmslos über bidirektionale TCP/IP-Sockets.

Die Information über die wahrscheinlichste Benutzerintention kann sehr vielfältig genutzt werden. Nahe liegend ist eine adaptive Dialogführung oder ein Tutorsystem basierend auf *AMPlan*. Als konkretes System wurde ein benutzeradaptives Assistenzsystem entwickelt, das dem Benutzer eine automatische Ausführung seiner nächsten Aktionen bzw. Teilpläne anbietet. Da es sich hierbei um

ein sehr komplexes System zur adaptiven Dialogmodellierung handelt, wird an dieser Stelle aber nicht näher darauf eingegangen, sondern auf [Sch00] und [Hof01c] verwiesen.

5

Intentionsbasierte Interpretation von Situationen und Aktionen: Benutzermodellierung

Dieses Kapitel beschreibt die intentionsbasierte Interpretation von Situationen und Aktionen. Durch die Fähigkeit, Vorhersagen bezüglich der Benutzerintention zu treffen, ist das entwickelte System in der Lage, eine adaptive, auf den Nutzer und die Situation individuell zugeschnittene Dialogführung anzubieten.

5.1 Grundidee

Im Kapitel 3 wurde das sprachverstehende System *Insense* vorgestellt. Die durch Sprache steuerbare Applikation entsprach den gegenwärtigen informationstechnischen Einrichtungen eines Automobils. Sprachsteuerung erweist sich gerade für die Automobil-Domäne als besonders sinnvoll, weil die Ablenkung des Fahrers erheblich minimiert und somit die Sicherheit der Fahrzeuginsassen erhöht wird. Vor allem Fahrzeug-Navigationssystemen müssen durch den Fahrer komplexe Informationen, wie zum Beispiel Zielortsnamen, zur Verfügung gestellt werden. Deshalb sollten gerade Navigationssysteme aus Sicherheits- und Komfortaspekten sprachgesteuert bedienbar gemacht werden. Erschwert wird eine reibungslose Sprachsteuerung von Navigationssystemen jedoch durch zwei Randbedingungen:

- Ein Problem bei der sprachlichen Eingabe eines Ortsnamens besteht in der Tatsache, dass der Fahrer oft den korrekten offiziellen Ortsnamen nicht kennt bzw. nicht spricht. In der Regel wird der Fahrer beispielsweise *Neustadt* als Ziel angeben und nicht die korrekte Bezeichnung, z.B. *Neustadt an der Aisch*. Aus der Mehrdeutigkeit ca. 50 Prozent aller deutschen Ortsnamen ergibt sich somit das Problem, dass das Navigationssystem nicht in der Lage ist, die interpretierte Information dem richtigen Ort zuzuweisen. Aktuelle Navigationssysteme können diese Mehrdeutigkeiten nicht auflösen und erfordern Klärungsdialoge. Kognitiv nicht unmittelbar erfassbare, komplexe Systemrückfragen erhöhen jedoch die Ablenkung des Fahrers und damit das Sicher-

heitsrisiko. Nicht zuletzt reduzieren sie den Komfort und die Freude am Umgang mit dem System.

- Ein weiteres Problem liegt in der Tatsache, dass das Spracherkenner-Vokabular für die sprachliche Eingabe des Zielorts allein für Deutschland ca. 100000 Einträge umfasst. Proportional zu dieser Zahl ist die Fehlerrate des Spracherkenners, d.h., bei dem derzeitigen Stand der Technik müsste der Fahrer rein statistisch dem System den Zielort mehrfach per Sprache mitteilen, bevor der Zielort korrekt erkannt wird. Im Fall einer Häufung einer solchen Situation würde die Akzeptanz der sprachlichen Schnittstelle und somit die Akzeptanz des gesamten Systems reduziert. Da unvorhersehbares Verhalten eines informationstechnischen Systems für zusätzliche Ablenkung sorgt, beeinträchtigen Fehlerkennungen zudem die Sicherheit.

Für eine sprachliche Interaktion mit Navigationssystemen ist die Reduzierung der erwähnten Phänomene unumgänglich. Für einen Beifahrer stellen die diskutierten Phänomene kein Problem dar. Ein Beifahrer könnte einen vom Fahrer nicht korrekt oder undeutlich artikulierten Ortsnamen interpretieren, vorausgesetzt er kennt die Interessen und Präferenzen des Fahrers. Mit diesem Hintergrundwissen könnte er den Ortsnamen nicht nur wahrnehmen, sondern verstehen und dem Wunschziel des Fahrers zuordnen. An dieser Fähigkeit des Menschen sollte sich auch eine leistungsfähige Eingabeschnittstelle orientieren. Ziel ist es, den intentionsbasierten Ansatz auf die hier vorgestellte Problematik so anzupassen, dass das Navigationssystem in der Lage ist, den Wunschzielort zu verstehen. Bei dem Wunschzielort handelt es sich in diesem Fall um die Benutzerintention. Da zunächst nichts von individuellen Interessen und Präferenzen des Fahrers bezüglich seiner Zielorte bekannt ist, muss das System diese erlernen. Somit resultiert der intentionsbasierte Ansatz in ein System zur Benutzermodellierung, das sich adaptiv an den aktuellen Fahrer anpasst und ihn kennen lernt. Das erlernte Wissen wird schließlich herangezogen, um nicht eindeutige Zielorteingaben zu *verstehen*.

Die Benutzermodellierung dient schließlich dazu, die Kommunikation mit einem Fahrzeugnavigationssystem hinsichtlich folgender Aspekte zu verbessern:

- **Disambiguierung von Zielorteingaben**
  Das erlernte Wissen über die Interessen und Präferenzen des Fahrers bezüglich seiner potenziellen Zielorte wird herangezogen, um Zielorteingaben zu verstehen und aus mehrdeutigen Angaben das tatsächliche Wunschziel zu ermitteln. Die Auflösung dieser Mehrdeutigkeiten wird als *Disambiguierung* bezeichnet.

- **Begrenzung bzw. Gewichtung des Ortsnamen-Vokabulars**
  Für Fahrer, die im Allgemeinen nur in bestimmten Gegenden agieren, wird dem Spracherkenner ein entsprechend reduziertes Vokabular angeboten, um die Erkennungsleistung des Spracherkenners zu erhöhen. Zudem ermöglicht das Verfahren die Nachevaluierung und Interpretation des Erkennungsergebnisses, um dieses auf Basis der Benutzermodellierung auf Plausibilität zu untersuchen und gegebenenfalls andere Erkennerhypothesen vorzuziehen.

- **Bestimmung eines Default-Zielorts**
  Aktuelle Navigationssysteme bringen folgende Situation mit sich: Startet der Fahrer das Navigationssystem, so wird ihm als Default-Zielort der zuletzt eingegebene Zielort angeboten. In der Regel handelt es sich dabei allerdings um den aktuellen Standort. Um diese paradoxe Situation aufzulösen, ist das dritte Ziel des Verfahrens eine intelligente, adaptive Wahl des Default-Zielorts, so dassder Fahrer lediglich bestätigen muss oder gegebenenfalls einen anderen Zielort eingeben kann. Die Zielgruppe für diese Systemeigenschaft sind zum Beispiel Pendler, die regelmäßig die gleichen Orte aufsuchen.

Das übergeordnete Ziel des Verfahrens ist eine Reduzierung von Klärungsdialogen und von Fehlerkennungen. Die resultierende schnellere Konvergenz des Dialogs führt zu einer benutzerfreundlicheren, robusteren sprachlichen und damit intuitiveren Interaktion mit Navigationssystemen. In diesem Kapitel wird das hierfür entwickelte Navigationsassistenzsystem *Adaptive Compass* [Hof01b] vorgestellt.

Adaptive Compass erfasst Korrelationen zwischen der aktuellen Situation und den Interessen des Fahrers an einem bestimmten Ort. Diese semantischen Zusammenhänge werden vom Intentionsmodell bei eindeutigen Zielorteingaben anhand einer *Trainingsphase* erlernt. Ein ausreichend trainiertes Intentionsmodell erlaubt Aussagen über die Zielortinteressen des Fahrers für jede beliebige Situation. Der Navigationsassistent ist somit in der Lage, Zielorte situativ vorherzusagen, um gegebenenfalls Mehrdeutigkeiten aufzulösen oder um das Spracherkenner-Vokabular auf die wahrscheinlichsten Orte zu reduzieren. Darauf wird im Folgendem als *Zielortprädiktion* verwiesen. Die Realisierung von *Adaptive Compass* orientiert sich an der zweiten Ausprägung des intentionsbasierten Ansatzes (Kapitel 2).

Eine besondere Fähigkeit von *Adaptive Compass*, die für einen sinnvollen praktischen Einsatz Voraussetzung ist, wird an dieser Stelle kurz betont. Der Navigationsassistent ist in der Lage, Aussagen über bisher noch nicht beobachtete Situationen sowie über bisher noch nicht vom Fahrer aufgesuchte Orte zu treffen. In Analogie zum menschlichen Verstehen von Zusammenhängen wird in *Adaptive Compass* die Fähigkeit des *Transferdenkens* umgesetzt. Dies und die Tatsache, dass die Intentionsbibliothek bis zu 100000 Intentionshypothesen umfassen kann, macht die Berücksichtigung einzelner Orte unmöglich und erfordert eine Modellierung der Fahrerinteressen an bestimmten Orten in Abhängigkeit von den charakteristischen Eigenschaften von Gegenden. Im Detail wird dies in den folgenden Kapiteln vorgestellt.

5.2 Stand der Technik

In der Literatur wurden keinerlei Hinweise auf ein mit *Adaptive Compass* vergleichbares System gefunden, obwohl die Probleme, die das Verfahren adressiert, in der Automobilindustrie allgemein bekannt sind. Lösungen für das Problem, bei nicht eindeutigen Zielorteingaben die richtige Alternative zu wählen, wurden bisher nicht untersucht.

Für das Problem der Vokabularbegrenzung wurden nur ein sehr pragmatischer Ansätze untersucht: Das Spracherkenner-Vokabular beschränkte sich dabei ausschließlich auf die 5000 größten Städte

Deutschlands. Dieser Ansatz benötigt zwar wenig Rechenleistung, berücksichtigt aber nicht einmal ca. fünf Prozent aller Orte Deutschlands. Für Fahrer, die ländliche Gegenden aufsuchen wollen, ist eine direkte sprachliche Eingabe des Zielorts nicht möglich. Premium-Fahrzeuge der nächsten Generation sehen im Falle schlecht erkannter Ortseingaben die Frage nach einem benachbarten Ort vor. Aus beiden unter Umständen schlecht erkannten Ortsnamen lässt sich relativ robust der tatsächliche Zielort bestimmten. Bei einem Wunschziel in einer für den Fahrer unbekannten Gegend schlägt dieser Ansatz jedoch fehl.

In der Literatur finden sich mehrere Hinweise auf den Einsatz von Bayes'schen Netzen zur Benutzermodellierung [Jam96][Hor97]. Die Bandbreite reicht dabei von Systemen zur adaptiven Hilfestellung [Hor98] bis hin zu lehrenden Tutorsystemen [Con97]. Darüber hinaus wurden Bayes'sche Netze erfolgreich als Kernkomponente für Vorhersagesysteme eingesetzt [Dag92]. Diese Systeme sind jedoch nur schwer miteinander vergleichbar, da nicht das Netz selbst, sondern die für das Netz meist nötige Aufbereitung von Informationen und Daten den Hauptaufwand dargestellt und sehr stark durch das Anwendungsgebiet geprägt ist. Dies gilt auch für das hier vorgestellte Navigationsassistenzsystem *Adaptive Compass*.

5.3 Systemarchitektur

Zunächst wird die grundlegende Architektur von *Adaptive Compass* anhand von Abbildung 5.1 allgemein erläutert. Diese ist nahezu identisch mit der in Abbildung 2.4 vorgestellten allgemeinen Struktur des intentionsbasierten Ansatzes. Die Eingangsgrößen variieren mit der Betriebsart. Im *Trainingsmodus* wird stets die aktuelle Zieleingabe sowie die aktuelle Situation an die Merkmalsextraktion übergeben. Im *Prädiktionsmodus* dient lediglich die Situation als Eingangsgröße, während die Abwägung des Zielorts die Aufgabe des Navigationsassistenzsystems darstellt.

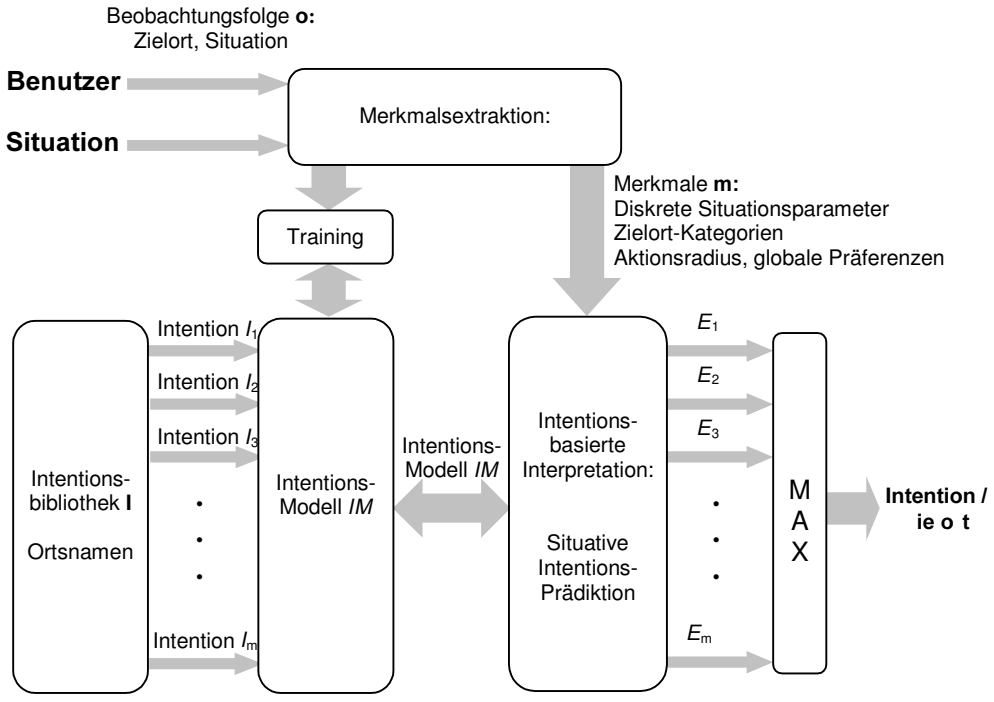


Abbildung 5.1: Systemüberblick über *Adaptive Compass*

Da die Benutzerintention im Rahmen von *Adaptive Compass* und der Wunschzielort äquivalent sind, handelt es sich bei der Intentionsbibliothek um eine Datenbank mit allen potenziellen Zielorten als Intentionshypothesen.

Der Zweck der Merkmalsextraktion ist eine diskrete Beschreibung der aktuellen Situation sowie die Charakterisierung eines eingegebenen Zielorts bezüglich bestimmter Eigenschaften. Diese charakteristischen Attribute der Intentionshypothesen werden in der Trainingsphase durch das Intentionsmodell mit den Situationsparametern in Bezug gesetzt.

Bekommt der Navigationsassistent schließlich vom Dialogmodell des Navigationssystems die Aufforderung, Mehrdeutigkeiten aufzulösen, so werden die in Frage kommenden Orte durch eine intentionsbasierte Interpretation der Situation quantitativ bewertet. Der Ort mit dem maximalen Evaluierungsmaß E ist aus Sicht von *Adaptive Compass* das wahrscheinlichste Ziel der Fahrers und somit die Benutzerintention.

Die angesprochenen Komponenten von *Adaptive Compass* werden in den folgenden Abschnitten ausführlich beschrieben. Schwerpunkte bilden hierbei die Merkmalsextraktion, die Umsetzung und das Training des Intentionsmodells sowie die situative Prädiktion der Benutzerintention.

5.4 Intentionsbibliothek

Die Domäne von *Adaptive Compass* ist eine Komponente eines Fahrzeugnavigationssystems, die Mensch-Maschine-Schnittstelle für die Eingabe eines Zielortes. Da die Benutzerintention und der Wunschzielort des Fahrers identisch sind, enthält die Intentionsbibliothek **I** alle potenziellen Zielortschaften. Je nach Betriebsmodus variieren allerdings die Intentionshypothesen:

- **Disambiguierung:** Im Falle nicht eindeutiger Zielorteingaben besteht die Intentionsbibliothek aus allen in Frage kommenden Alternativen. Mit 23 Alternativen bringt die Stadt *Neustadt* die meisten Intentionshypothesen mit sich.

- **Einschränken des Ortsnamenvokabulars; Interpretation sprachlicher Zielorteingaben:** Soll das Ortsnamenvokabular für eine robustere Spracherkennung fahrer- und situationsadaptiv begrenzt werden, so beinhaltet die Intentionsbibliothek alle im Navigationssystem berücksichtigten deutschen Orte, d.h. ca. 100000. Soll eine sprachliche Zielorteingabe auf Plausibilität geprüft werden, so besteht die Intentionsbibliothek aus den Zielorthypothesen, die der Spracherkenner ausgibt. Für jede dieser Intentionshypothesen wird zudem das durch den Spracherkenner bestimmte Konfidenzmaß c in der Intentionsbibliothek gespeichert.

- **Bestimmung des Default-Zielorts:** Soll bei Fahrtantritt dem Fahrer ein sinnvoller, konkreter Zielort als Default-Einstellung angeboten werden, so besteht die Intentionsbibliothek in diesem Fall aus allen bereits vom Fahrer aufgesuchten Orten. Mit jedem neu eingegebenen eindeutigen Ziel wird die Intentionsbibliothek um diesen Ort als neue Intentionshypothese erweitert.

5.5 Merkmalsextraktion

Eine direkte Korrelation zwischen Zielort und Situation im Trainings-Modus zu berechnen und als Basis für eine Zielort-Prädiktion zu verwenden ist in der Praxis nicht sinnvoll, da der Navigationsassistent dann nur in der Lage wäre, Aussagen über bereits aufgesuchte Orte zu treffen. Zudem wären in diesem Fall Prädiktionen ausschließlich für exakt bekannte Situationen möglich. Somit wären zu viele Beobachtungen und Trainingsphasen nötig, bevor das System zuverlässige Aussagen über den potenziellen Zielort des Fahrers treffen könnte. Dies erfordert zwei grundlegende Fähigkeiten von *Adaptive Compass*:

- **Evaluierung bisher unbekannter Situationen:** Durch Modellierung der Korrelation von Zielortinteressen und einzelnen Situationsparametern ist *Adaptive Compass* in der Lage, Zielort-Prädiktionen auch für unbekannte bzw. noch nicht exakt beobachtete Situationen vorzunehmen.

- **Evaluierung vom Fahrer noch nicht aufgesuchter Orte:** Durch Modellierung der Korrelation von Situation und bestimmten charakteristischen Eigenschaften von Orten ist *Adaptive Compass* in der Lage, Aussagen über vom Fahrer noch nicht aufgesuchte Orte zu treffen.

Beide Aspekte bestimmen die Aufgabe der Merkmalsextraktion, aus den Informationen bezüglich der Situation und des eingegebenen Zielorts, die für das Intentionsmodell und die situative Prädiktion relevanten Merkmale herauszufiltern, um somit Situationen und Orte interpretierbar zu machen. Neben *Situations-* und *Ortsmerkmalen* werden *Aktionsradiusmerkmale* und *globale Benutzerpräferenzen* ausgewertet, die direkt aus der Beobachtungsfolge gewonnen werden können. Die verschiedenen Merkmalstypen und deren Bestimmung werden im folgenden Abschnitt im Detail erläutert.

5.5.1 Situationsmerkmale

Zunächst wird Situationsbegriff im Rahmen des Systems definiert. Der Situationsbegriff umfasst unterschiedlichste Parameter aus den Bereichen Gesamtfahrzeug, Fahrzeugumgebung und den informationstechnischen Einrichtungen des Fahrzeugs. Abbildung 5.2 stellt dies schematisch dar:

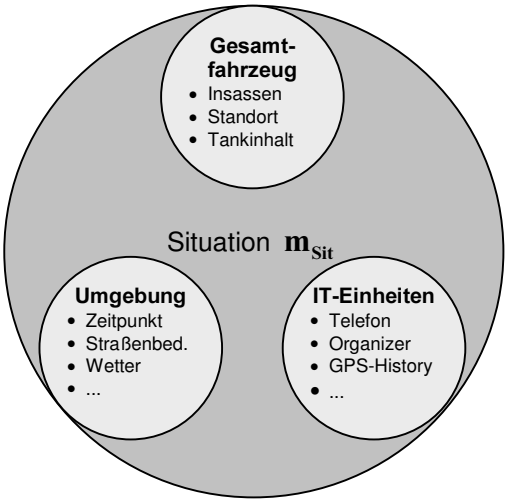


Abbildung 5.2: Eine Situationsbeschreibung setzt sich aus Parametern aus den Bereichen Gesamtfahrzeug, Umgebung und den IT-Einrichtungen des Fahrzeugs zusammen

Generell können zwei Arten von Information als Situationsparameter für den Navigationsassistenten dienen: Zum einen Parameter, die möglicherweise direkten Einfluss auf die Intention des Fahrers haben könnten - schönes Wetter könnte zum Beispiel den Fahrer dazu veranlassen, einen See oder die Berge zu aufzusuchen - zum anderen Parameter, die als Begleiterscheinung bestimmter Zielinteressen zu interpretieren sind - zum Beispiel ist zu erwarten, dass während der Arbeitszeit andere Ziele als am Wochenende bevorzugt werden. Situationsparameter, die keiner der beiden Gruppen angehören, eignen sich nicht zur Klassifikation im Sinne von *Adaptive Compass* und bleiben deshalb unberücksichtigt.

Die Aufgabe der Merkmalsextraktion besteht zunächst in der Beschreibung einer „kontinuierlichen" Situation durch einen aus diskreten Zustandsvariablen $m_{\mathrm{Sit}k}$ bestehenden *Situationsvektor* $\mathbf{m}_{\mathbf{Sit}}$:

$$\mathbf{m}_{\mathbf{Sit}} = \begin{pmatrix} m_{\mathrm{Sit1}} \\ m_{\mathrm{Sit2}} \\ \vdots \\ m_{\mathrm{Sit}l} \end{pmatrix} \tag{5-1}$$

Bei den Zustandsvariablen $m_{\mathrm{Sit}k}$ zur Beschreibung eines so genannten *Situationsparameters* handelt es sich um Boole'sche Zustandsvariablen, um nicht rivalisierende Ereignisse mit dem Intentionsmodell statistisch erfassen zu können. Deshalb wird keine Zustandsvariable *Tag* verwendet mit allen sieben Tagen als Zustandsraum, sondern für jeden Tag eine eigene Zustandsvariable mit Boole'schem Zustandsraum. Kapitel 5.5 wird die Notwendigkeit dieser Maßnahme verdeutlichen.

Bei der Wahl der Situationsparameter ist ein Kompromiss zwischen detaillierter Beschreibung der Situation und Einschränkung der Komplexität zu treffen. Der Situationsvektor $\mathbf{m}_{\mathbf{Sit}}$ ist demnach stets als Näherung an die reale Situation zu sehen, zumal die Verfügbarkeit von Informationen bezüglich einer Situation, sei es aus technischen, ökonomischen oder zeitlichen Gründen unter normalen Umständen bereits zur Abstraktion zwingt.

Ein Situationsmerkmal, das nicht anhand des Situationsvektors $\mathbf{m}_{\mathbf{Sit}}$ beschrieben wird, ist die aktuelle Position des Fahrzeugs bzw. der Standort während der Bedienung des Navigationssystems. Der aktuelle Ort l_{aktuell} und sein Kreis k_{aktuell} werden separat repräsentiert. Dies trifft bei einer eindeutigen Eingabe auch auf den Zielort l_{ein} und dessen Kreis k_{ein} zu.

5.5.2 Ortsmerkmale

Eine erfolgreiche Zielort-Prädiktion setzt die Fähigkeit des Navigationsassistenten voraus, auch Aussagen über noch nicht aufgesuchte Orte treffen zu können. Um dies zu gewährleisten, ist das Intentionsmodell in der Lage, durch Analyse der bisher angefahrenen Zielorte zu ermitteln, welche charakteristischen Eigenschaften eines Ortes oder einer Gegend für den Fahrer in der jeweiligen Situation interessant sein könnten. Hierfür wurden alle Orte Deutschlands durch 28 Kenngrößen charakterisiert, die sowohl den wirtschaftlichen, industriellen Besonderheiten als auch dem Freizeitwert einer Gegend Rechnung tragen. Die wirtschaftlichen Kenngrößen könnten Aussagen über berufliche Interessen des Fahrers gestatten, Tourismus-Kenngrößen können die Erfassung von Präferenzen des Fahrers in seiner Freizeit ermöglichen.

100000 Orte einzeln zu charakterisieren ist in der Praxis nicht möglich und nicht sinnvoll, da dies zu einem äußerst komplexen Gesamtsystem führen würde. Dieses Problem kann dadurch vermieden werden, dass Orte im Rahmen der Intentionsmodellierung nicht einzeln, sondern auf Kreisebene berücksichtigt werden. Durch diese Bündelung von Orten kann Deutschland auf Basis von nur ca. 450 Kreisen beschrieben werden. Die Charakterisierung der Orte erfolgt deshalb ebenfalls auf Kreisebene. Hierzu wurden Daten des Statistischen Bundesamts verwendet.

Anhand der Daten des Statistischen Bundesamts wurde berechnet, wie charakteristisch diverse Industriezweige für die einzelnen Kreise sind. Hierzu wurde zunächst der Anteil der Erwerbstätigen in einer bestimmten Branche *KG* (*Kenngröße*) für den Kreis bestimmt. Für einen Kreis geschieht dies durch Division der Anzahl der Erwerbstätigen $n_{\mathrm{KG}_{\mathrm{Kreis}}}$ in der jeweiligen Branche durch die Gesamtzahl der Erwerbstätigen $n_{\mathrm{Gesamt}_{\mathrm{Kreis}}}$. Somit ergibt sich der Anteil der Erwerbstätigen $\mu_{\mathrm{KG}_{\mathrm{Kreis}}}$ für die Branche *KG* in dem betrachteten Kreis:

$$\mu_{\mathrm{KG}_{\mathrm{Kreis}}} = \frac{n_{\mathrm{KG}_{\mathrm{Kreis}}}}{n_{\mathrm{Gesamt}_{\mathrm{Kreis}}}} \tag{5-2}$$

Um aus dieser Information die Charakteristik einer Branche für einen Kreis abzuleiten, wird das entsprechende bundesweite Mittel, die Bezugsgröße $\mu_{\mathrm{KG}_{\mathrm{Bund}}}$, bestimmt. Der Quotient $\mu_{\mathrm{KG}_{\mathrm{Bund}}}$, der durch Division der Anzahl der bundesweit in der entsprechenden Branche *KG* Erwerbstätigen $n_{\mathrm{KG}_{\mathrm{Bund}}}$ und der Gesamtzahl der bundesweit Erwerbstätigen $n_{\mathrm{Gesamt}_{\mathrm{Bund}}}$ berechnet wird, beschreibt den Anteil der Erwerbstätigen, bezogen auf alle Kreise Deutschlands:

$$\mu_{\mathrm{KG}_{\mathrm{Bund}}} = \frac{n_{\mathrm{KG}_{\mathrm{Bund}}}}{n_{\mathrm{Gesamt}_{\mathrm{Bund}}}} \tag{5-3}$$

$\mu_{\mathrm{KG}_{\mathrm{Bund}}}$ ist der bundesweite Mittelwert, der schließlich mit dem entsprechenden Wert auf Kreisebene in Bezug zu setzen ist. Um zu bestimmen, inwiefern eine Branche *KG* für einen Kreis charakteristisch ist, wird die Streuung $\sigma_{\mathrm{KG}_{\mathrm{Kreis}}}$ berechnet:

$$\sigma_{\mathrm{KG}_{\mathrm{Kreis}}} = \mu_{\mathrm{KG}_{\mathrm{Kreis}}} - \mu_{\mathrm{KG}_{\mathrm{Bund}}} \tag{5-4}$$

Eine Branche gilt dann als charakteristisch für einen Kreis, wenn die Streuung $\sigma_{\mathrm{KG}_{\mathrm{Kreis}}}$ einen positiven Wert annimmt, d.h., wenn sich der Anteil der Erwerbstätigen in diesem Kreis in dieser Branche als überdurchschnittlich hoch erweist. Abbildung 5.3 stellt die diskretisierten Streuungswerte aller Kreise für die Kenngröße *familiäre Beherbergungsbetriebe* mittels eines Histogramms dar. Eine Vielzahl von Kreisen hat eine negative Streuung, d.h. die entsprechende Branche ist in diesen Kreisen nur unterdurchschnittlich vertreten und wird somit als nicht charakteristisch für diese Kreise interpretiert.

Aus dem Histogramm wird nun für alle Kreise abgeleitet, inwiefern die entsprechende Branche bezeichnend für diese sind. Hierfür werden die Kreise für jede Kenngröße entsprechend der Streuung bzw. ihrer Charakteristik in fünf Kategorien eingeteilt. Die fünf Kategorien werden so bestimmt, dass alle Kreise miteinander in Bezug stehen. Die höchste Kategorie kat4 wird jenen Kreisen zuge-

ordnet, die zu den 25 Prozent aller Kreise mit der größten Streuung gehören. Die übrigen Kategorien kat1 bis kat3 werden durch die 25-, 50- und 75-Prozentgrenze analog bestimmt. Zunächst werden diejenigen Streuungswerte σ_{25}, σ_{50} und σ_{75} berechnet, die der Streuung des $0{,}25 \cdot n_{\sigma>0}$-ten, $0{,}5 \cdot n_{\sigma>0}$-ten und $0{,}75 \cdot n_{\sigma>0}$-ten Kreises entsprechen, wobei es sich bei dem Faktor $n_{\sigma>0}$ um die Anzahl der Kreise mit positiver Streuung handelt. Kreise, für die die betrachtete Branche nicht bezeichnend ist, d.h. Kreise mit negativer Streuung, werden durch Kategorie kat0 beschrieben.

Abbildung 5.3 zeigt die Zuweisung der Kategorien anhand der Kenngröße *familiäre Beherbergungsbetriebe* auf Basis der diskretisierten Streuungen $\sigma_{\mathrm{KG}_{\mathrm{Kreis}}}$.

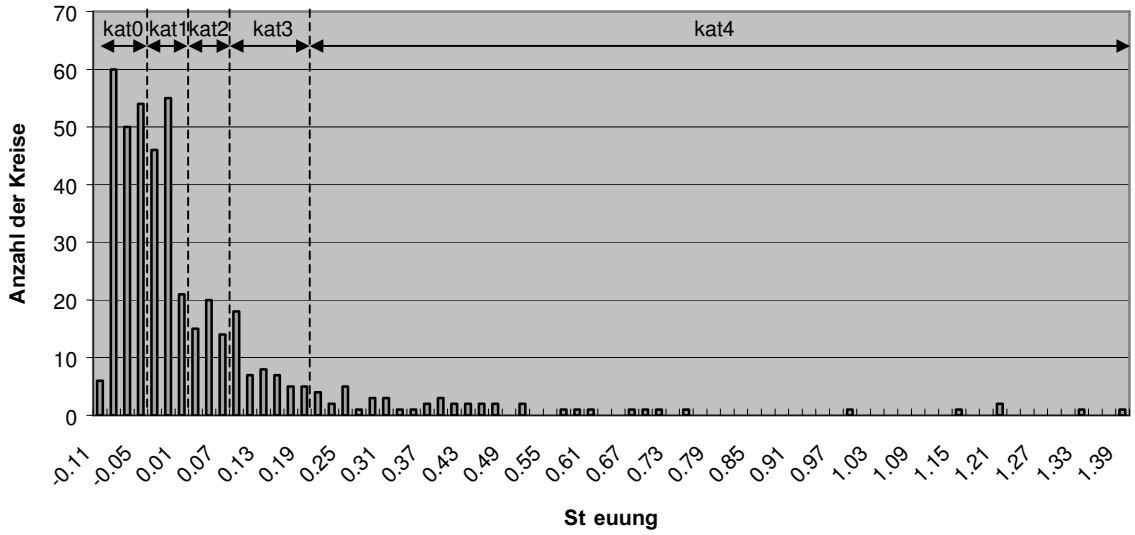


Abbildung 5.3: Diskretisierte Streuung und Kategorisierung aller Kreise für die Kenngröße *familiäre Beherbergungsbetriebe*

Zusammenfassend ergibt sich folgende Kategorisierung für eine beliebige Kenngröße *KG*:

$$kat_{\mathrm{KG,Kreis}} = \begin{cases} \text{kat0} & \text{für } \sigma_{\mathrm{KG}_{\mathrm{Kreis}}} < 0 \\ \text{kat1} & \text{für } 0 \le \sigma_{\mathrm{KG}_{\mathrm{Kreis}}} < \sigma_{25} \\ \text{kat2} & \text{für } \sigma_{25} \le \sigma_{\mathrm{KG}_{\mathrm{Kreis}}} < \sigma_{50} \\ \text{kat3} & \text{für } \sigma_{50} \le \sigma_{\mathrm{KG}_{\mathrm{Kreis}}} < \sigma_{75} \\ \text{kat4} & \text{für } \sigma_{\mathrm{KG}_{\mathrm{Kreis}}} \ge \sigma_{75} \end{cases} \tag{5-5}$$

Abbildung 5.4 visualisiert die Kategorien der Kenngröße *familiäre Beherbergungsbetriebe* für alle Kreise Deutschlands. Die weiß dargestellten Kreise entsprechen der Kategorie kat0, d.h., die betrachtete Kenngröße ist nicht bezeichnend für den Kreis. Je dunkler die Darstellung eines Kreises, desto höher die Kategorie, d.h. umso charakteristischer ist die betrachtete Branche für den entsprechenden Kreis. Die Intentionsmodellierung muss dies entsprechend berücksichtigen. Die Darstellung zeigt, welche Gegenden besonders charakteristisch für *familiäre Beherbergungsbetriebe* und damit für Tourismus in ländlichen Gegenden sind. Wie zu erwarten trifft dies auf die klassischen Erholungsgebiete wie Nord- und Ostseeküste, Alpenrand, Bayrischer Wald, Schwarzwald, etc. zu.

Dies ist ein Hinweis darauf, dass die hier vorgestellten Überlegungen zur Beschreibung von Gegenden authentisch sind.

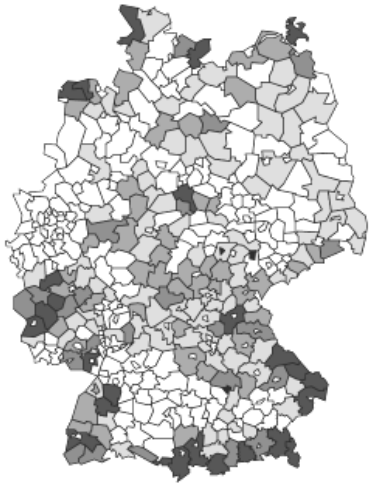


Abbildung 5.4: Visualisierung der Kategorien für *familiäre Beherbergungsbetriebe*

Die Kategorien bestimmen den Zustandsraum einer Kenngrößen-Zustandsvariable:

$$m_{\mathrm{KG}i} = \begin{cases} \mathrm{kat0} & \text{für kat0} \\ \mathrm{kat1\_4} & \text{für kat1, kat2, kat3, kat4} \end{cases} \tag{5-6}$$

Die fünf Kategorien werden nur durch zwei Zustände repräsentiert, weil die Aufgabe des Intentionsmodells darin besteht, lediglich Aussagen darüber zu treffen, ob eine bestimmte Branche für den Fahrer in der jeweiligen Situation interessant ist (kat1_4) oder nicht (kat0). Die Feingliederung in fünf Kategorien ist lediglich für das Trainingsverfahren, das in Kapitel 5.7 diskutiert wird, relevant.

Da *Adaptive Compass* 28 verschiedene Kenngrößen aus den Bereichen Industrie, Wirtschaft und Freizeit zur Charakterisierung eines Kreises berücksichtigt, dient der so genannte *Kenngrößenvektor* $\mathbf{m}_{\mathbf{KG}}$ der vollständigen Beschreibung eines Kreises:

$$\mathbf{m}_{\mathbf{KG}} = \begin{pmatrix} m_{\mathrm{KG1}} \\ m_{\mathrm{KG2}} \\ \vdots \\ m_{\mathrm{KG28}} \end{pmatrix} \tag{5-7}$$

5.5.3 Aktionsradiusmerkmale

Für viele Benutzer ist eine Abhängigkeit typischer Reisestrecken von der Situation denkbar. Privat könnte ein Fahrer beispielsweise in der Regel nur Ziele im Umkreis von 50 km um seinen Wohnort aufsuchen, während für berufliche Fahrten ein Aktionsradius von bis zu 500 km für ihn typisch sein könnte. Für einen Fahrer charakteristische, situationsabhängige Aktionsradien werden deshalb erlernt und für die Zielortprädiktion herangezogen.

Der Aktionsradius des Fahrers wird situationsabhängig modelliert, um eine potenzielle Korrelation zwischen Situation und durchschnittlicher Reisestrecke zu erfassen. Hierfür enthält der Navigationsassistent eine Datenbank mit Koordinaten aller Kreise zur Berechnung der Distanz zwischen den Kreisen des eingegebenen und des aktuellen Ortes.

Immer auf den aktuellen Standort (Koordinaten x_{aktuell}, y_{aktuell}) bezogen, wird die Entfernung zum Kreis des eingegebenen Zielorts (Koordinaten x_{Ziel}, y_{Ziel}) anhand von Gleichung (5-8) berechnet:

$$d_{\text{Ziel}} = \sqrt{(x_{\text{aktuell}} - x_{\text{Ziel}})^2 + (y_{\text{aktuell}} - y_{\text{Ziel}})^2} \tag{5-8}$$

Die berechnete Distanz wird schließlich durch einen Kreisring d diskret beschrieben. Für die Diskretisierung stehen folgende fünf Kreisringe zur Verfügung:

$$d = \begin{cases} \mathrm{d}_1 & \text{für} \quad d_{\text{Ziel}} < 50\,\text{km} \\ \mathrm{d}_2 & \text{für} \quad 50\,\text{km} \le d_{\text{Ziel}} < 100\,\text{km} \\ \mathrm{d}_3 & \text{für} \quad 100\,\text{km} \le d_{\text{Ziel}} < 250\,\text{km} \\ \mathrm{d}_4 & \text{für} \quad 250\,\text{km} \le d_{\text{Ziel}} < 500\,\text{km} \\ \mathrm{d}_5 & \text{für} \quad d_{\text{Ziel}} \ge 500\,\text{km} \end{cases} \tag{5-9}$$

Abbildung 5.5 visualisiert die fünf Kreisringe mit dem aktuellen Ort bzw. Kreis als Mittelpunkt:

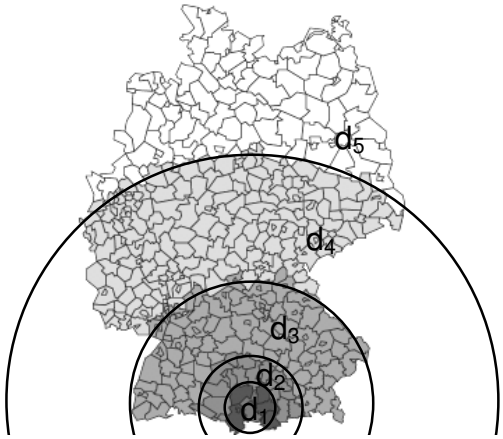


Abbildung 5.5: Kreisringe um den aktuellen Ort (München) zur Modellierung von Aktionsradien

5.5.4 Globale Präferenzen

Neben den bereits vorgestellten Merkmalstypen werden generelle, situationsunabhängige Fahrerinteressen, die *globalen Benutzerpräferenzen,* ausgewertet. Dabei handelt es sich unter anderem um den *örtlichen Schwerpunkt* des Fahrerinteresses, der dazu dient, die Gegend zu erfassen, in der sich der Fahrer bevorzugt aufhält. Grundlage hierfür ist auch in diesem Fall das Koordinatensystem, das auf Kreisebene die Positionen aller Orte beschreibt. Der Schwerpunkt entspricht dabei dem Ort, der die kürzeste Distanz zu allen bereits angefahrenen Orten besitzt. Dieser Ort kann als Mittelpunkt eines geschlossenen Polygonzugs, der durch die Koordinaten aller bereits beobachteten Zielorte

beschrieben wird, interpretiert werden. Die Lage des Schwerpunktes wird durch den Vektor **s** beschrieben, der durch das arithmetische Mittel der Vektoren $\mathbf{k}_S$, die zur Definition der Positionen aller bereits aufgesuchten L Orte bzw. Kreise dienen, bestimmt wird. Somit folgt für die Berechnung des Schwerpunkts der Vektor **s**:

$$\mathbf{s} = \begin{pmatrix} x_S \\ y_S \end{pmatrix} = \frac{1}{L} \sum_{l=1}^{L} \mathbf{k}_{Sl} \tag{5-10}$$

Abbildung 5.6 zeigt ein Beispiel für den Schwerpunktvektor **s** für einen Fahrer, dessen Interesse vor allem dem süddeutschen Raum gilt. Die dunklen Kreise wurden bereits vom Fahrer aufgesucht, die hellen Kreise noch nicht.

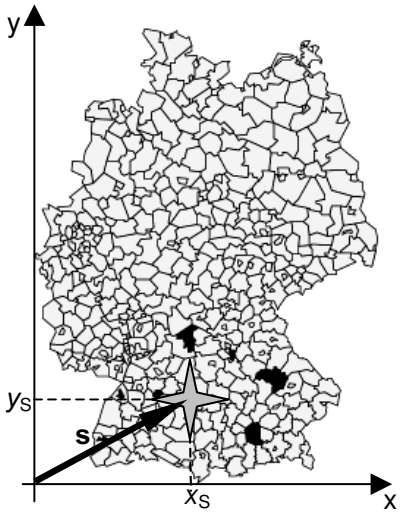


Abbildung 5.6: Schwerpunktvektor **s** zur Erfassung bevorzugter Gegenden

Neben dem Schwerpunktvektor werden noch alle in den informationstechnischen Geräten zu findenden Einträge, die auf Interessen des Fahrers bezüglicher bestimmter Orte schließen lassen, berücksichtigt. Dies betrifft Einträge der Telefon-History, des Adressbuchs und des Kalenders des elektronischen Organizers bzw. Handheld-Computers (PDA). Diese Orte werden auf Kreisebene durch den Vektor $\mathbf{k}_{IT}$ beschrieben.

5.5.5 Überblick über den Merkmalsvektor

Abbildung 5.7 stellt die möglichen Konstellationen des Merkmalsvektors **m** noch einmal schematisch dar. Im Trainingsmodus, d.h., falls ein eindeutig interpretierbarer Zielort eingegeben wurde, besteht der Merkmalsvektor aus dem Situationsvektor $\mathbf{m}_{Sit}$, der alle für das Intentionsmodell relevanten Faktoren einer Situation mittels diskreter Zustandsvariablen erfasst. Der eingegebene Ort und dessen Kreis sind ebenfalls Teil des Merkmalsvektors. Darüber hinaus wird dieser Ort anhand des Kenngrößenvektors $\mathbf{m}_{KG}$ hinsichtlich seines wirtschaftlichen Potenzials und des Freizeitwerts beschrieben. Die Entfernung d des Standorts zum eingegebenen Ort wird ebenfalls berücksichtigt. Globale Präferenzen sind nicht Teil des Merkmalsvektors für das Training des Intentionsmodells; diese Komponente ist dennoch Teil der Merkmalsextraktion, da der Schwerpunktvektor bezüglich des eingegebenen Zielortes aktualisiert wird.

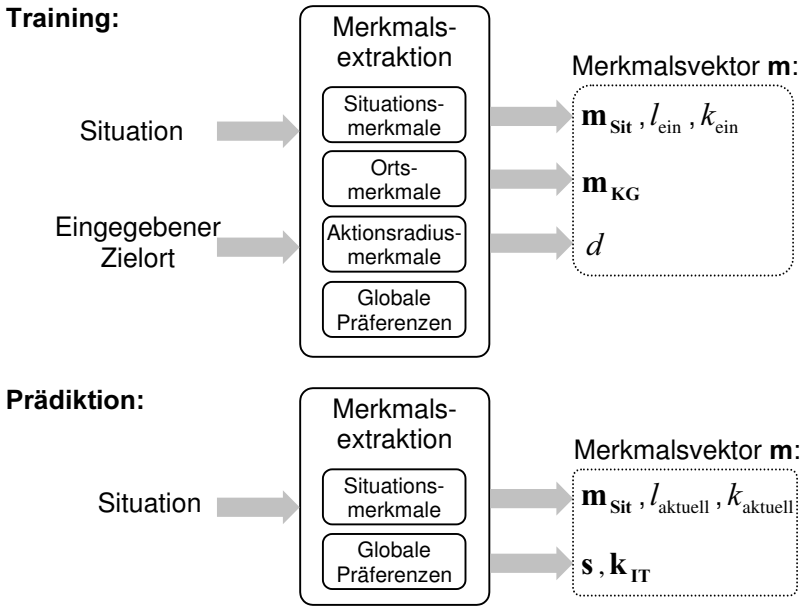


Abbildung 5.7: Der Merkmalsvektors **m** für das Training des Intentionsmodells und für die Prädiktion der Benutzerintention

Eine mehrdeutige Zieleingabe aktiviert den Modus Zielortprädiktion. Die Aufgabe des Navigationsassistenten besteht dann in der Beurteilung aller Intentionshypothesen mit Hilfe des in den Intentionsmodellen gespeicherten Wissens über situative Benutzerpräferenzen. Entsprechend enthält der Merkmalsvektor **m** keinerlei Informationen über den tatsächlichen unbekannten Zielort. Die globalen Präferenzen sind nun Teil des Vektors, da sie für die Zielortprädiktion herangezogen werden.

5.6 Intentionsmodell

Wie in Kapitel 2 diskutiert dient das Intentionsmodell dazu, einen syntaktisch-semantischen Zusammenhang zwischen den Intentionshypothesen und dem Merkmalsvektor **m** herzustellen. In diesem Kapitel wird zunächst die allgemeine Struktur des Intentionsmodells von *Adaptive Compass* und schließlich die konkrete Umsetzung mittels Bayes'scher Netze behandelt.

5.6.1 Struktur des Intentionsmodells

In *Adaptive Compass* werden alle Intentionshypothesen anhand eines einzigen Intentionsmodells repräsentiert. Die Struktur des Intentionsmodells orientiert sich an der Aufgabenstellung. Für die Disambiguierung und die Begrenzung des Ortsnamenvokabulars dient die in Abbildung 5.8 links dargestellte Struktur. Die Struktur für die intelligente Bestimmung des Default-Zielorts ist rechts dargestellt.

Die Ebenen des Intentionsmodells werden nun einzeln vorgestellt:

- **Intentionsebene:** Die Intentionsebene lässt wahrscheinlichkeitsbasierte Aussagen über die Einträge der Intentionsbibliothek zu. Wie in Kapitel 5.3 bereits erwähnt sind die Intentionshypothesen durch den jeweiligen Betriebsmodus von *Adaptive Compass* bestimmt.

- **Syntaktisch-semantische Ebene:** Auf dieser Ebene werden Korrelationen zwischen den Situationsparametern und den für den Fahrer interessanten Ortseigenschaften modelliert. Analog wird mit dem Aktionsradius verfahren. Diese Informationen gestatten schließlich im Rahmen der intentionsbasierten Interpretation situative Aussagen über die Interessen des Fahrers und ermöglichen somit Aussagen über den Wunschzielort.

  Die syntaktisch-semantische Ebene spielt für die Bestimmung des Default-Zielorts keine Rolle, da in diesem Fall der Bezug zwischen Merkmalen und Intentionshypothesen direkt hergestellt wird und Aussagen über Zielkreise zu unkonkret wären.

- **Merkmalsebene:** Die Merkmalsebene ist die Schnittstelle des Intentionsmodells zur Merkmalsextraktion. Für die Prädiktion der Benutzerintention dienen die Knoten dieser Ebene der Modellierung des Situationsvektors.

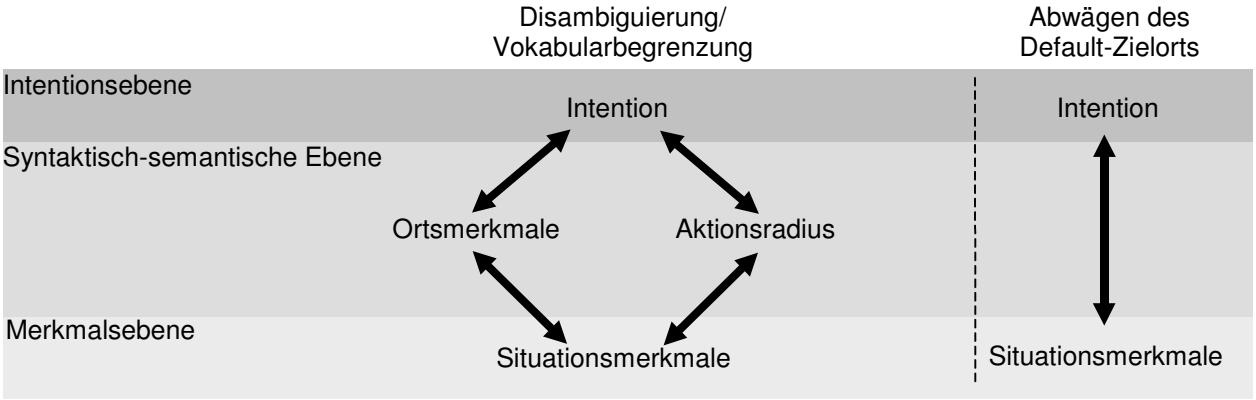


Abbildung 5.8: Struktur des Intentionsmodells von *Adaptive Compass*

Die konkrete Umsetzung des *Adaptive Compass*-Intentionsmodells anhand Bayes'scher Netze wird im folgenden Abschnitt vorgestellt.

5.6.2 Realisierung des Intentionsmodells durch Bayes'sche Netze

Für die Realisierung des Intentionsmodells werden drei verschiedene Typen von Bayes'schen Netzen verwendet: *Kenngrößennetze*, das *Distanznetz* und das *Defaultnetz*. Diese Netze werden nun vorgestellt:

- **Kenngrößennetz**
  Um einen Zusammenhang zwischen Situation und Zielort herzustellen, wird die Korrelation der aktuellen Situation mit den Kenngrößen, die diesen Ort beschreiben, ermittelt. Da für jede Kenngröße ein Bayes'sches Netz nötig ist, ergeben 28 Kenngrößen entsprechend viele so genannte *Kenngrößennetze*. Um Speicher- und Rechenaufwand für den Trainings- sowie für den Prädiktions-Modus zu minimieren, wurden die Strukturen der Netze möglichst einfach gehalten. Abbildung 5.9 zeigt die Struktur eines Kenngrößennetzes.

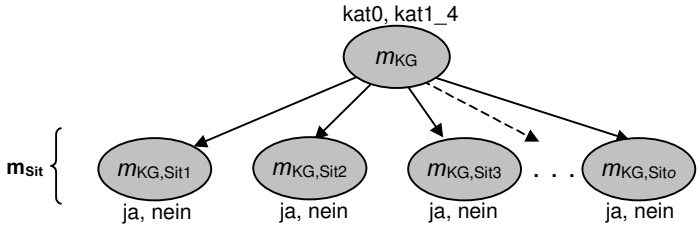


Abbildung 5.9: Topologie eines Kenngrößennetzes

Die Struktur entspricht einem einfachen Maximum-a-posteriori-Klassifikator. Bei einem Kenngrößennetz besteht der Zustandsraum des Wurzelknotens aus den möglichen Zuständen der entsprechenden Kenngröße-Zustandsvariable, also aus den in Kapitel 5.5.2 diskutierten erwähnten Zuständen kat0 und kat1_4. Die a-priori-Wahrscheinlichkeit $P(m_{KG})$ des Wurzelknotens wird neutral gewählt (0,5; 0,5), da ohne Einfluss der Situation keine qualitativen und quantitativen Aussagen über die Fahrerpräferenzen möglich sind.

Die Merkmalsknoten visualisieren Boole'sche Variablen, deren bedingte Wahrscheinlichkeiten $P(m_{KG,Sit} \mid m_{KG})$ zu Beginn ebenfalls neutral definiert werden; im Rahmen des Trainingsmodus werden sie schließlich auf jeden neuen Datensatz adaptiert. Die Merkmalsknoten des Kenngrößennetzes modellieren nur die Situationsparameter von $\mathbf{m_{Sit}}$, die für die Erfassung einer Korrelation von Situation und Fahrerinteressen relevant sind. Informationen bezüglich der letzten Betankung sind zwar für die Bestimmung des typischen Aktionsradius interessant, lassen aber keinerlei Aussage über bevorzugte Ortsmerkmale zu.

Soll ein Situationsaspekt auf eine Zustandsvariable abgebildet werden, so wird diese Variable auf den Zustand „ja" gesetzt, d.h., im Sommer wird die Zustandsvariable *Sommer* gleich „ja" gesetzt, die Zustandsvariablen der anderen Jahreszeiten bleiben unberücksichtigt. Sie werden behandelt, als läge keine Information bezüglich dieses Situationsparameters bzw. Merkmalsknotens vor. Grund dafür ist, dass der Fahrer beispielsweise im Sommer Orte mit bestimmten Eigenschaften bevorzugt. Dies lässt jedoch keinen Schluss auf die Fahrerpräferenzen in den übrigen Jahreszeiten zu, obwohl es sich um komplementäre Situationsparameter handelt.

Die Struktur spiegelt die Philosophie von *Adaptive Compass* wieder, den Zusammenhang jedes einzelnen Situationsparameters mit einer Kenngröße zu modellieren. Dadurch können auch bisher nicht beobachtete Situationen urteilt werde. Dieser Aspekt ist von erheblicher Bedeutung, da somit der „Einschwingvorgang" des Systems maßgeblich reduziert wird, da nicht exakt jede Situation beobachtet werden muss, bevor die bedingten Wahrscheinlichkeiten der Merkmalsknoten gegen repräsentative Werte konvergieren. Der Einfluss einer Situation wird auf alle beobachteten Situationsparameter gleichmäßig verteilt.

- **Distanznetz**

  Der zweite Netztyp ist das *Distanznetz*, dessen Aufgabe die Erfassung der Korrelation zwischen Situationsparametern und der Entfernung des Zielorts vom aktuellen Standort ist. Somit wird der Aktionsradius eines Fahrers situationsadaptiv trainiert bzw. vorhergesagt. Dieses Modell trägt der Annahme Rechnung, dass typische Aktionsradien situationsabhängig sind. Hierfür wird die

Entfernung eines Ortes bzw. Kreises zur aktuellen Position beschrieben, indem um den aktuellen Kreis Kreisringe gezogen werden.

Die Abbildung 5.10 zeigt die Topologie des Distanznetzes; sie ist mit Ausnahme des Wurzelknotens *d* und der Auswahl an Situationsparametern analog zu der Struktur eines Kenngrößennetzes. Zur Modellierung der in Kapitel 5.5.3 behandelten fünf Kreisringe hält der Zustandsraum des Wurzelknotens je einen Zustand bereit.

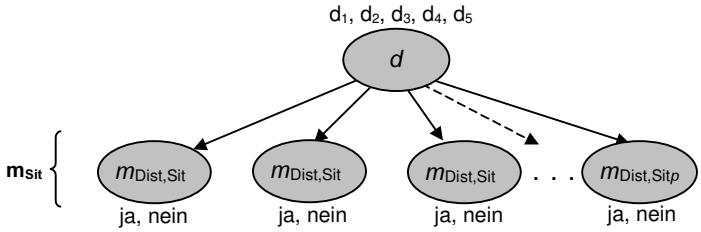


Abbildung 5.10: Topologie des Distanznetzes

- **Defaultnetz**
  Die Topologie des Defaultnetzes (Abbildung 5.11) wurde ebenfalls analog zu den Kenngrößennetzen gewählt. Die Aufgabe des Defaultnetzes besteht darin, den in der jeweiligen Situation wahrscheinlichsten Zielort konkret zu bestimmen. Da die Intentionsbibliothek in diesem Fall aus allen bereits aufgesuchten Orten besteht, hält der Zustandsraum des Wurzelknotens für jeden schon einmal angefahrenen Ort einen Zustand bereit.

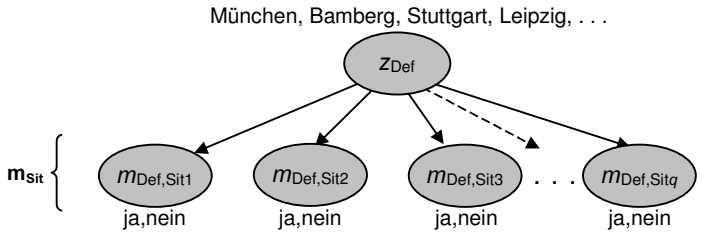


Abbildung 5.11: Topologie des Defaultnetzes

Die a-priori-Wahrscheinlichkeit $P(z_{\mathrm{Def}})$ wird für jeden Ort bzw. Zustand identisch gewählt, sodass zunächst alle Hypothesen als gleich wahrscheinlich gewertet werden. Mit jedem neuen Zielort muss der Zustandsraum des Wurzelknotens erweitert werden.

Die unterschiedlichen Namen der Merkmalsknoten der verschiedenen Netze ($m_{\mathrm{KG,Sit}}$, $m_{\mathrm{Dist,Sit}}$, $m_{\mathrm{Def,Sit}}$) wurden für die obigen Abbildungen gewählt, um zu betonen, dass die Netze unterschiedliche Situationsparameter berücksichtigen können. Für jeden Netztyp sind unterschiedliche Elemente des Situationsvektors von Interesse. Im weiteren Verlauf wird eine Abfolge von Situationsvektoren nur pauschal mit dem Situationsvektor $\mathbf{m}_{\mathbf{Sit}}$ beschrieben und die auf den Netztyp angepasste Nomenklatur nicht weiter verwendet.

In diesem Abschnitt wurde auf die Strukturen der verwendeten Bayes'schen Netze eingegangen. Der folgende Abschnitt befasst sich mit dem Training der bedingten Wahrscheinlichkeiten der Merkmalsknoten dieser Netze.

5.7 Training des Intentionsmodells

Das Intentionsmodell befindet sich im Trainingsmodus, sobald der Fahrer einen eindeutigen Zielort eingegeben hat. Die Merkmalsextraktion kann in diesem Fall dem Intentionsmodell alle Informationen zur situativen Adaption der bedingten Wahrscheinlichkeiten an die Fahrerinteressen zur Verfügung stellen. Abbildung 5.12 gibt einen Überblick über das Online-Training des Intentionsmodells bzw. der Bayes'schen Netze. Das Intentionsmodell erhält die Beschreibung der aktuellen Gesamtsituation $\mathbf{m}_{\mathbf{Sit}}$, die Charakterisierung des eingegebenen Ortes $\mathbf{m}_{\mathbf{KG}}$, die Entfernung d dieses Zielorts zur aktuellen Position sowie den eingegebenen Zielort l_{ein} als Datensatz. Das Ziel des Trainings besteht in der iterativen Adaption der bedingten Wahrscheinlichkeiten der Situationsparameter $P(m_{\text{Sit}} \mid m_{\text{KG}})$ für die Kenngrößennetze, $P(m_{\text{Sit}} \mid d)$ für das Distanznetz und $P(m_{\text{Sit}} \mid z_{\text{Def}})$ für das Defaultnetz an diesen neuen Datensatz unter Berücksichtigung alle bisherigen Beobachtungen bzw. Datensätze.

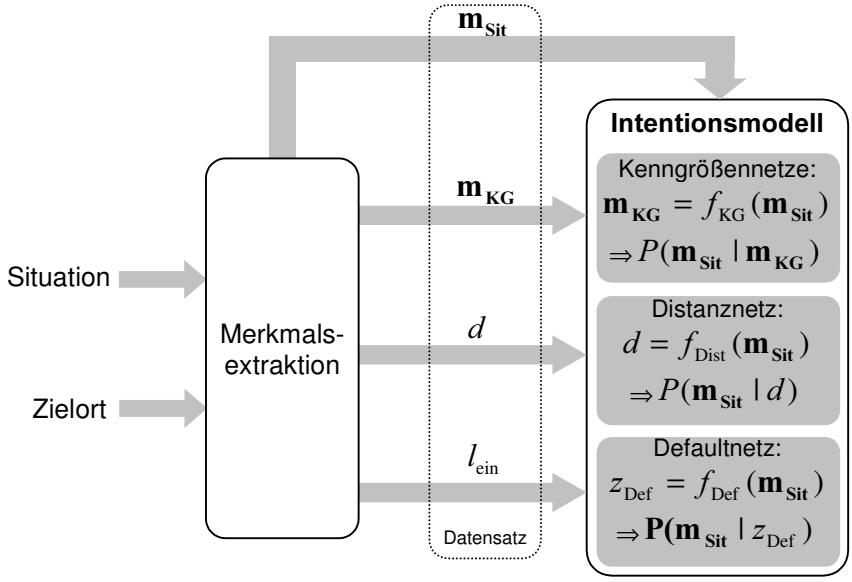


Abbildung 5.12: Überblick über die Adaption des Intentionsmodells an einen neuen Datensatz

Um eine hohe Effizienz und damit niedrigen Rechenaufwand zu erreichen, wurde das Trainingsverfahren speziell für die zum Einsatz kommenden Topologien der Bayes'schen Netze konzipiert. Zum besseren Verständnis des Algorithmus wird zunächst die Nomenklatur einiger Größen definiert. Bei Wahrscheinlichkeiten $P_n(\cdot)$ deutet der Index n darauf hin, dass die entsprechenden Werte aus allen bisher beobachteten n Datensätzen berechnet wurden. Mit dem Index „n+1" wird angedeutet, dass die Wahrscheinlichkeit $P_{n+1}(\cdot)$ sowohl alle bisherigen n Beobachtungen als auch den neuesten Datensatz repräsentiert. Im Rahmen der Iterationsschleife sich verändernde Wahrscheinlichkeiten werden mit $P_{n,i}(\cdot)$ bezeichnet.

Die grundlegende Idee des Trainings- bzw. Adaptionsverfahrens wird anhand von Abbildung 5.13 vorgestellt. Links ist ein schematisiertes Netzwerk, bestehend aus dem Wurzelknoten r und den Merkmalsknoten zur Modellierung der Situationsparameter, dargestellt. Der Wurzelknoten des Netzes, auf das sich die Abbildung bezieht, wird allgemein mit r bezeichnet, um deutlich zu machen,

dass dieses Verfahren für alle im Rahmen des Intentionsmodells zum Einsatz kommenden Netze verwendet wird.

Das Netz ist zunächst auf alle bisherigen Beobachtungen, auf *n* Datensätze trainiert. Dies äußert sich durch eine a-priori-Wahrscheinlichkeit $P_n(r)$ und durch die bedingten Wahrscheinlichkeiten der Merkmalsknoten $P_n(m_{\text{Sit}k} \mid r)$. Der Istzustand des Systems berechnet sich aus der a-posteriori-Wahrscheinlichkeit $P_n(r \mid \mathbf{m}_{\mathbf{Sit}})$, die situationsabhängige Aussagen über die durch den Zustandsraum des Wurzelknotens modellierten Ereignisse erlaubt.

Wird nun ein neuer Datensatz beobachtet, so müssen $P_n(r)$ und $P_n(m_{\text{Sit}k} \mid r)$ derart verändert werden, dass sie sowohl alle bisherigen *n* Datensätze als auch den neuen Datensatz repräsentieren, d.h. die Wahrscheinlichkeiten müssen so gewählt werden, dass sie einen neuen Sollzustand $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$ des Netzes ermöglichen. Dies geschieht durch ein iteratives Lernverfahren.

Zunächst wird die neue a-priori-Wahrscheinlichkeit des Wurzelknotens $P_{n+1}(r)$ bestimmt und schließlich die bedingten Wahrscheinlichkeiten der Merkmalsknoten $P_{n+1}(m_{\text{Sit}k} \mid r)$. Da eine geeignete Kombination aus $P_{n+1}(r)$ und $P_{n+1}(m_{\text{Sit}k} \mid r)$ den Sollzustand des Bayes'schen Netzes realisiert, verhält sich der Einfluss der Situation auf den Wurzelknoten proportional zur Differenz von $P_{n+1}(r)$ und $P_n(r)$. Setzt man $P_{n+1}(r)$ gleich $P_n(r)$, so können ausschließlich die Merkmalsknoten den neuen Sollzustand ermöglichen, d.h., der Einfluss der Situation wird maximiert. Darauf wird zu einem späteren Zeitpunkt noch Bezug genommen.

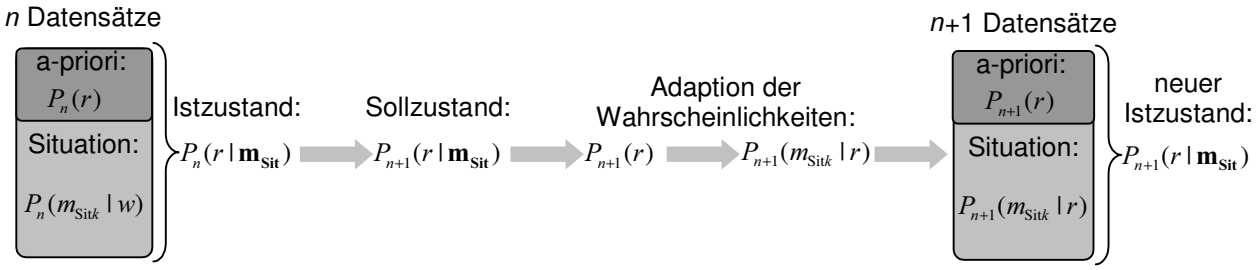


Abbildung 5.13: Grundidee des Adaptionsverfahrens

Für das Adaptieren der bedingten Wahrscheinlichkeiten an einen neuen Datensatz ist entscheidend, dass die Merkmalsknoten nicht separat betrachtet werden. Alle relevanten Situationsparameter tragen einen bestimmten Teil zu einer Gesamtsituation bei. Die Beschreibung einer Situation wird auf diese Weise auf alle relevanten Merkmalsknoten verteilt. Da ein Merkmalsknoten ausschließlich von seinem Wurzelknoten statistisch abhängig modelliert ist, ermöglichen die Topologien der Bayes'schen Netze eine schnellere Konvergenz des Intentionsmodells an repräsentative Wahrscheinlichkeiten und somit Aussagen über Intentionshypothesen in noch nicht beobachteten Situationen.

Die folgende Abbildung gibt einen Überblick über das iterative Trainings- bzw. Adaptionsverfahren eines Bayes'schen Netzes im Rahmen des *Adaptive-Compass*-Intentionsmodells:

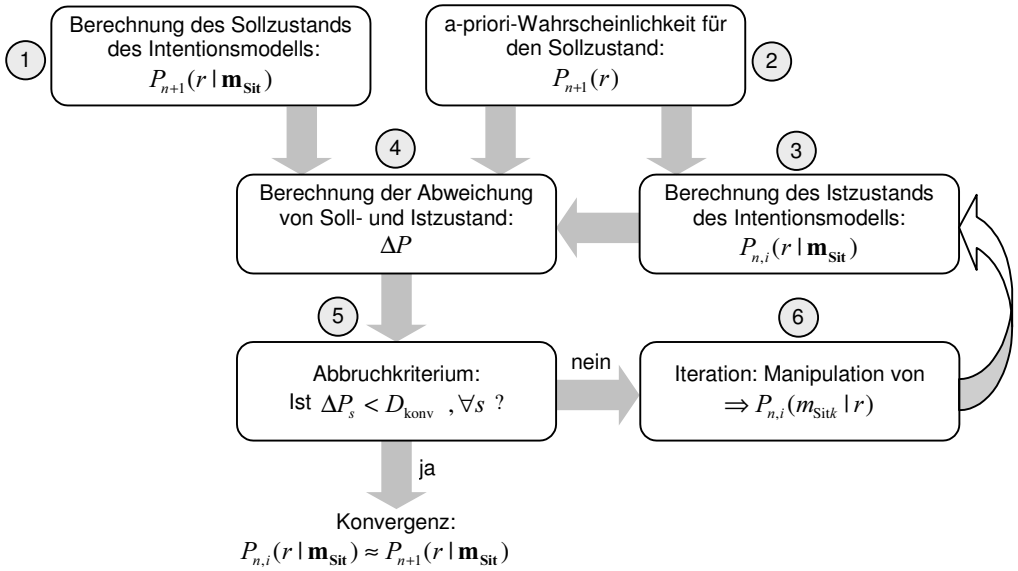


Abbildung 5.14: Ablauf des iterativen Adaptionsverfahrens

Das Verfahren wird nun in den der Abbildung dargestellten Schritten ① bis ⑥ beschrieben:

① Sobald ein neuer Datensatz zur Verfügung steht, wird der zur Repräsentation der neuen und aller bisherigen Beobachtungen erwünschte Sollzustand des Intentionsmodells berechnet. Allgemein handelt es sich dabei um die a-posteriori-Wahrscheinlichkeit $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$, die bei bekannter Situation Aussagen über die Benutzerpräferenzen ermöglicht. Dabei ist zunächst festzulegen, wie stark der neue Datensatz bezüglich aller bisherigen Datensätze zu gewichten ist. Folgende Abbildung 5.15 illustriert die Grundidee:

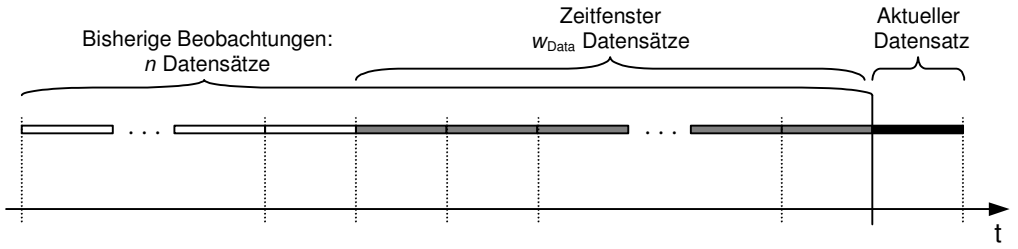


Abbildung 5.15: Gewichtung des neuen Datensatzes gegenüber bisherigen Beobachtungen

Die Vergangenheit wird durch *n* Datensätze repräsentiert, die nicht explizit vorliegen, sondern anhand der Wahrscheinlichkeit $P_n(r \mid \mathbf{m}_{\mathbf{Sit}})$ den aktuellen Modellzustand bestimmen. Die Bestimmung des an den aktuellen Datensatz adaptierten Systemzustands $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$ erfolgt ausgehend von der Systemvergangenheit $P_n(r \mid \mathbf{m}_{\mathbf{Sit}})$ auf Basis eines Zeitfensters, das die letzten w_{Data} Datensätze berücksichtigt. Die Größe des Zeitfensters w_{Data} bestimmt die Gewichtung des neuen Datensatzes gegenüber der Systemvergangenheit. Der Sollzustand des adaptierten Intentionsmodells wird anhand folgender Gleichung berechnet:

$$P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}}) = \frac{P_{\text{Data}}(r) + w_{\text{Data}} \cdot P_n(r \mid \mathbf{m}_{\mathbf{Sit}})}{1 + w_{\text{Data}}} \tag{5-11}$$

Die Wahrscheinlichkeit $P_{\text{Data}}(r)$ ist direkt dem neuen Datensatz zu entnehmen. Die Werte entsprechen, je nach Zustand, entweder dem Wert 0 oder 1, da von einer sicheren Information ausgegangen

werden kann; grundsätzlich können auf diese Weise jedoch auch unsichere Datensätze verarbeitet werden, zum Beispiel bei wahrscheinlichkeitsbasierter Abwägung einer Ortskategorie im Falle einer fremden, in den Datenbanken nicht modellierten Gegend. Die Berechnung des Sollzustands des Intentionsmodells geschieht auf Basis von Häufigkeiten; dabei dient der zweite Summand des Zählers der Rekonstruktion der Zeitfenster-Datensätze. Ingesamt werden $1+w_{\text{Data}}$ Datensätze (Nenner) ausgewertet, der neue Datensatz und die Daten des Zeitfensters.

Der Einfluss des Zeitfensters gleicht einer Tiefpassfilterung. Je größer das Zeitfenster bzw. der Gewichtungsfaktor w_{Data}, desto geringer ist die Auswirkung eines neuen Datensatzes auf den Systemzustand $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$. Wird der Gewichtungsfaktor gleich Null gesetzt, adaptiert sich das Modell auf den neuen Datensatz, ohne die Systemvergangenheit mit einzubeziehen; in diesem Fall verhält sich das Modell gedächtnislos.

Abbildung 5.16 zeigt einige Adaptionskurven am konkreten Beispiel eines Kenngrößennetzes. Dargestellt sind Wahrscheinlichkeitswerte $P_{n+1}(m_{\text{KG}} = \text{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}})$ für unterschiedliche Gewichtungsfaktoren. Der Ausgangswert von 0,5 entspricht der a-priori-Wahrscheinlichkeit $P(m_{\text{KG}} = \text{kat1\_4})$, d.h., es wurden noch keine Beobachtungen verarbeitet. Die ersten zehn Datensätze stehen für jeweils einen eingegebenen Ort mit überdurchschnittlicher Ausprägung der betrachteten Branche; der Wurzelknoten des Intentionsmodells nimmt somit den Zustand kat1_4 an. Um das Adaptionsverhalten des eingeschwungenen Modells auf konträre Bedingungen zu dokumentieren, wurde der Wurzelknoten für die Datensätze 11 bis 20 auf den Zustand kat0 gesetzt. Somit lässt sich eine deutliche Änderung der Fahrerpräferenzen simulieren.

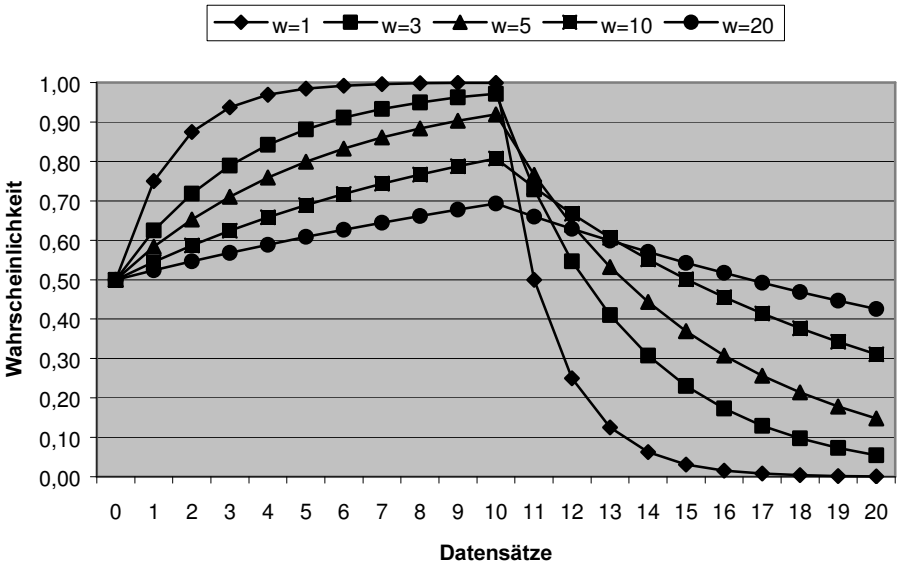


Abbildung 5.16: Abhängigkeit der Adaption an neue Datensätze von verschiedenen Gewichtungsfaktors w_{Data}

Die Adaptionskurve mit niedrigen Gewichtungsfaktoren ($w_{\text{Data}} = 1$) konvergiert bereits nach wenigen Datensätzen gegen den a-priori-Wahrscheinlichkeitswert des Datensatzes $P_{\text{Data}}(m_{\text{KG}} = \text{kat1\_4})$. Dieser Fall steht für die Annahme des Modells, dass der Fahrer in der entsprechenden Situation sicher Orte mit überdurchschnittlicher Ausprägung der betrachteten Branche aufsucht. Nach Adap-

tion auf den Datensatz 11 verändert sich der Wahrscheinlichkeitswert derart, dass mit diesem neuen Datensatz der Einfluss aller vorherigen Beobachtungen neutralisiert wird. Entsprechend schnell konvergiert die Adaptionskurve für die folgenden Datensätze.

Generell passt sich das Intentionsmodell bei Training mit niedrigen Gewichtungsfaktoren w_{Data} neuen Datensätzen zu schnell an. Das Gegenteil ist der Fall bei zu hohen Gewichtungsfaktoren, d.h. bei zu starker Berücksichtigung der Modellvergangenheit. Die Lernkurve verläuft dann zu flach, um eine sinnvolle Adaption zu gewährleisten.

Ein sinnvolles Lernverhalten eines Intentionsmodells lässt sich erzielen, wenn der Gewichtungsfaktor im Bereich zwischen 3 und 10 gewählt wird. Bei Kenngrößennetzen erfolgt die Wahl von w_{Data} dynamisch, abhängig von der Kenngrößenkategorie. Orte mit hohen Kategorien, d.h. mit hoher Charakteristik bezüglich einer Kenngröße, werden für das Training durch einen niedrigeren Faktor w_{Data} im Verhältnis zu vorherigen Datensätzen stärker gewichtet. Weniger charakteristische Orte werden entsprechend schwächer gewichtet. Folgende Einteilung wurde gewählt:

$$w_{\text{Data}} = \begin{cases} 5 & \text{für} \quad \text{kat0}, \text{kat1}, \text{kat2} \\ 4 & \text{für} \quad \text{kat3} \\ 3 & \text{für} \quad \text{kat4} \end{cases} \tag{5-12}$$

Abbildung 5.17 zeigt den typischen Verlauf einer Lernkurve eines Kenngrößennetzes:

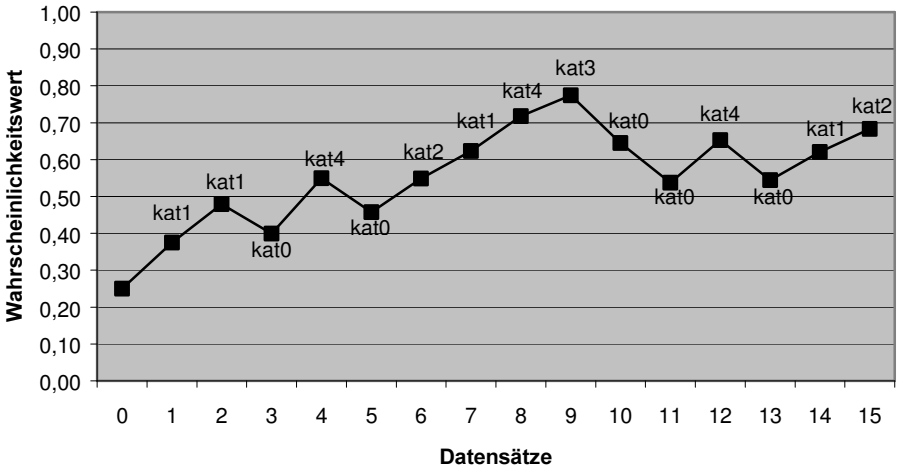


Abbildung 5.17: Beispiel einer Lernkurve eines Kenngrößennetzes mit dynamischer Wahl des Gewichtungsfaktors w_{Data}

Dargestellt ist der Wahrscheinlichkeitswert $P_{n+1}(m_{\text{KG}} = \text{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}})$ in Abhängigkeit von einer Reihe von Beobachtungen bzw. Datensätzen. Die Kategorien der betrachteten Ortskenngröße ist für jeden neu eingegebenen Ort dargestellt. Ausgehend von dem ursprünglichen Wert 0,25 passt sich $P_{n+1}(m_{\text{KG}} = \text{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}})$ mit dynamischer Wahl des Gewichtungsfaktors w_{Data} an die neuen Datensätze an. Deutlich erkennbar ist, dass Datensätze mit Kategorie kat0 die Wahrscheinlichkeit für den Zustand kat1_4 senken. Die verwendeten Datensätze simulieren den Fall, dass der Fahrer ein

Interesse für Orte mit besonderer Ausprägung der betrachteten Branche entwickelt, sporadisch sind aber Orte ohne diese Eigenschaft interessant.

Für das Distanznetz und das Defaultnetz wurde ein konstanter Gewichtungsfaktor w_{Data} von 5 gewählt um alle Kreisringe bzw. alle Default-Zielalternativen einheitlich zu behandeln.

② Die Berechnung der Soll-a-priori-Wahrscheinlichkeit $P_{n+1}(r)$ geschieht in Analogie zu der beschriebenen Bestimmung des Gesamt-Sollzustands:

$$P_{n+1}(r) = \frac{P_{\text{Data}}(r) + w_r \cdot P_n(r)}{1 + w_r} \tag{5-13}$$

Dieser Schritt wird ausschließlich für das Defaultnetz verwendet, da sich die Bestimmung des Default-Zielorts an Pendler richtet, die regelmäßig eine bestimmte Auswahl an Orten anfährt. Anhand von $P_{n+1}(r)$ gehen die Häufigkeiten aller bereits angefahrenen Zielorte mit in die Überlegungen ein. Diese trägt vor allem dazu bei, bei noch wenigen Datensätzen sinnvolle Aussagen über den Default-Zielort zu treffen. Je mehr Datensätze situative Aussagen ermöglichen, desto mehr tritt der Einfluss von $P_{n+1}(r)$ in den Hintergrund. Ermöglicht wird dies durch einen relativ schwachen Einfluss einer neuen Beobachtung, der durch einen hohen Gewichtungsfaktor w_{r} von 40 erreicht wird.

Bei Kenngrößennetzes und dem Distanznetz wird auf diesen Schritt verzichtet, somit bleibt die Soll-a-priori-Wahrscheinlichkeit konstant:

$$P_{n+1}(r) = P_n(r) = const \tag{5-14}$$

Dadurch erfolgen Aussagen über Fahrerpräferenzen bezüglich bestimmter Ortseigenschaften und Reisedistanzen ausschließlich situationsabhängig.

③ Nach der Berechnung des Sollzustands des Kenngrößennetzes $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$ in Schritt ① wird der Istzustand nach *i* Iterationen $P_{n,i}(r \mid \mathbf{m}_{\mathbf{Sit}})$ bestimmt. Vor der ersten Iteration entspricht diese a-posteriori-Wahrscheinlichkeit den Werten $P_n(r \mid \mathbf{m}_{\mathbf{Sit}})$, für jede weitere Iteration muss sie neu berechnet werden.

④ Um den Istzustand mit dem Sollzustand zu vergleichen, wird die Differenz der entsprechenden Wahrscheinlichkeiten für alle Zustände der Zustandsvariable r gebildet. Nach Betragsbildung erhält man schließlich die Abweichung der beiden Systemzustände ΔP:

$$\Delta P = \left| P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}}) - P_{n,i}(r \mid \mathbf{m}_{\mathbf{Sit}}) \right| \tag{5-15}$$

⑤ Sobald die Differenzwerte ΔP_s aller s Zustände eine Toleranzgrenze D_{konv} unterschreiten, deutet dies auf eine ausreichende Ähnlichkeit zwischen Ist- und Sollzustand hin. In diesem Fall konvergiert die Wahrscheinlichkeit $P_{n,i}(r \mid \mathbf{m}_{\mathbf{Sit}})$ auf Grund der Adaption gegen den Sollwert $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$. Somit ist das Abbruchkriterium der Iterationsschleife erfüllt und das Bayes‘sche Netz approximativ an den neuen Datensatz adaptiert.

Bei der Wahl der Konstanten D_{konv} ist dabei ein Kompromiss zwischen Genauigkeit und niedriger Anzahl von Iterationsschritten bzw. Geschwindigkeit des Adaptionsprozesses zu treffen. Untersuchungen ergaben eine sinnvolle Toleranz von 5 Prozent ($D_{\text{konv}} = 0{,}05$).

Bei nicht erfülltem Abbruchkriterium der Iterationsschleife werden die Wahrscheinlichkeitswerte sukzessive verändert.

⑥ Die Anpassung des Istzustands $P_{n,i}(r \mid \mathbf{m}_{\mathbf{Sit}})$ an den Sollzustand $P_{n+1}(r \mid \mathbf{m}_{\mathbf{Sit}})$ wird durch iteratives Verändern der bedingten Wahrscheinlichkeiten aller Merkmalsknoten $P_n(m_{\text{Sit}} \mid r)$, die den Zustand „ja“ einnehmen, vorgenommen. Die übrigen Merkmalsknoten sind für die Adaption nicht relevant.

Die durch Iteration veränderte bedingte Wahrscheinlichkeit des *k*-ten instanziierten Situationsparameters wird nun mittels $P_{n,i}(m_{\text{Sit}k} \mid r)$ beschrieben. Für eine adäquate Modifikation der bedingten Wahrscheinlichkeiten aller relevanten Situationsparameter dient die zu Gleichung (5-12) analoge, auf den *k*-ten instanziierten Situationsparameter bezogene folgende Gleichung:

$$P_{n,i}(m_{\text{Sit}k} \mid r) = \frac{P_{\text{Data}}(m_{\text{Sit}k}) + w_{\text{Sit}} \cdot P_{n,i-1}(m_{\text{Sit}k} \mid r)}{1 + w_{\text{Sit}}} \tag{5-16}$$

Für den ersten Iterationsschritt entspricht $P_{n,0}(m_{\text{Sit}k} \mid r)$ der ursprünglichen, unveränderten Wahrscheinlichkeit $P_n(m_{\text{Sit}k} \mid r)$. $P_{\text{Data}}(m_{\text{Sit}k})$ ist direkt dem neuen Datensatz zu entnehmen und der Faktor w_{Sit} regelt die Berücksichtigung vergangener Beobachtungen.

Gleichung (5-16) verändert nur die Werte der Wahrscheinlichkeit $P_n(m_{\text{Sit}k} \mid r)$, deren Zustände durch den neuen Datensatz angesprochen werden. Ein Kenngrößen-Datensatz mit den Beobachtungen $\{ m_{\text{Sit}k} = \text{ja} \}$ und $\{ r = \text{kat1\_4} \}$ bezieht sich lediglich auf den Wert $P_n(m_{\text{Sit}k} = \text{ja} \mid r = \text{kat1\_4})$, aber nicht auf die übrigen Werte wie zum Beispiel $P_n(m_{\text{Sit}k} = \text{nein} \mid r = \text{kat1\_4})$ oder $P_n(m_{\text{Sit}k} = \text{nein} \mid r = \text{kat0})$. Auf Grund der Netztopologien können die möglichen Zustände des Wurzelknotens bei der Adaption anhand von Gleichung (5-16) nicht in Rivalität treten. Deshalb werden zusätzliche, „künstliche” Datensätze erzeugt, um quantitative Aussagen über alle Einträge der Wahrscheinlichkeitstabelle machen zu können. Für jeden im aktuellen Datensatz nicht beobachteten Zustand des Wurzelknotens eines Netzwerks wird ein Datensatz geschaffen, der den entsprechenden Zustand aktiviert und außerdem die Situationsvariable auf den Zustand „nein“ setzt. Somit kann ein Situationsparameter für einen bestimmten Zustand des Wurzelknotens mehr und für die übrigen Zustände entsprechend weniger charakteristisch sein. Abbildung 5.18 zeigt die Erstellung eines künstlichen Datensatzes für ein Kenngrößennetz.

Für alle Werte der zu bestimmenden bedingten Wahrscheinlichkeit $P(m_{\text{Sit}} \mid r)$ wird Gleichung (5-16) auf alle Datensätze angewendet. Die Idee dabei ist, dass in dem Maße, wie ein Situationsparameter für einen Zustand des Wurzelknoten charakteristisch ist, dieser Situationsparameter im gleichen Maße uncharakteristisch für alle anderen Zustände des Wurzelknotens ist. Die iterativ zu verändernden Wahrscheinlichkeiten $P(m_{\text{Sit}} \mid m_{\text{KG}})$ werden ausgehend von ihrem ursprünglichen Wert durch den Faktor w_{Sit} in Richtung des für die Konvergenz erforderlichen Wertes $P_{n+1}(m_{\text{KG}} \mid \mathbf{m}_{\mathbf{Sit}})$ entwickelt, indem Gleichung (5-16) auf alle neuen Datensätze angewandt wird. Ausgehend von

einem Startwert 0 für w_{Sit} wird dieser Faktor in jeder Iteration schrittweise erhöht. Danach folgt wieder Schritt ③ des Adaptionsverfahrens.

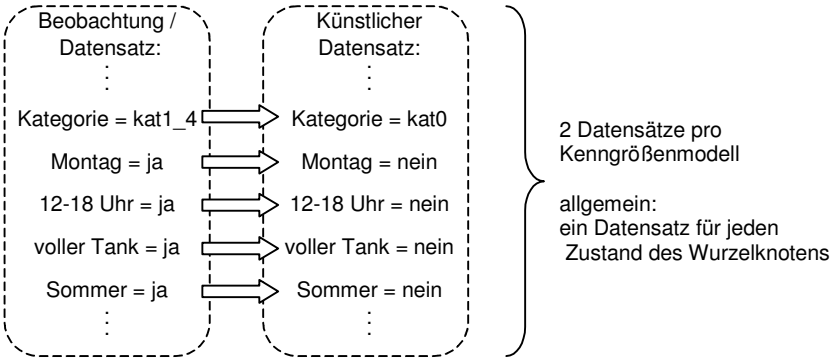


Abbildung 5.18: Erzeugung „künstlicher" Datensätze für ein Kenngrößennetz, um alle Werte der bedingten Wahrscheinlichkeiten der Merkmalsknoten $P(m_{\text{Sit}} \mid m_{\text{KG}})$ zu adaptieren

Mit jeder neuen eindeutig interpretierbaren Eingabe ist das vorgestellte Adaptionsverfahren auf alle 28 Kenngrößennetze, auf das Distanznetz sowie auf das Defaultnetz anzuwenden.

5.8 Intentionsbasierte Interpretation von Situationen und Aktionen

Für die situative Zielort-Prädiktion dient die aktuelle Situation, beschrieben durch den Situationsvektor $\mathbf{m}_{\text{Sit}}$ und dem aktuellen Ort l_{aktuell}, als Eingangsgröße des Intentionsmodells. Die Beobachtungen bezüglich der Situation werden auf das Intentionsmodell abgebildet, um Wahrscheinlichkeiten zu berechnen, die als Basis für die Prädiktion dienen.

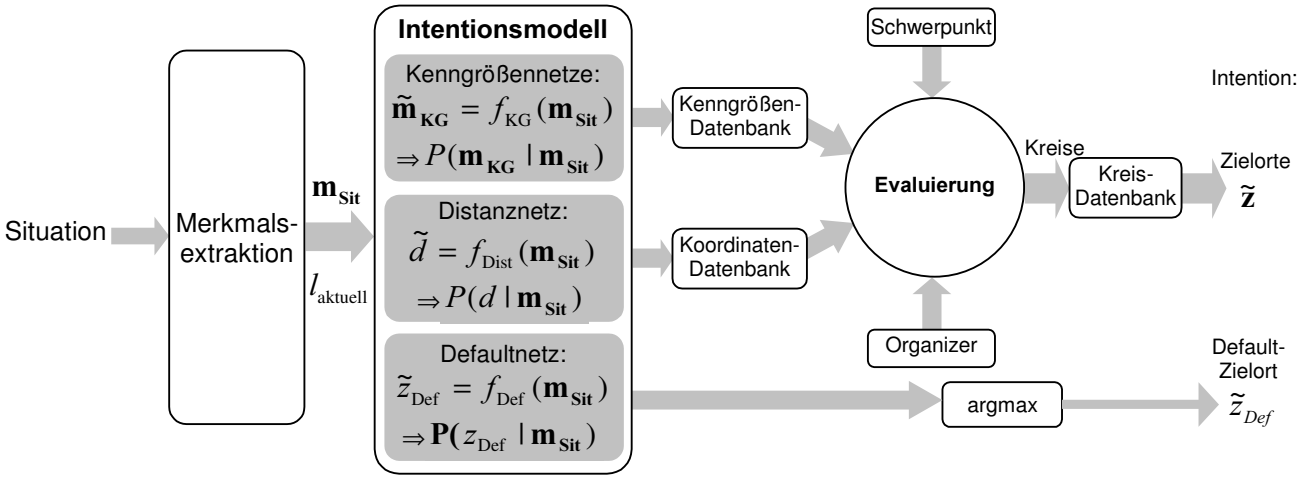


Abbildung 5.19: Überblick über die intentionsbasierte Interpretation von Situationen zur Prädiktion der Benutzerintention

Abbildung 5.19 zeigt das Prinzip der situativen Prädiktion der Benutzerintention. Dabei sind zwei Arten zu unterscheiden. Die erste Art besteht in der Bewertung aller Orte auf Basis der situativen Fahrerinteressen, mit dem Ziel, eine Aussage darüber zu treffen, inwiefern eine Intentionshypothese ein potenzieller Zielort ist. Die zweite Ausprägung der Zielort-Prädiktion ist die Bestimmung eines konkreten Orts als Default-Zielorts. Da sich beide Prädiktionsarten grundsätzlich unterscheiden, werden sie in den folgenden beiden Abschnitten getrennt behandelt.

5.8.1 Zielort-Prädiktion

Gibt der Benutzer einen für das Navigationssystem nicht eindeutigen Zielort ein, so ist es die Aufgabe von *Adaptive Compass,* diese Mehrdeutigkeiten aufzulösen. Hierzu werden alle Intentionshypothesen auf Kreisebene hinsichtlich ihrer Relevanz als Zielort quantitativ und situationsabhängig evaluiert. Somit können alle Zielort-Alternativen miteinander verglichen und der wahrscheinlichste Zielort ermittelt werden. Entsprechend wird bei der Begrenzung des Ortsnamenvokabulars vorgegangen. In diesem Fall ist es auf einfache Weise möglich, zum Beispiel die 5000 wahrscheinlichsten Zielorte zu bestimmen.

In die Berechnung des Evaluierungsmaßes eines Kreises $E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit}})$ gehen die Präferenzen bezüglich bestimmter Ortseigenschaften mit dem Faktor $E_{\text{KG}}(\mathbf{m}_{\mathbf{Sit}})$, die Distanz vom aktuellen Kreis zum zu evaluierenden Kreis mit $E_{\text{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ und die globalen Benutzerpräferenzen mit E_{Global} ein. Da die Evaluierungswerte Wahrscheinlichkeiten sind, werden diese Faktoren unter der Annahme statistischer Unabhängigkeit miteinander multipliziert, um eine situative, quantitative Gesamtaussage $E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit}})$ über den zu bewertenden Kreis zu treffen:

$$E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit}}) = E_{\text{KG}}(\mathbf{m}_{\mathbf{Sit}}) \cdot E_{\text{Distanz}}(\mathbf{m}_{\mathbf{Sit}}) \cdot E_{\text{Global}} \tag{5-17}$$

Zur Berechnung von $E_{\text{KG}}(\mathbf{m}_{\mathbf{Sit}})$ ist die quantitative Annahme $Bel_{\text{KG}i}(\mathbf{m}_{\mathbf{Sit}})$, dass der zu evaluierende Kreis für den Fahrer in der Situation $\mathbf{m}_{\mathbf{Sit}}$ auf Grund einer charakteristischen Eigenschaft (Kenngröße KG*i*) von Interesse ist, zu bestimmen. Grundlage hierfür ist die Interpretation der Wahrscheinlichkeit $P(m_{\text{KG}i} = \text{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}})$. Ein Wert größer als 0,5 bedeutet, dass der Fahrer in der Situation $\mathbf{m}_{\mathbf{Sit}}$ Gegenden bevorzugt, für die ein bestimmter Wirtschafts- oder Freizeitaspekt (Kenngröße KG*i*) überdurchschnittlich charakteristisch ist. $Bel_{\text{KG}i}(\mathbf{m}_{\mathbf{Sit}})$ wird dann gleich $P(m_{\text{KG}i} = \text{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}})$ gesetzt, falls die Kategorie des betrachteten Kreises für die entsprechende Kenngröße mindestens kat1 oder höher ist. Wurde dem Kreis die Kategorie kat0 der Kenngröße KG*i* zugewiesen, so nimmt $Bel_{\text{KG}i}(\mathbf{m}_{\mathbf{Sit}})$ den Wert $P(m_{\text{KG}i} = \text{kat0} \mid \mathbf{m}_{\mathbf{Sit}})$ an. Unterschreitet $P(m_{\text{KG}i} = \text{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}})$ den Wert 0,5, d.h., der komplementäre Wert $P(m_{\text{KG}i} = \text{kat0} \mid \mathbf{m}_{\mathbf{Sit}})$ überschreitet 0,5, so lässt sich dies als Gleichgültigkeit gegenüber dem betreffenden Wirtschaftszweig interpretieren, da die Tatsache, dass eine Branche in einer Gegend nur unterdurchschnittlich vertreten ist, für den Fahrer keine Motivation für einen Besuch dieses Kreises sein dürfte. Diese Gleichgültigkeit bezüglich einer Branche wird mit dem Wert 0,5 quantitativ beschrieben.

Die gleiche Wertzuweisung (0,5) erfolgt im Falle nicht verfügbarer Informationen bezüglich einer Kenngröße eines bestimmten Kreises. Gleichung (5-18) fasst die Bestimmung von $Bel_{\text{KG}i}(\mathbf{m}_{\mathbf{Sit}})$ bezüglich der Kenngröße KG*i* zusammen:

$$Bel_{\mathrm{KG}i}(\mathbf{m}_{\mathbf{Sit}}) = \begin{cases} P(m_{\mathrm{KG}i} = kat(\mathrm{KG}i) \mid \mathbf{m}_{\mathbf{Sit}}) & \text{für } P(m_{\mathrm{KG}i} = \mathrm{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}}) \geq 0{,}5 \\ 0{,}5 & \text{für } P(m_{\mathrm{KG}i} = \mathrm{kat1\_4} \mid \mathbf{m}_{\mathbf{Sit}}) < 0{,}5 \\ 0{,}5 & \text{falls Kategorie unbekannt} \end{cases} \tag{5-18}$$

Diese Evaluierung wird für alle 28 Kenngrößen durchgeführt. Da es sich bei den Werten von $Bel_{\mathrm{KG}i}(\mathbf{m}_{\mathbf{Sit}})$ um Wahrscheinlichkeitswerte handelt, führt dies unter der Annahme statistischer Unabhängigkeit der Kenngrößeninteressen und durch Normierung zu folgender Gleichung:

$$E_{\mathrm{KG}}(\mathbf{m}_{\mathbf{Sit}}) = \frac{1}{Bel_{\mathrm{KG,max}}(\mathbf{m}_{\mathbf{Sit}})} \prod_{i=1}^{28} Bel_{\mathrm{KG}i}(\mathbf{m}_{\mathbf{Sit}}) \tag{5-19}$$

Für den Evaluierungsfaktor $E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ wird mit Hilfe des Distanz-Modells der fahrertypische Aktionsradius situativ modelliert. Die Wahrscheinlichkeitsverteilung $P(d \mid \mathbf{m}_{\mathbf{Sit}})$ gibt Aufschluss über die in entsprechenden Situationen typischerweise zurückgelegten Distanzen. Abhängig von der Entfernung des zu evaluierenden Kreises zum aktuellen Standort d_{Kreis}, wird das Maß $Bel_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ für eine quantitative Aussage über den geschätzten Aktionsradius in der aktuellen Situation ermittelt. Hierfür werden die Wahrscheinlichkeiten der einzelnen Kreisringe, abhängig von der Entfernung des zu evaluierenden Kreises vom aktuellen Standort, in folgender Weise addiert:

$$Bel_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}}) = \begin{cases} 1 & d_{\mathrm{Kreis}} < 50km \\ P(d = \mathrm{d}_2 \mid \mathbf{m}_{\mathbf{Sit}}) + P(d = \mathrm{d}_3 \mid \mathbf{m}_{\mathbf{Sit}}) & \\ \quad + P(d = \mathrm{d}_4 \mid \mathbf{m}_{\mathbf{Sit}}) + P(d = \mathrm{d}_5 \mid \mathbf{m}_{\mathbf{Sit}}) & 50km \leq d_{\mathrm{Kreis}} < 100km \\ P(d = \mathrm{d}_3 \mid \mathbf{m}_{\mathbf{Sit}}) + P(d = \mathrm{d}_4 \mid \mathbf{m}_{\mathbf{Sit}}) & \\ \quad + P(d = \mathrm{d}_5 \mid \mathbf{m}_{\mathbf{Sit}}) & 100km \leq d_{\mathrm{Kreis}} < 250km \\ P(d = \mathrm{d}_4 \mid \mathbf{m}_{\mathbf{Sit}}) + P(d = \mathrm{d}_5 \mid \mathbf{m}_{\mathbf{Sit}}) & 250km \leq d_{\mathrm{Kreis}} < 500km \\ P(d = \mathrm{d}_5 \mid \mathbf{m}_{\mathbf{Sit}}) & d_{\mathrm{Kreis}} \geq 500km \end{cases} \tag{5-20}$$

Zur Bestimmung von $E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ wird $Bel_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ normiert, indem $Bel_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ des betrachteten Kreises durch das über alle Kreise bestimmte Maximum $Bel_{\mathrm{Distanz,max}}(\mathbf{m}_{\mathbf{Sit}})$ dividiert wird:

$$E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}}) = \frac{1}{Bel_{\mathrm{Distanz,max}}(\mathbf{m}_{\mathbf{Sit}})} Bel_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}}) \tag{5-21}$$

Der Evaluierungsfaktor $E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ beschreibt quantitativ die Möglichkeit eines Kreises, als Zielkreis in Frage zu kommen. Diese Aussage wird abhängig von der Distanz des Kreises zum aktuellen Kreis und abhängig von den in der Situation $\mathbf{m}_{\mathbf{Sit}}$ typischerweise zurückgelegten Reisedistanzen getroffen. Durch den Faktor $E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$ werden Orte, die näher am aktuellen Standort liegen, generell stärker gewichtet als Gegenden größerer Entfernung. Wie Gleichung (5-21) zeigt, werden

Orte in unmittelbarer Nähe des aktuellen Orts generell maximal gewichtet. Weit entfernte Orte werden niemals stärker gewichtet als näher liegende Gegenden, höchstens gleich stark.

Die Berücksichtigung offensichtlicher, nicht situativer Interessen des Fahrers erfolgt durch den Evaluierungsfaktor E_{Global}, der dem Aktions-Schwerpunkt des Fahrers, sowie Organizer-Einträgen und der Telefon-History Rechnung trägt. Die Koordinaten des Aktions-Schwerpunkts (x_S, y_S) geben den Mittelpunkt des Gebiets an, in dem sich der Fahrer im Allgemeinen aufhält. Orte, deren Entfernung s_{Kreis} zu diesem Schwerpunkt gering ist, werden in der Evaluierung daher stärker gewichtet als weiter entfernte Orte. Gleichung (5-22) beschreibt die Berechnung des verwendeten Evaluierungsfaktors E_{Global}, abhängig von den Koordinaten des aktuellen Kreises:

$$E_{\text{Global}} = \left(1 - \frac{s_{\text{Kreis}}}{2s_{\max}}\right) = \left(1 - \frac{\sqrt{(x_{\text{aktuell}} - x_S)^2 + (y_{\text{aktuell}} - y_S)^2}}{2s_{\max}}\right) \tag{5-22}$$

Die Entfernung $s_{\max}$ entspricht dabei der größten, beobachteten Entfernung unter allen Kreisen, um die Evaluierung eines Kreises mit den anderen Kreisen in Bezug zu setzen. Der Faktor 2 in Nenner verhindert eine zu schwache Gewichtung zu weit entfernter Kreise.

Informationen aus dem Organizer, wie Kalender-Einträge, Telefonnummern, Adressen, sowie Telefonnummern aus der Telefon-History, werden ebenfalls durch den Faktor E_{Global} modelliert. Die Idee dabei ist, dass Orte, die in diesen Datenbanken beobachtet werden, für den Fahrer permanent von Interesse sein könnten, unabhängig vom aktuellen Standort und von der Entfernung dieser Orte vom Aktions-Schwerpunkt. Für diese Orte bzw. Kreise wird somit der Wert s_{Kreis} aus Gleichung (5-22) zu Null gesetzt, um eine maximale Gewichtung dieser Kreise für die Evaluierung zu erzielen.

Zusammenfassend ergibt sich aus den Gleichungen (5-17), (5-19), (5-21) und (5-22) und mit Normierung folgendes Evaluierungsmaß eines Kreises:

$$E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit}}) = \frac{\prod_{i=1}^{28} Bel_{\text{KG}i}(\mathbf{m}_{\mathbf{Sit}}) \cdot Bel_{\text{Distanz}}(\mathbf{m}_{\mathbf{Sit}}) \cdot \left(1 - \frac{s_{\text{Kreis}}}{2s_{\max}}\right)}{E_{\max}(\mathbf{m}_{\mathbf{Sit}}) \cdot Bel_{\text{KG,max}}(\mathbf{m}_{\mathbf{Sit}}) \cdot Bel_{\text{Distanz,max}}(\mathbf{m}_{\mathbf{Sit}})} \tag{5-23}$$

Für die Disambiguierung werden alle Alternativen auf Kreisebene evaluiert und der wahrscheinlichste Zielort dem Fahrer angeboten. Die linke Darstellung von Abbildung 5.20 zeigt die Evaluierungsergebnisse der Kreise, die ein *Neustadt* beinhalten.

Je dunkler die Darstellung eines Kreises, desto größer die Annahme, dass es sich um das korrekte *Neustadt* handelt. In dem dargestellten Fall wurden *Neustadt bei Holstein* und *Neustadt im Vogtland* mit ähnlichem Ergebnis evaluiert. Somit werden dem Fahrer diese beiden Städte zur Auswahl angeboten, da alle Alternativen mit einem Evaluierungsmaß von mindestens 0,7 dem Benutzer zur Auswahl angeboten werden.

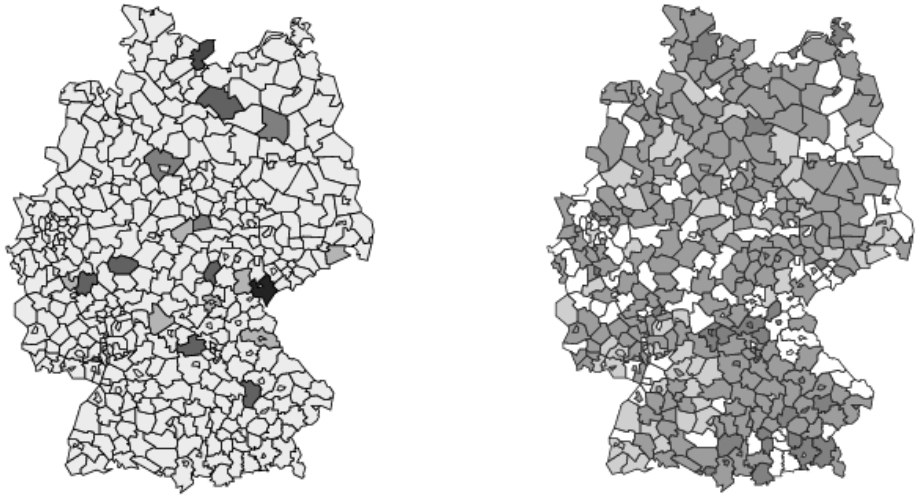


Abbildung 5.20: Visualisierung der Evaluierungsmaße für alle Neustadt-Alternativen und für alle Kreise Deutschlands. Je dunkler, desto wahrscheinlicher ist der entsprechende Kreis als Zielkreis.

Die rechte Darstellung von Abbildung 5.20 zeigt das Ergebnis der Evaluierung aller Orte Deutschlands auf Kreisebene. Je dunkler ein Kreis ausgefüllt, desto wahrscheinlicher ist dieser Kreis als Zielkreis in der aktuellen Situation. Das Ergebnis der Evaluierung kann nun zur Vokabularbegrenzung oder zur Interpretation der Spracherkenner-Ergebnisse verwendet werden. Für die Vokabularbegrenzung können beispielsweise die 50 wahrscheinlichsten Zielkreise berechnet und deren Orte als Wortschatz für die Spracherkennung herangezogen werden. Interessanter ist die Möglichkeit von *Adaptive Compass*, eine sprachliche Eingabe eines Zielorts zu verstehen. Wie in Kapitel 5.4 erwähnt besteht in diesem Fall die Intentionsbibliothek aus den aus Sicht des Spracherkenners n wahrscheinlichsten Worthypothesen inklusive deren Konfidenzmaße *c*. Das Ergebnis des Spracherkenners wird nun mit dem Ergebnis der Zielort-Prädiktion von *Adaptive Compass* verrechnet, um die Intentionshypothesen auf Plausibilität zu prüfen und eine Entscheidung bezüglich der Benutzerintention zu treffen. Die Konfidenzwerte können hierzu als Wahrscheinlichkeiten mit entsprechendem Wertebereich zwischen 0 und 1 interpretiert werden. Um nun die Ergebnisse der Spracherkennung und der Zielort-Prädiktion in Bezug zu setzen, werden die Konfidenzwerte aller Intentionshypothesen auf Eins normiert und schließlich mit dem Evaluierungsmaß $E_{Kreis}(\mathbf{m}_{\mathbf{Sit}})$ des Kreises, in dem die Zielorthypothese liegt, unter der Annahme statistischer Unabhängigkeit multipliziert:

$$E_{\mathrm{I}}(\mathbf{m}_{\mathbf{Sit}}) = \frac{1}{c_{\max}} c_{\mathrm{I}} \cdot E_{\mathrm{Kreis}}(\mathbf{m}_{\mathbf{Sit}}) \tag{5-24}$$

Diese Gleichung gilt für eine Intentionshypothese mit dem Konfidenzmaß c_{I} in der Situation $\mathbf{m}_{\mathbf{Sit}}$. $c_{\max}$ ist das maximale beobachtete Konfidenzmaß einer Hypothese. Diese Berechnung ist für alle Intentionshypothesen durchzuführen. Der Worthypothese mit dem maximalen Evaluierungswert $E_{\mathrm{I}}(\mathbf{m}_{\mathbf{Sit}})$ wird der Vorzug gegeben. Das Ergebnis der Spracherkennung wurde somit auf Plausibilität geprüft, indem Hintergrundwissen über die Situation und über den Benutzer hinzugezogen wurde. Wie schon die in Kapitel 3 und 4 vorgestellten Systeme unterstreicht auch *Adaptive Compass* die Fähigkeit des intentionsbasierten Ansatzes, den Benutzer zu *verstehen*.

5.8.2 Prädiktion des Default-Zielorts

Zur Berechnung des Default-Zielorts wird die aktuelle Situation auf das Defaultnetz abgebildet. Jeder Zustand des Wurzelknotens entspricht einer Intentionshypothese I, d.h., jeder Eintrag der Intentionsbibliothek wird durch einen Zustand repräsentiert.

Der Evaluierungswert entspricht der a-posteriori-Wahrscheinlichkeit $P(z \mid \mathbf{m}_{\mathbf{Sit}})$ und lässt sich direkt aus dem Default-Netz bestimmen. Für die *t*-te Intentionshypothese gilt:

$$E_{\mathrm{I}_t,\mathrm{def}} = P(z = z_t \mid \mathbf{m}_{\mathbf{Sit}}) \tag{5-25}$$

Die Intentionshypothese mit dem größten Evaluierungsmaß ist schließlich der Default-Zielort $\tilde{z}_{\mathrm{def}}$:

$$\tilde{z}_{\mathrm{def}} = \arg\max_i \{P(z = z_i \mid \mathbf{m}_{\mathbf{Sit}})\} \tag{5-26}$$

Dieses Ergebnis wird dem Fahrer angeboten, sobald er in das Navigationsmenü wechselt und der zuletzt eingegebene Zielort dem aktuellen Standort entspricht. Im Falle einer korrekten Prädiktion muss der Fahrer den Zielort lediglich bestätigen. Hat der Fahrer ein anderes Ziel, so kann der Default-Zielort ignoriert und das tatsächliche Ziel eingegeben werden.

5.9 Ergebnisse und Diskussion

Die Fähigkeiten von *Adaptive Compass*, über die Fahrerintention Aussagen zu treffen, wurde anhand von Testdaten eingehend evaluiert und dokumentiert. Dieses Kapitel stellt diese Ergebnisse für Disambiguierung von Zieleingaben sowie für die Prädiktion des Default-Zielorts dar.

5.9.1 Prädiktion zur Disambiguierung der Zieleingabe

Um *Adaptive Compass* adäquat zu testen und zu evaluieren, ist die Protokollierung der Fahrgewohnheiten in Abhängigkeit von allen Situationsparametern über einen möglichst großen Zeitraum nötig. Zudem wäre die Analyse von Personen mit unterschiedlichstem beruflichem und privatem Hintergrund sinnvoll. Die Akquirierung dieser Daten bedeutet einen immensen Aufwand und künstlich erzeugte Datensätze können für die Disambiguierung kaum authentische Verhältnisse widerspiegeln.

Da keine Daten aus größeren Versuchsreihen zur Verfügung standen, wurde die situative Disambiguierung von Ortsnamen nur auf Basis der Daten einer Person evaluiert. Dabei handelt es sich um 25 Datensätze, die sich aus entsprechend vielen Situationsbeschreibungen und den jeweiligen Fahrerintentionen zusammensetzen. Da die Disambiguierungsleistung des Navigationsassistenten erfasst werden soll, sollten die Datensätze möglichst viele mehrdeutige Zielorteingaben beinhalten, um statistische Aussage treffen zu können. Aus diesem Grund wurden einige eindeutige Zieleingaben durch Orte mit nicht eindeutigen Ortsnamen aus demselben Kreis ersetzt. Da die Evaluierung von Orten auf Kreisebene vollzogen wird, sind aus Systemsicht alle Orte eines Kreises äquivalent

und können für die Disambiguierung deshalb beliebig ausgetauscht werden, ohne dass dies zu einer Verfälschung des Ergebnisses oder des Systemzustands führt.

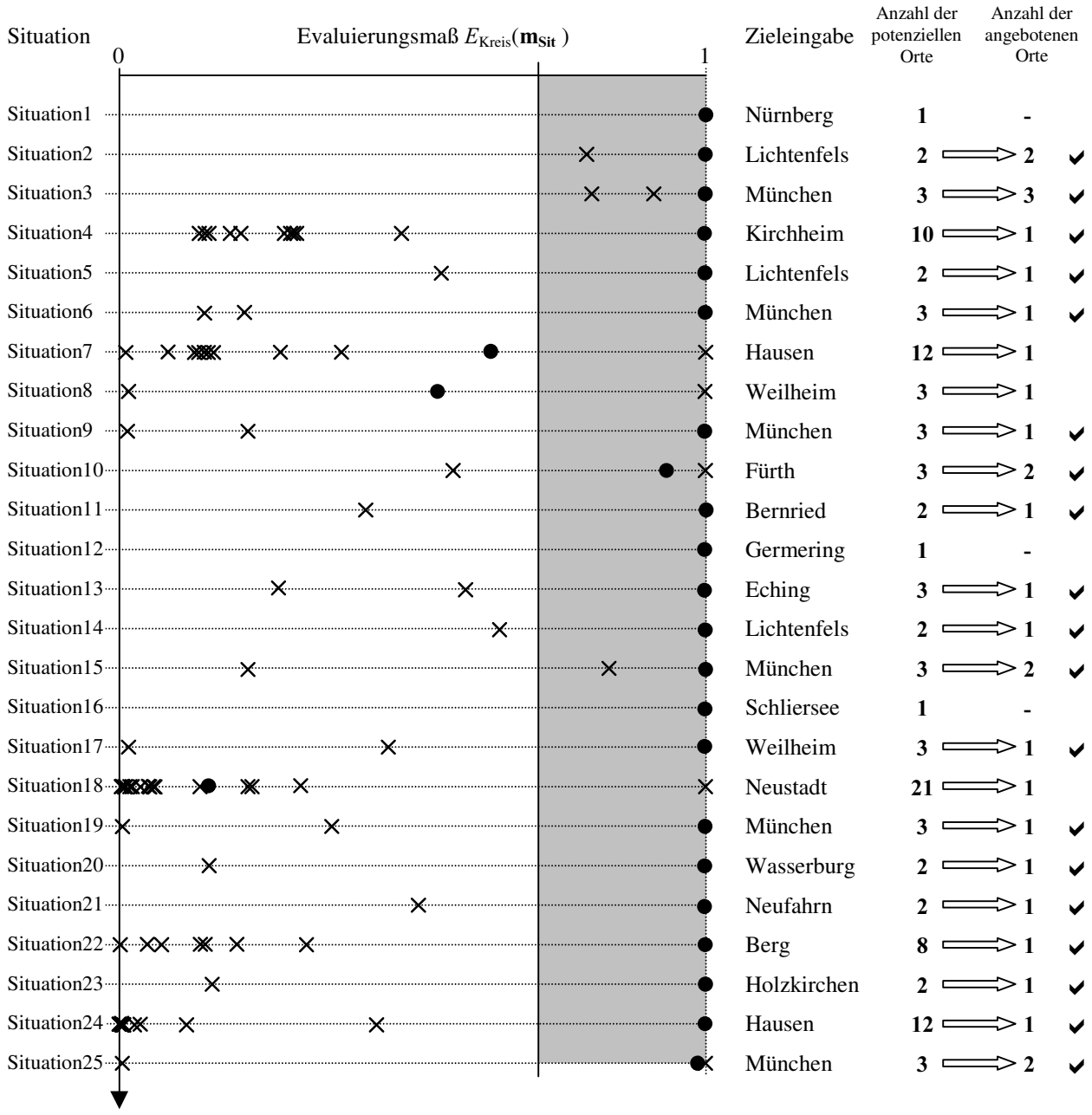


Abbildung 5.21: Ergebnisse der Disambiguierung für 25 Zielorteingaben bzw. Situationen

Abbildung 5.21 stellt die Disambiguierungsergebnisse für 25 Situationen und Zielorteingaben dar. Da es sich bei den Evaluierungsmaßen um Wahrscheinlichkeiten handelt, liegt der Wertebereich zwischen 0 und 1. Der grau unterlegte Wertebereich ($0{,}7 \leq E_{\text{Kreis}}(\mathbf{m}_{\text{Sit}}) \leq 1$) dient zur Bestimmung der für die Dialogmodellierung nötigen wahrscheinlichsten Zielortalternativen. Alle potenziellen Orte, deren Evaluierungswerte in diesem Wertebereich liegen, werden dem Fahrer zur Auswahl angeboten. Der Evaluierungswert des vom Fahrer gewünschten Ortes ist durch einen schwarzen Kreis auf der Skala markiert, die Evaluierungswerte der übrigen Alternativen sind durch ein Kreuz visualisiert. Weiterhin sind die in den jeweiligen Situationen getätigten Zieleingaben aufgelistet

sowie die Anzahl der Orte, auf welche die nicht eindeutige Zieleingabe zutrifft, d.h. die Anzahl der Ziel-Alternativen.

Abbildung 5.21 zeigt schließlich noch die Anzahl der Orte, die dem Fahrer zur Bestätigung bzw. zur Auswahl angeboten werden. Diese Informationen sind schließlich das eigentliche Ergebnis der Disambiguierung. Eine korrekte Disambiguierung wurde erreicht, wenn das Evaluierungsmaß des Wunschzielorts (schwarzer Kreis) im grau unterlegten Wertebereich liegt. Aus diesen Ergebnissen formuliert der Navigationsassistent weitere Dialogschritte. Der Fahrer bekommt von *Adaptive Compass* einen Ort vorgeschlagen, den er lediglich zu bestätigen braucht, falls ausschließlich das Evaluierungsmaß des Wunschzielorts die Entscheidungsschwelle überschreitet. Oder der Fahrer kann aus einer stark reduzierten Liste an Ortsnamen seinen Wunschzielort auswählen, falls mehrere Alternativen auf Grund ihrer Evaluierungsmaße zu berücksichtigen sind. Korrekte Auflösungen von Mehrdeutigkeiten sind am rechten Rand durch einen Haken vermerkt.

Von den 25 Zieleingaben waren 22 nicht eindeutig interpretierbar und somit die Disambiguierung durch *Adaptive Compass* gefordert. In 19 Situationen war dies erfolgreich, was einer Klassifikationsrate von 86,4 % entspricht. Bei der Interpretation dieser Aussage ist zu beachten, dass sich diese Zahl nicht auf den eingeschwungenen Systemzustand, d.h. auf den Zustand sinnvoll konvergierter bedingter Wahrscheinlichkeiten bezieht, sondern auf alle in Abbildung 5.21 dargestellten Beobachtungen. Vor allem in den ersten Situationen modellieren die bedingten Wahrscheinlichkeiten der Bayes'schen Netze noch kein Gedächtnis, das sinnvolle Inferenzen über die Fahrerintention zulässt. Wann sich die Intentionsmodelle im eingeschwungenen Zustand befinden, ist nicht exakt bestimmbar, da auf Grund der Vielfalt möglicher Situationen nicht abzuschätzen ist, über welche Situation bereits ausreichend Aussagen getroffen wurden bzw. welche Situationen überhaupt noch nicht beobachtet wurden. Dennoch zeigt die Abbildung eine Tendenz, die auf eine gewisse Konvergenz der Intentionsmodelle hindeutet. Mit wachsender Anzahl von Beobachtungen sinken die Evaluierungswerte der falschen Zielalternativen beträchtlich. Dieser Trend zeigt, dass sich der Navigationsassistent mit jedem neuen Datensatz der idealen Klassifikation, die ein Evaluierungmaß mit dem Wert 1 für den richtigen Ort und die Evaluierungsmaße 0 für alle falschen Orte zum Ergebnis hätte, annähert.

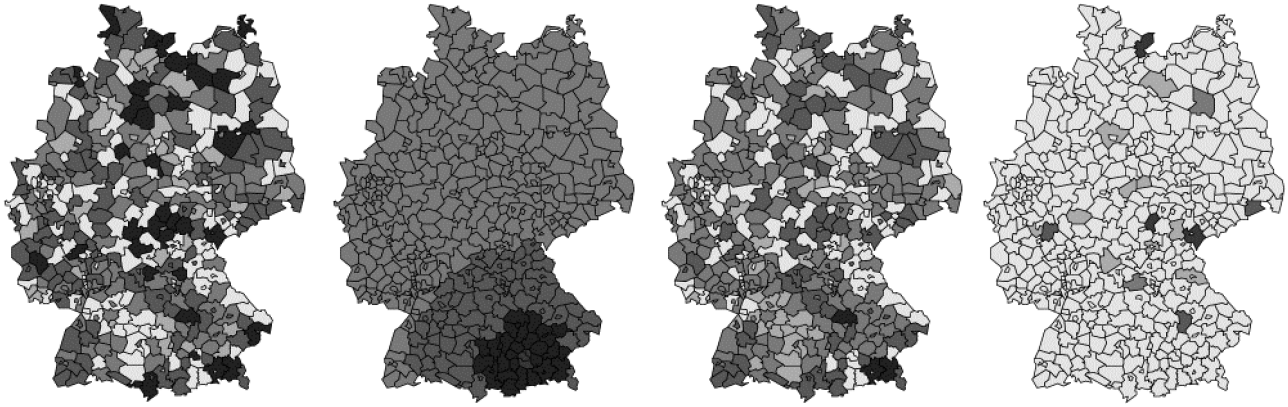
Die Tatsache, dass die ersten fünf Disambiguierungen korrekte Resultate zum Ergebnis haben, ist durch die anfängliche schwächere Gewichtung weiter entfernter Orte sowie auf das Miteinbeziehen von Organizer-Informationen zurückzuführen. Im weiteren Verlauf nehmen die Kenngrößenmodelle immer größeren Einfluss auf die Evaluierung der Kreise.

Von den 19 korrekten Disambiguierungen hatten 14 einen einzigen Ort zum Ergebnis, d.h., zu 73,7 Prozent konnte dem Fahrer der Wunschzielort direkt zur Bestätigung angeboten werden. Für die im Rahmen der Untersuchungen verwendeten Datensätze gab es durchschnittlich 4,86 Zielortalternativen pro Zielorteingabe, die durchschnittliche Anzahl der dem Fahrer korrekt zur Auswahl angebotenen Orte beträgt 1,37. Damit konnte *Adaptive Compass* die Auswahllisten durch Entfernung sehr unwahrscheinlicher Orte oder durch ausschließliches Anbieten des wahrscheinlichsten Ortes um 71,8 Prozent reduzieren. Vor allem dieses Ergebnis dokumentiert den Gewinn bringenden Einsatz des Navigationsassistenten für die Disambiguierung von Zielorteingaben, da eine Reihe irrelevanter

Ortsnamen aussortiert wird und der Fahrer sich somit erheblich zielgerichteter auf die Auswahl seines Wunschziels konzentrieren kann.

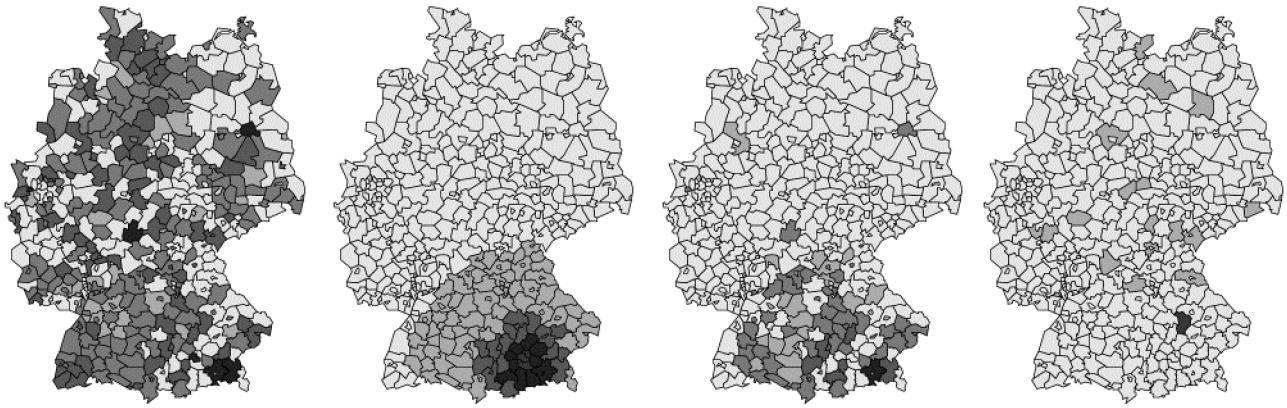
Abbildung 5.21 zeigt drei Fehldisambiguierungen. In jedem dieser Fälle wurde ein falscher Ort mit einem erheblich höherem Evaluierungsmaß bewertet als die übrigen Alternativen und als der eigentliche Wunschzielort. Dies könnte zum Beispiel darauf zurückzuführen sein, dass diese Orte in nächster Nähe zum aktuellen Standort liegen, oder dass diese Orte durch einen Eintrag im Adressbuch oder dem Kalender des Organizers stärker gewichtet werden. Bei allen fehlerhaften Prädiktionen wurde dem Fahrer nur ein einziger Ort zur Bestätigung oder Ablehnung angeboten. Der Fahrer kann somit unmittelbar erfassen, dass der angebotene Ort nicht seiner Intention gleicht, diesen Ort als Ziel ablehnen und schließlich aus der Liste der Alternativen das eigentliche Ziel mit dem kompletten Namen auswählen.

Situation1:
Wetter: Sonne, Temp.: 26 °C, Zeit: 9:50 Uhr, Tag: Sonntag, Monat: Mai, Straßenbed.: trocken, Insassen: 4, aktueller Ort: München, Tankinhalt:98 %, Letzte Betankung: gestern



$E_{\mathrm{KG}}(\mathbf{m}_{\mathbf{Sit1}})$ $E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit1}})$ $E_{\mathrm{Kreis}}(\mathbf{m}_{\mathbf{Sit1}})$ $E_{\mathrm{Neustadt}}(\mathbf{m}_{\mathbf{Sit1}})$

Situation2:
Wetter: Sonne, Temp.: 18 °C, Zeit: 7:24 Uhr, Tag: Mittwoch, Monat: Mai, Straßenbed.: trocken, Insassen: 1, aktueller Ort: München, Tankinhalt:34 %, Letzte Betankung: ≥ 2d



$E_{\mathrm{KG}}(\mathbf{m}_{\mathbf{Sit2}})$ $E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit2}})$ $E_{\mathrm{Kreis}}(\mathbf{m}_{\mathbf{Sit2}})$ $E_{\mathrm{Neustadt}}(\mathbf{m}_{\mathbf{Sit2}})$

Abbildung 5.22: Beispiel für den Einfluss der Situation aus die Zielort-Prädiktion. Je dunkler ein Kreis gezeichnet ist, desto größer die Annahme, dass er für den Fahrer von Interesse ist.

Generell können Fehldisambiguierungen auftreten, wenn der Navigationsassistent Aussagen über Situationen treffen muss, die noch nicht ausreichend erlernt wurden. Wie einflussreich die Situation auf die Prädiktion der Benutzerintention sein kann, wird nun anhand zweier verschiedener Situationen und der Zieleingabe *Neustadt* dokumentiert. Hierzu sind für die beiden grundsätzlich unterschiedlichen Situationen die einzelnen situationsabhängig modellierten Aspekte der Evaluierungsmaße eines Kreises in Abbildung 5.22 grafisch dargestellt.

Situation1 spiegelt private bzw. Freizeitinteressen wieder, während *Situation2* einen typischen Berufsalltag modelliert. $E_{KG}(\mathbf{m}_{\mathbf{Sit}})$ zeigt je nach Situation die unterschiedlichen Interessen des Fahrers. In der Freizeit bevorzugt der Fahrer vor allem Gegenden mit hohem Freizeitwert. In ähnlichen Situationen hat der Fahrer bereits Orte am Alpenrand sowie an Nord- und Ostsee aufgesucht. Daraus wird eine stärkere Gewichtung der Kreise mit ähnlichen Eigenschaften für diese Art Situation abgeleitet. Die Tatsache, dass der Fahrer von seinem Wohnort München aus in seiner Freizeit Orte in Norddeutschland aufgesucht hat, schlägt sich in der Evaluierung aller Kreise bezüglich des typischen Aktionsradius nieder. Selbst Ziele mit einer Entfernung von mehr als 500 km vom aktuellen Standort werden als ziemlich wahrscheinlich in die quantitativen Betrachtungen mit einbezogen. Dies wirkt sich schließlich auch auf die Evaluierungsmaße $E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit1}})$ aus, sodass kaum eine lokale Begrenzung von Interessen modelliert wird. Als Ergebnis erhält man drei ähnlich wahrscheinliche Orte, die dem Fahrer direkt zur Auswahl abgeboten werden können.

Zu einem anderen Ergebnis führen die gleichen Berechnungen für *Situation2*, dem beruflichen Alltag. Der Vergleich der beruflichen Interessen $E_{\text{KG}}(\mathbf{m}_{\mathbf{Sit2}})$ mit den privaten Interessen $E_{\text{KG}}(\mathbf{m}_{\mathbf{Sit1}})$ zeigt deutlich die unterschiedlichen Präferenzen des Fahrers in grundsätzlich verschiedenen Situationen. Im Berufsalltag suchte der Fahrer in der Vergangenheit in erster Linie verschiedene Orte im oberbayrischen Raum auf. Diese lokale Orientierung zeigt sich im Aktionsradius, der das Evaluierungsresultat eines Kreises $E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit2}})$ durch den Faktor $E_{\text{Radius}}(\mathbf{m}_{\mathbf{Sit2}})$ derart beeinflusst, dass $E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit2}})$ für Orte mit größerer Distanz zum Standort entsprechend stärker gedämpft und eventuell herausgefiltert werden. Dies hat eine maximale Evaluierung der *Neustadt*-Alternative mit der geringsten Entfernung zum aktuellen Ort zu Folge. Der Ort *Neustadt im Kreis Kelheim* wird schließlich dem Fahrer als Ergebnis der Disambiguierung angeboten.

Die diskutierten Ergebnisse der Untersuchungen von *Adaptive Compass* bestätigten das Potenzial des intentionsbasierten Ansatzes zur Interpretation mehrdeutiger Benutzereingaben. Wie die bereits in den Kapitel 3 und 4 vorgestellten Systeme orientiert sich der Navigationsassistent dabei am zwischenmenschlichen Dialog, indem Wissen über mögliche Gesprächsinhalte bzw. Benutzerintentionen mit den Benutzeraktionen in Bezug gesetzt wird. Die Intentionsmodelle werden herangezogen, um Interessen des Benutzers zu erfassen und um Aussagen über seine Ziel-Präferenzen für konkrete Situationen treffen zu können. Dies und die Charakterisierung von Gegenden anhand von Kenngrößen ermöglicht *Adaptive Compass* ein Transferdenken, sodass selbst bei dem Fahrer unbekannten Orten die Frage beantwortet werden kann, inwiefern diese Gegenden als potenzielle Zielorte zu berücksichtigen sind. Darüber hinaus hat diese Interpretation des intentionsbasierten Ansatzes gezeigt, dass selbst Intentionsbibliotheken mit bis zu 100000 Einträgen möglich sind, wenn inhaltlich verwandte Intentionshypothesen gebündelt und anhand konkreter Attribute beschrieben werden. Die Intentionsmodelle erfassen dann die Korrelation der Merkmale mit diesen Attributen.

5.9.2 Prädiktion des Default-Zielorts

Das Abwägen des Default-Zielorts richtet sich vor allem an Pendler, die in ähnlichen Situationen immer wieder gleiche Ziele haben. Der Fahrer fährt zum Beispiel am Morgen zu dem Ort seines Arbeitplatzes, am Abend ist sein Wohnort das Ziel. Dem Fahrer soll in diesen Situationen die Möglichkeit gegeben werden, den vom Navigationsassistenten vorhergesagten Zielort zu bestätigen oder im Falle einer fehlerhaften Prädiktion den Default-Zielort zu ignorieren und das tatsächliche Wunschziel einzugeben. Somit erfordern Fehlprädiktionen keinen weiteren Dialogschritt, da die Angabe eines anderen Zielorts als Ablehnung des bestimmten Default-Zielorts interpretiert wird.

Zur Evaluierung dieser Systemfunktion wurden 28 Datensätze so konstruiert, dass sie die typische Zielwahl eines Berufspendlers widerspiegeln, der morgens am Montag, Dienstag und Freitag von München nach Augsburg, am Mittwoch und Donnerstag von München nach Ulm und jeden Abend zu seinem Wohnort München fährt.

Abbildung 5.23 zeigt, inwiefern der Navigationsassistent alle bisher angefahrenen Orte in die Prädiktion des Default-Zielorts mit einbezieht. In der linken Spalte sind die einzelnen Situationen aufgeführt, die ausschließlich aus dem Tag, der Zeit und dem aktuellen Ort bestehen. Die zweite Spalte gibt die tatsächliche Fahrerintention an. Die dritte Spalte stellt die Evaluierungswerte der bis zum jeweiligen Zeitpunkt angefahrenen Orte als Histogramm dar. Der graue Balken gibt an, welcher Ort aus der Sicht von *Adaptive Compass* der wahrscheinlichste Zielort ist. Dieser Ort wird in der vierten Spalte noch einmal explizit für jede Situation aufgezeigt. Ein Haken hinter dem berechneten Default-Zielort symbolisiert eine korrekte Prädiktion, d.h., Wunschziel und Default-Ziel stimmen überein.

Zu Beginn enthält der Hypothesenraum keine Orte, da der Fahrer mit Hilfe des Navigationssystems noch keinen Ort aufgesucht hat. Durch jeden Datensatz mit einem neuen, bisher noch nicht angefahrenen Ort wird die Intentionsbibliothek um diesen Ort erweitert.

In der zweiten Situation wird dem Fahrer kein Default-Ort angeboten, da Augsburg der dem System bisher einzige bekannte Ort ist, das Fahrzeug sich aber in dieser Situation gerade im Augsburg befindet.

In der fünften Situation hat der Fahrer Ulm zum Ziel, als Default-Zielort wird ihm aber Augsburg angeboten, da dem System zu diesem Zeitpunkt Ulm als potenzielle Hypothese unbekannt ist. Für alle darauf folgenden Situationen wird Ulm als Teil des Hypothesenraums mit einbezogen. Im weiteren Verlauf zeigt sich anhand der Evaluierungswerte deutlich, dass die Fahrerpräferenzen mit zunehmender Anzahl von Datensätzen durch das Default-Netz immer besser erfasst werden.

Von den 26 berechneten Default-Zielorten entsprachen 19 der tatsächlichen Fahrerintention, d.h., die Erkennungsrate beträgt 73,1 Prozent. Betrachtet man nur die Arbeitstage, da der Fahrer nur während dieser Tage bezüglich der Zielwahl regelmäßiges Verhalten zeigt, so wird in 18 von 21 Situationen die Fahrerintention als Default-Zielort vorhergesagt. Die Erkennungsrate für diesen Fall beträgt somit 85,7 Prozent.

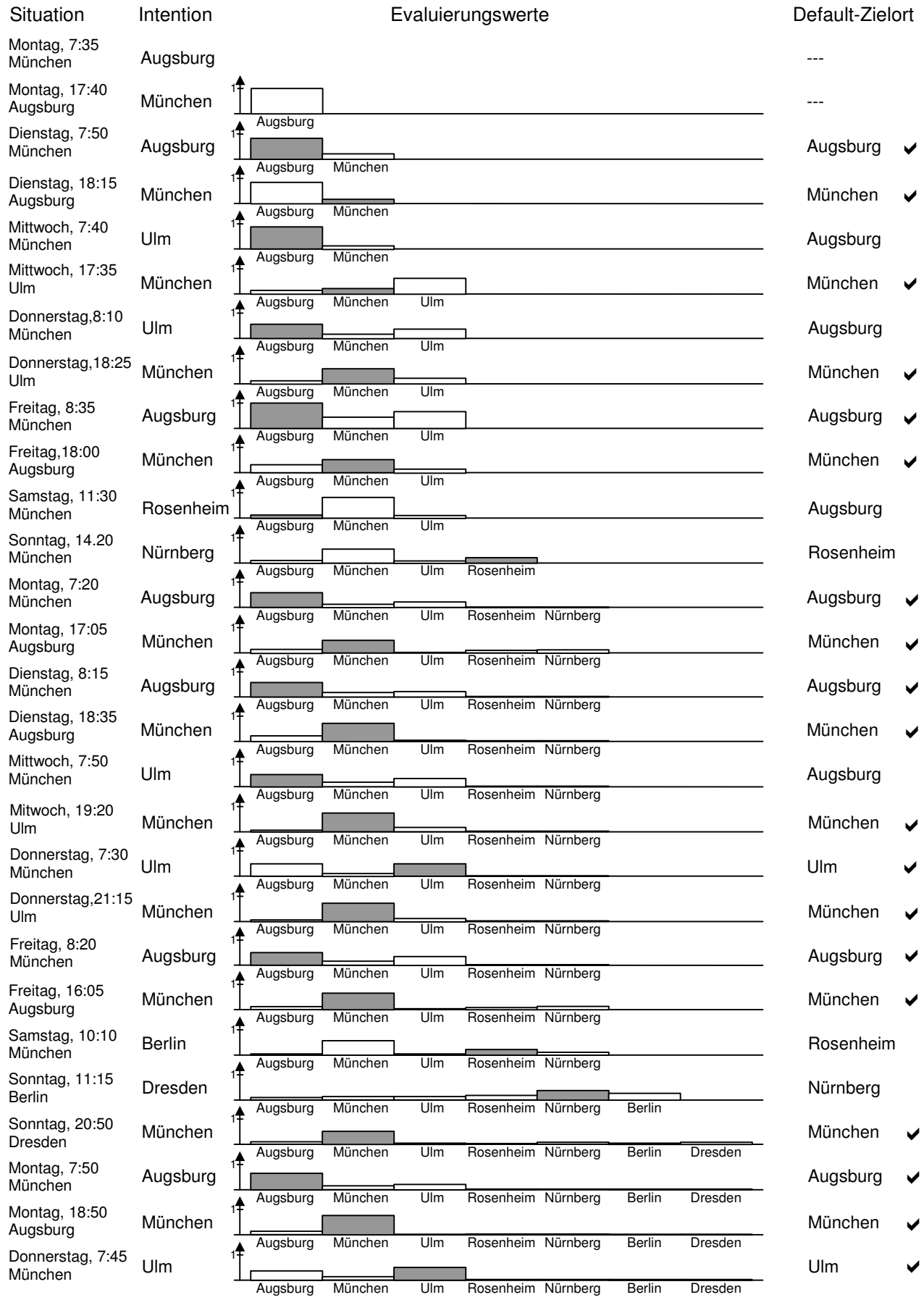


Abbildung 5.23: Zielort-Prädiktion zur Bestimmung des Default-Zielortes

5.10 Implementierung von *Adaptive Compass*

Die Implementierung des Navigationsassistenzsystems wurde unter Linux in C++ [Str92] realisiert. Die Echtzeitfähigkeit wurde auf einem Pentium-III-PC mit 600 Mhz Taktfrequenz nachgewiesen. Abbildung 5.24 zeigt die Systemarchitektur der Implementierung von *Adaptive Compass*.

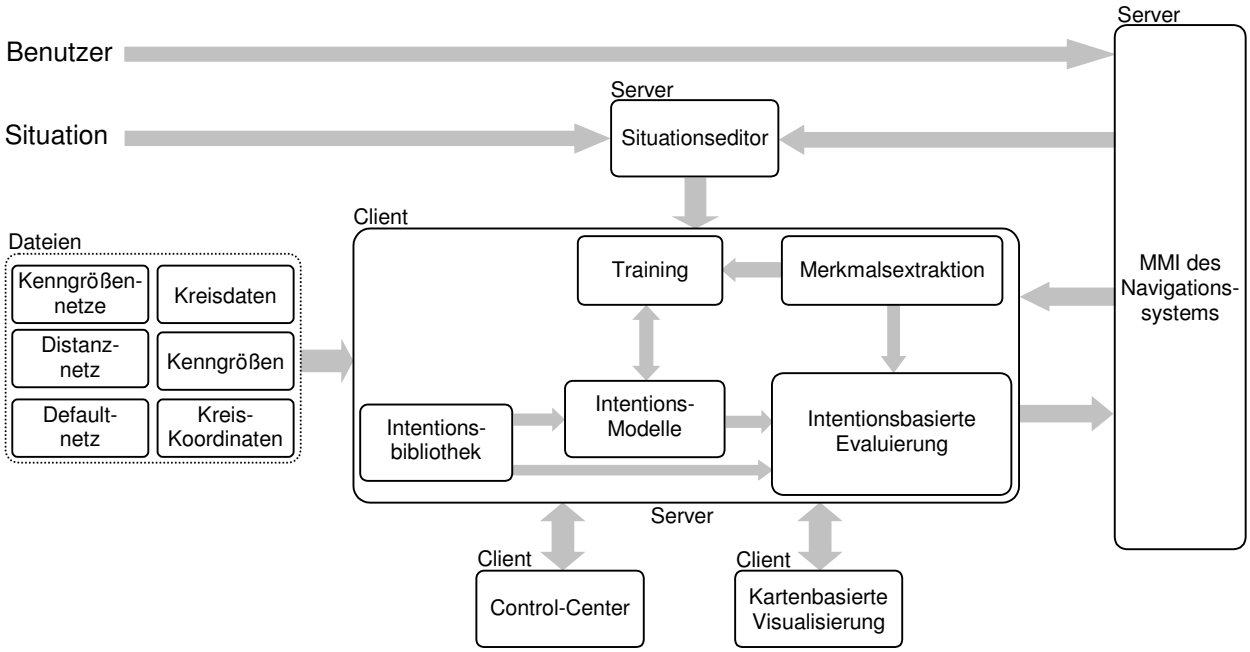


Abbildung 5.24: Systemarchitektur der *Adaptive Compass*-Implementierung

Bei der Endanwendung kommt der Benutzer mit dem Navigationsassistenten nicht direkt in Kontakt, sondern über das MMI des Navigationssystems. Aus diesem Grund wurde für die Zielorteingabe eine Schnittstelle zu einem fiktiven Navigationssystem realisiert. Abbildung 5.25 zeigt einen Screenshot der in Tcl/Tk [Wel99] als Server implementierten grafischen Oberfläche.

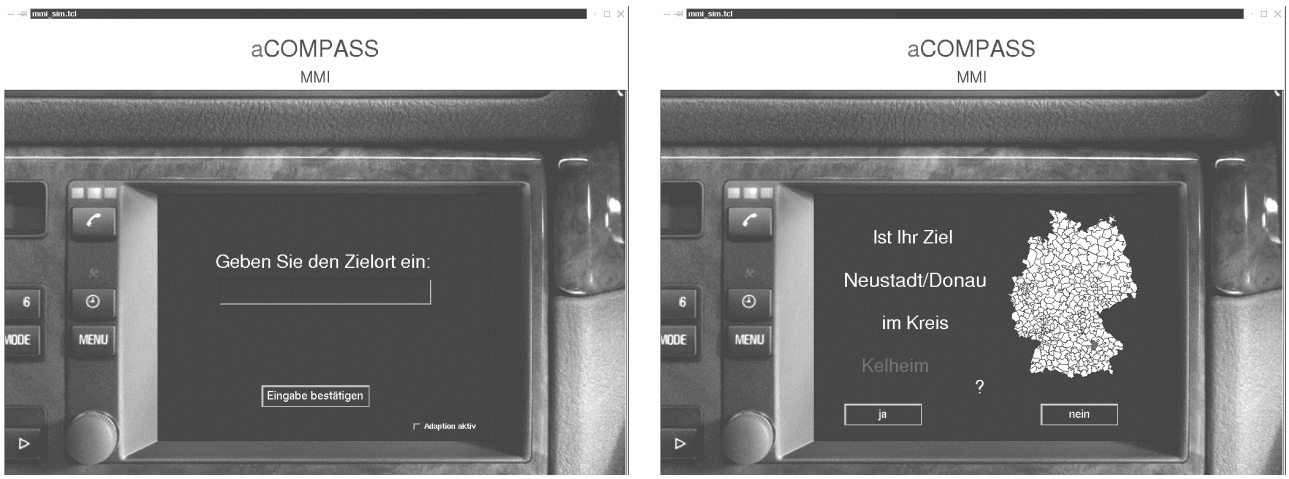


Abbildung 5.25: MMI des *Adaptive-Compass*-Navigationssystems

Diese Komponente steuert den Navigationsassistenten an und entscheidet je nach Interpretierbarkeit der Benutzereingabe zwischen Trainings- oder Prädiktionsmodus. Im Falle einer nicht eindeutigen Zieleingabe fordert das MMI Aussagen über alle Intentionshypothesen an und triggert somit den Prädiktionsprozess. Im linken Bild ist die Zielorteingabemaske zu sehen. Das rechte Bild zeigt eine Darstellung, in der der Benutzer nach Eingabe eines mehrdeutigen Ziels einen konkreten Ort angeboten bekommt. Wurde die Eingabe korrekt disambiguiert, so muss der Fahrer lediglich diesen Ort als neues Navigationsziel bestätigen.

Da das System bisher nicht in ein Fahrzeug integriert wurde, sondern nur als Desktop-Version implementiert wurde, müssen die einzelnen Parameter einer Situation anhand eines Situationseditors eingegeben werden. Hierzu dient ein als Server in Tcl/Tk realisierter Situationseditor. Abbildung 5.26 zeigt einen Screenshot des Situationseditors.

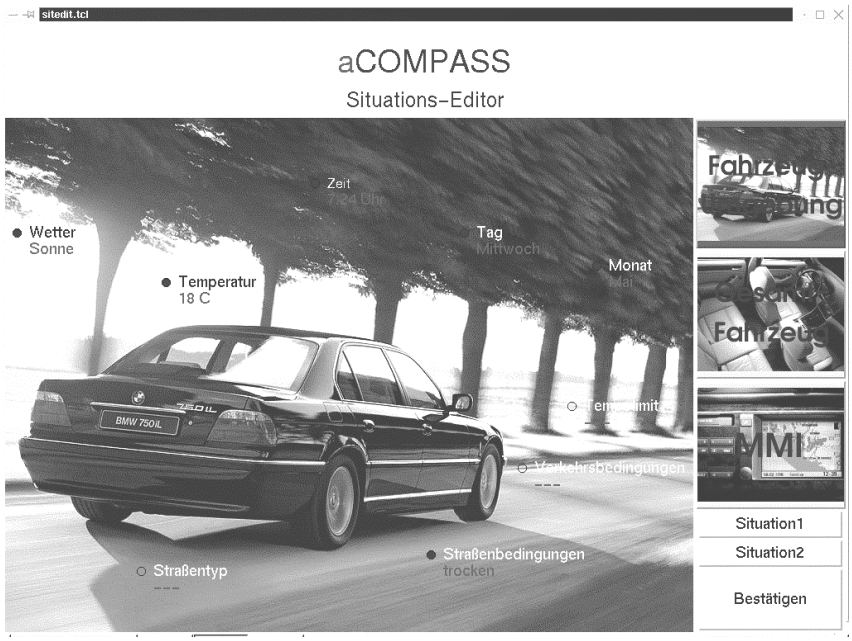


Abbildung 5.26: Situationseditor zur Simulation einer Situation bestehend aus Fahrzeugumgebung, Gesamtfahrzeug und MMI/IT-Komponenten

Im Falle einer Integration von *Adaptive Compass* in ein Fahrzeug muss der Situationseditor durch eine Komponente ersetzt werden, die auf Basis von CAN- und MOST-Busdaten selbständig alle für die Beschreibung einer Situation nötigen Daten akquiriert.

Der Navigationsassistent selbst wurde in C++ als Client des MMIs realisiert. Bei Start werden alle zur Initialisierung von *Adaptive Compass* benötigten Daten eingelesen. Dabei handelt es sich um Dateien zur Definition der Bayes'schen Netze sowie um Orts- und Kreisdaten.

Die Hauptkomponente von *Adaptive Compass* kommuniziert als Server mit zwei Clients zur Parametrisierung und Überwachung des Verfahrens und zur Visualisierung von Prädiktionsergebnissen. Abbildung 5.27 zeigt einen Screenshot des Control-Centers. Diese Schnittstelle ermöglicht Eingriffe in das Adaptionsverfahren, indem beispielsweise die Gewichtungsfaktoren verändert werden

können. Wie die Abbildung zeigt, können die exakten Evaluierungswerte aller Intentionshypothesen ausgegeben werden. Das Speichern und Laden von Intentionsmodellen ist ebenfalls möglich.

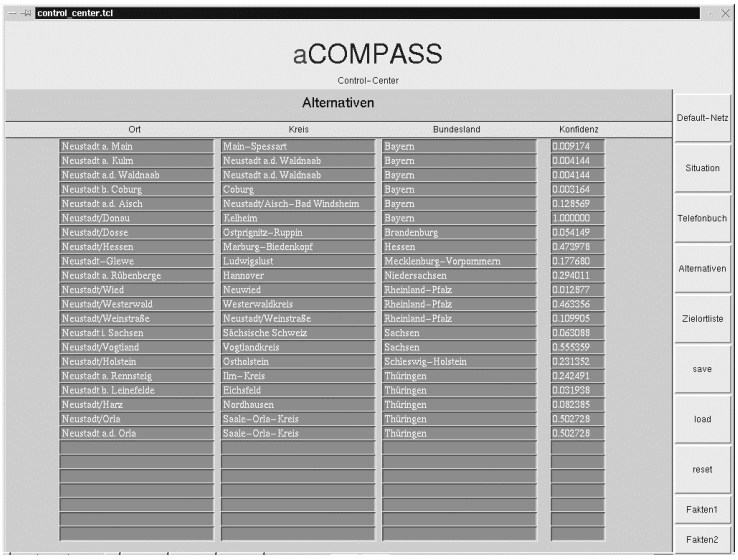


Abbildung 5.27: Screenshot von Control-Center

Abbildung 5.28 zeigt die kartenbasierte Visualisierung der Evaluierungswerte. Für jeden Kreis können die Evaluierungswerte $E_{\text{Kreis}}(\mathbf{m}_{\mathbf{Sit}})$, $E_{\text{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$, $E_{\text{KG}}(\mathbf{m}_{\mathbf{Sit}})$ und E_{Global} grafisch dargestellt werden, um Gesamtüberblick darüber zu vermitteln, welche Präferenzen der Fahrer bezüglich seiner Ziele hat.

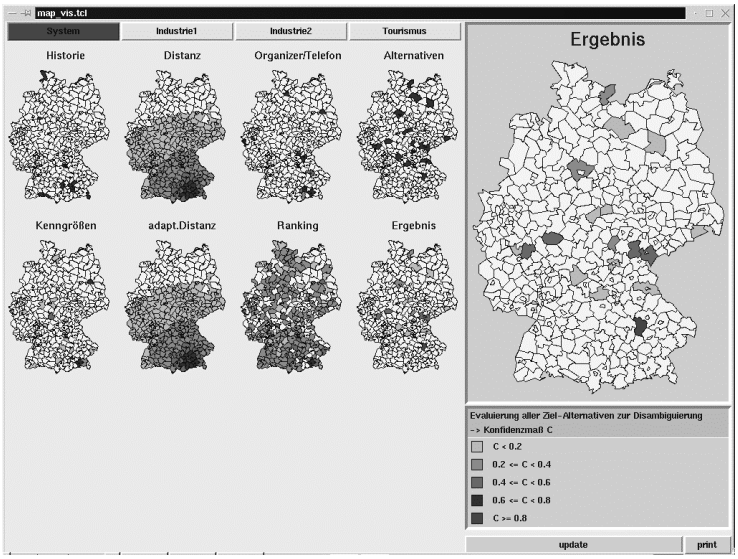


Abbildung 5.28: Screenshot des kartenbasierten Visualisierung

6

Intentionsbasierte Interpretation dynamischer Handgesten: Gestikerkennung

Gegenstand dieses Kapitels ist die Anpassung des intentionsbasierten Ansatzes für die Klassifikation dynamischer Handgesten. Aussagekräftige Merkmale gestatten eine einfache Struktur des Intentionsmodells und ermöglichen dadurch sowohl Training als auch Adaption des Gestikerkenners in Echtzeit.

6.1 Grundidee

Das Ziel, die Interaktion eines Menschen mit einer Maschine so intuitiv wie möglich zu gestalten, ist nur erreichbar, wenn der Benutzer die aus dem Alltag vertrauten und natürlichen Kommunikationskanäle nutzen kann. Neben der Sprache handelt es sich dabei auch um Gestik.

Sprache zeichnet sich dadurch aus, dass sie innerhalb eines Kulturkreises trotz enormer Komplexität allgemein verständlich ist und somit ohne vorhergehende Übereinkünfte als Kommunikationsmedium zwischen Menschen verwendet werden kann. Das Spektrum der Sprache reicht dabei von sehr einfachen Lauten oder Phrasen, die auf bestimmte Emotionen schließen lassen, bis hin zur Erörterung tief greifender Zusammenhänge. Die Problematik bei der Spracherkennung liegt vor allem im äußerst umfangreichen Vokabular.

Gestik als Kommunikationskanal weist ganz andere Charakteristika als Sprache auf. Im Alltag setzt der Mensch Gestik vor allem als unterstützendes Element für eine überzeugendere Darstellung der zu vermittelnden Inhalte ein. Da dies im allgemeinen unbewusst geschieht, ist die Art zu gestikulieren erheblich durch die Person und weniger durch den Kulturkreis geprägt. Es gibt somit keine Übereinkünfte bezüglich eines bestimmtes Gestenvokabulars und dessen Interpretation. Dies wirkt sich gravierend auf die Nutzung von Gestik für einen natürlichen Mensch-Maschine-Dialog aus, da Usability-Studien [Zob02] zwar Aussagen über typischerweise verwendete Gesten erlauben, die interpersonellen Unterschiede allerdings zu deutlich sind, um mit einem festgelegten Gestenvokabular allen potenziellen Benutzern intuitiv ausführbare Referenzgesten bereitzustellen. Hinzu kommt eine starke Abhängigkeit des Gestenvokabulars von der zu steuernden Applikation. Diese Aspekte erfordern die Fähigkeit des Gestikerkenners, sich jedem Benutzer individuell

Aspekte erfordern die Fähigkeit des Gestikerkenners, sich jedem Benutzer individuell anzupassen, zum einen um die Erkennungsleistung deutlich zu erhöhen, zum anderen um das System auf andere Referenzgesten zu trainieren, die aus Sicht des Benutzers intuitiver sind als die vorgesehenen Gestentypen. Da das zu berücksichtigende Vokabular eines Gestikerkenners verglichen mit der Spracherkennung sehr begrenzt ist, ist eine rasche Adaption aller Referenzgesten möglich.

Die Adaption des Gestikerkenners kann durch kooperatives Verhalten des Benutzers geschehen. Der Benutzer führt eine Geste aus und teilt dem System bei falscher Klassifikation seine eigentliche Intention mit. Die Kooperation des Benutzers lässt sich allerdings unter zwei Bedingungen erreichen. Zum einen muss sich die Erkennungsleistung schon nach sehr wenigen vom Benutzer induzierten Adaptionsschritten spürbar verbessern, zum anderen darf ein Adaptionsschritt den Interaktionsfluss mit der Applikation nicht behindern. Dies ist vor allem durch Echtzeitadaption bzw. Echtzeittraining zu gewährleisten. Aus diesen Ansprüchen an einen Gestikerkenner wurden folgende drei Anforderungen an das hier vorgestellte System *byHand* [Hof02] abgeleitet:

- Robuste Klassifikation von Handgesten
- Echtzeit-Training und rasche -Adaption an den Benutzer
- Einsatz für unterschiedlichste Gestenvokabulars

Für die Entwicklung des Gestikerkenners *byHand* wurde die zweite Ausprägung des intentionsbasierten Ansatzes herangezogen. Während die Systeme *Insense* und *AMPlan* semantisch höherwertige, übergeordnete Intentionen modellieren und bestimmen können, bezieht sich *byHand* auf den einfachsten Fall einer Intention: Eine Intention ist äquivalent mit einer einzigen Aktion. Dies trifft zwar auch auf Adaptive Compass zu, durch Erlernen genereller Benutzerinteressen können jedoch dennoch Aussagen über übergeordnete Absichten gemacht werden. Der Gestikerkenner *byHand* ermittelt nicht die übergeordneten Ziele eines Benutzers im Umgang mit einer Applikation, sondern nur die zuletzt durchgeführte Aktion bzw. Geste. Nachdem in der Planerkennung (Kapitel 3) längerfristige Ziele, die nur durch komplexe Aktionssequenzen erreichbar sind, erfasst werden, wird im Rahmen der Gestikerkennung die einfachste Interpretation des Intentionsbegriffs behandelt. Diese große Bandbreite an unterschiedlichen Intentionsdefinitionen in den vier Systemen erlaubt schließlich eine umfassendere Gesamtbewertung des intentionsbezogenen Ansatzes.

Der Gestikerkenner *byHand* bezieht kein Kontextwissen über die Applikation ein, da drei sehr unterschiedliche Gestenarten verwendet wurden, die keine gemeinsame kontextuelle Betrachtung gestatten. Syntaktisch-semantische Zusammenhänge zwischen aufeinander folgenden Gesten wurden nicht modelliert, da zwei der drei getesteten Gestenvokabulars die Eingabe von Ziffern vorsehen und z.B. bei der Eingabe einer Telefonnummer keine charakteristischen Abhängigkeiten zwischen aufeinander folgenden Ziffern zu erwarten sind. Deshalb wurde der Fokus darauf gelegt, Gesten ohne Einbezug von Kontextwissen zu erkennen. Es handelt sich somit um ein klassisches Mustererkennungsproblem, das aber mit Hilfe des intentionsbasierten Ansatzes gelöst werden soll.

Für die Umsetzung des intentionsbasierten Ansatzes für die Klassifikation dynamischer Handgesten wird die Philosophie von *Adaptive Compass* (Kapitel 5) fortgeführt, aussagekräftige Merkmale mit einer möglichst einfachen Struktur des Intentionsmodells zu kombinieren. Dies gewährleistet Training und Adaption des Intentionsmodells in Echtzeit. Um die Struktur des Intentionsmodells mög-

lichst einfach zu halten, wird die zeitliche Modellierung einer Bildabfolge größtenteils auf die Merkmale verlegt. Das Intentionsmodell fügt dann die Merkmale, die Gesten in bestimmten zeitlichen Abschnitten repräsentieren, zu einer Einheit zusammen und ermöglicht so Aussagen über die Benutzerintention. Die Umsetzung des intentionsbasierten Ansatzes zur Interpretation von dynamischen Handgesten resultiert in dem innovativen System *byHand*.

6.2 Stand der Technik

Technologien zur Gestikerkennung, für die der Benutzer spezielle, mit Sensorik ausgestattete Kleidungsstücke tragen muss, wie zum Beispiel einen Datenhandschuh, werden mittlerweile sehr gut beherrscht und sind robust. Diese Randbedingung ist allerdings für alltägliche Anwendungsszenarien nicht hinnehmbar. Deshalb hat sich die videobasierte Gestikerkennung, unterstützt durch die zunehmende Leistungsfähigkeit von Rechnern, in den letzten Jahren zu einem zentralen Forschungsgebiet der Mustererkennung entwickelt.

Auf Grund des Erfolgs von Hidden-Markov-Modellen (HMM) zur Modellierung zeitlicher Prozesse in der Spracherkennung zeichnet sich ein starker Trend ab, diese Verfahren auf für die Klassifikation dynamischer Handgesten einzusetzen. Einer der ersten Forscher war Starner [Sta95], der HMMs nutzte, um eine begrenzte Zahl Symbole der amerikanischem Zeichensprache zu klassifizieren. Mittlerweile haben sich HMMs für die Erkennung dynamischer Hand- und Körpergesten zum Stand der Technik etabliert [Yam92][Rig97].

Morguet [Mor00] nutzt HMMs für die Klassifikation dynamischer Handgesten. In diesem Ansatz wird die Hand vom Hintergrund durch Farbsegmentierung unterschieden. Aufbauend darauf wird die Handform in jedem Frame einer Bildsequenz mathematisch erfasst, indem Hu-Momente auf Basis der Einzelbilder berechnet werden. Die Hu-Momente eines Frames, die Differenz von Hu-Momenten aufeinander folgender Bilder sowie die Position des Handschwerpunkts bilden schließlich den Merkmalsvektor zur Charakterisierung einer dynamischen Handgeste. Für die eigentliche Klassifikation werden schließlich semikontinuierliche HMMs verwendet.

Oka [Oka02] entwickelte ein Verfahren zur Gestikerkennung, das sowohl manipulative als auch symbolische Gesten erkennen kann. Zunächst werden die Fingerkuppen in den Einzelbildern erfasst. Über ein Kalman-Filter wird dann der Zusammenhang zwischen den Fingerkuppen aufeinanderfolgender Bilder hergestellt. Auf diese Weise werden die Trajektorien der Fingerkuppen erfasst, die den Merkmalsvektor bilden. HMMs klassifizieren schließlich die dynamische Handgeste.

Die Trainingsverfahren der HMMs sind zu rechenintensiv, um eine Echtzeit-Adaption an den Benutzer und neue Gesten zu ermöglichen. Das hier vorgestellte System *byHand* basiert auf einem (dynamischen) Bayes'schen Netz und ist in der Lage, sich online an einen Benutzer anzupassen und neue Gesten in Echtzeit zu lernen. Dadurch, dass dynamische Bayes'schen Netze nicht so effizient zeitliche Prozesse modellieren können wie HMMs, wird ein Teil der zeitlichen Modellierung von den *byHand*-Merkmalen übernommen. *ByHand* unterscheidet sich somit von gängigen Ansätzen zur Gestikerkennung in der Wahl des Klassifikators und in der Verwendung neuartiger Merkmale, die die Handform und die Handdynamik erfassen.

6.3 Systemarchitektur

Wie das sprachverstehende System *Insense*, der Planerkenner *AMPlan* und der Navigationsassistent *Adaptive Compass* ist der Gestikerkenner *byHand* entsprechend des intentionsbasierten Ansatzes konzipiert. Die Systemarchitektur des intentionsbasierten Ansatzes ist, auf die Klassifikation dynamischer Handgesten angepasst, in Abbildung 6.1 dargestellt. Der Benutzer gestikuliert in einem definierten, von einer Kamera überwachten Bereich. Die Beobachtungsfolge **o** entspricht somit einem Videosignal bzw. der korrespondierenden Bildfolge, die schließlich die Basis für die Merkmalsextraktion einer Handbewegung bildet.

Die Intentionsbibliothek umfasst das Gestenvokabular und enthält die Intentionshypothesen bzw. Gesten, die dem Benutzer durch den aktuellen Applikationskontext gestattet werden.

Das Bindeglied zwischen den Merkmalen und den Intentionshypothesen bildet das Intentionsmodell, durch das alle Intentionshypothesen mit den Merkmalen diskriminativ in Beziehung gesetzt werden. Im Trainingsmodus erfasst das Intentionsmodell einen statistischen Zusammenhang zwischen Gesten und deren Merkmalen. Im Adaptionsmodus wird das bestehende Intentionsmodell auf eine neu beobachtete Geste angepasst. Das trainierte Intentionsmodell bildet die Grundlage für die intentionsbasierte Interpretation der Merkmale, indem der Merkmalsvektor des zuletzt analysierten Videosignals auf das Intentionsmodell abgebildet wird und somit jeder Intentionshypothese ein Evaluierungsmaß zugeteilt wird. Die Hypothese mit dem maximalen Evaluierungsmaß entspricht dem Ergebnis des Klassifikationsprozesses, der Benutzerintention bzw. der durchgeführten Geste.

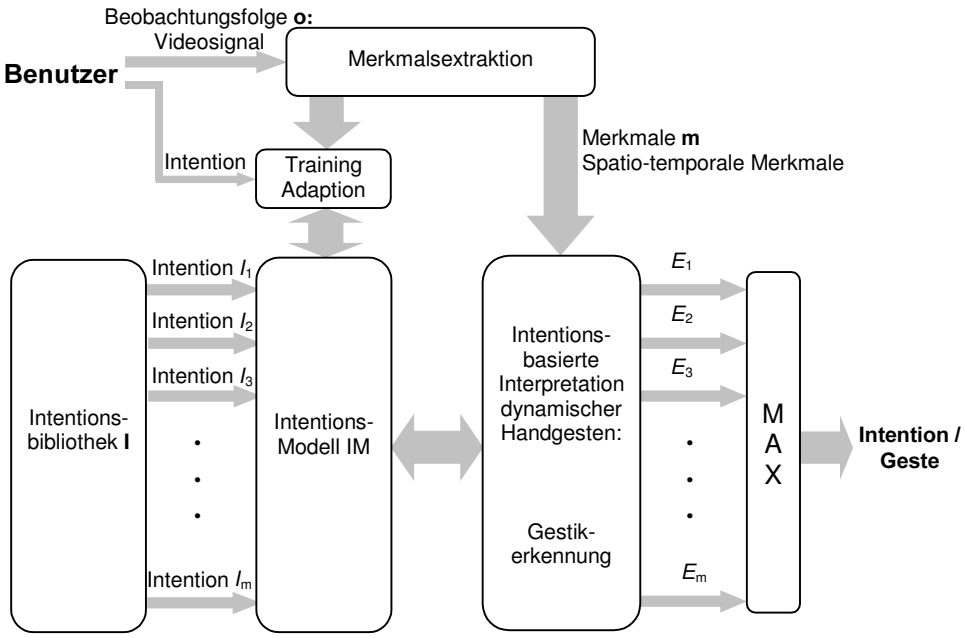


Abbildung 6.1: Intentionsbasierter Ansatz für Klassifikation dynamischer Handgesten

Die folgenden Kapitel erläutern die in Abbildung 6.1 dargestellten Komponenten im Detail. Die Schwerpunkte bilden dabei die Merkmalsextraktion, die Struktur des Intentionsmodells sowie die intentionsbasierte Evaluierung der Bildfolge.

6.4 Intentionsbibliothek

Die Intentionsbibliothek von *byHand* beinhaltet alle für die Klassifikation in Betracht zu ziehenden Gesten. Eine Intentionshypothese entspricht somit einer Geste. Um die Flexibilität und Leistungsfähigkeit des Gestikerkenners zu evaluieren und zu dokumentieren, werden drei grundsätzlich verschiedene Gestentypen als Intentionsbibliothek verwendet. Für jeden Gestentyp wurde ein eigener Demonstrator entwickelt, um die Steuerung einer Applikation durch dynamische Gesten erlebbar zu machen.

Um einen Eindruck davon zu vermitteln, wie die verschiedenen Gesten auszuführen sind und welche Gesten jeweils in der Intentionsbibliothek enthalten sind, werden in diesem Kapitel alle Gestentypen ausführlich mit Beispielen vorgestellt.

6.4.1 Dynamische Vollhandgesten

Der erste Gestentyp wird allgemein als *dynamische Vollhandgesten* bezeichnet, da nur durch Interpretation der Dynamik und der Form der kompletten Hand auf die Geste geschlossen werden kann. Inspiriert ist die Zusammenstellung dieser Intentionsbibliothek durch die Publikationen [Gei02][Zob02], die dieses Gestenvokabular für eine berührungslose Interaktion im Fahrzeug propagieren. Dennoch kann man diese Intentionsbibliothek als allgemein und applikationsunabhängig sehen, da sie eine Auswahl an Gesten bereithält, die für die Steuerung eines Cursors denkbar ist und nicht explizit auf eine bestimmte Bedienlogik zugeschnitten ist.

Für die Berücksichtigung der Dynamik der gesamten Hand wird der in Abbildung 6.2 dargestellte Aufbau verwendet. Die Kamera ist oberhalb des Greifraums angeordnet, um mittels einer zweidimensionalen Bildfolge die Handbewegung in der x-y-Ebene zu erfassen. Die Unterlage wird möglichst dunkel gewählt, sodass sich die Hand gut von dem Hintergrund abhebt.

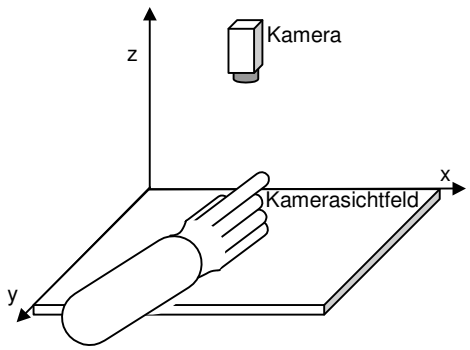


Abbildung 6.2: Kamerapositionierung zur Erfassung dynamischer Handgesten

Abbildung 6.3 zeigt einige Frames (Einzelbilder) der dynamischen Handgeste *Greifen*. Im ersten Frame befindet sich die Hand noch in Ruheposition. Im weiteren Verlauf bewegt sich die Hand in den oberen Bildrand und ballt sich zugleich zu einer Faust, um nach etwas zu greifen. Schließlich zieht die Hand das „Gegriffene“ nach unten, öffnet sich und bewegt sich in die Ausgangsposition zurück.

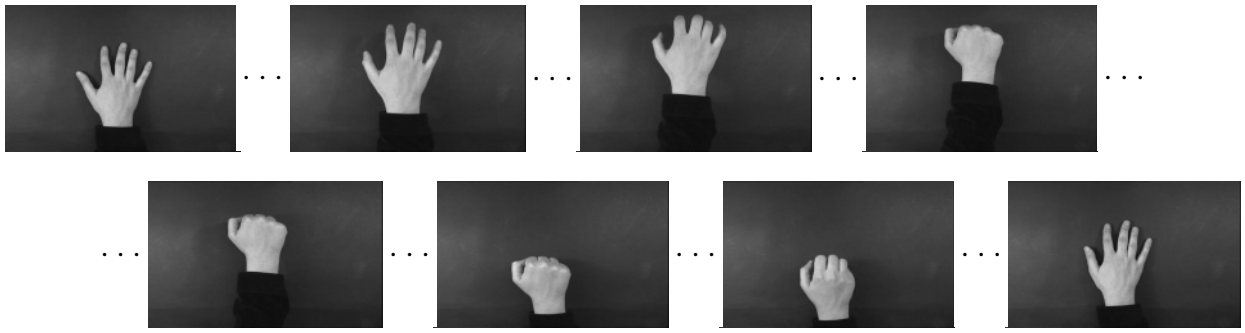


Abbildung 6.3: Auszug aus einer Bildfolge einer typischen *Greifen*-Geste

Die dargestellte Geste ist ein typischer Vertreter dieses Gestentyps. Das gesamte Intentionsbibliothek wird nun tabellarisch aufgezeigt:

Nr.	Geste	Beschreibung
1	Links	Der Benutzer winkt nach links, indem der die Hand nach links bewegt und dabei um 90° um die y-Achse (Daumen nach oben) dreht.
2	Rechts	Der Benutzer winkt nach rechts, indem der die Hand nach rechts bewegt und dabei um 90° um die y-Achse (Daumen nach oben) dreht.
3	Vor	Die Hand bewegt sich vom Körper weg.
4	Zurück	Die Hand bewegt sich zum Körper hin.
5	Zeigen	Der Benutzer bewegt die Finger und die Hand, als würde er mit dem Zeigefinger auf ein bestimmtes Objekt deuten.
6	Daumen oben	Die Hand bewegt sich nach oben und ballt sich mit abgespreizten Daumen zur Faust.
7	Daumen unten	Die Hand bewegt sich nach oben, ballt sich mit abgespreiztem Daumen zur Faust, dreht sich um 180° um die y-Achse und zeigt nach unten.
8	Daumen links	Die Hand bewegt sich nach oben, ballt sich mit abgespreiztem Daumen zur Faust, dreht sich um –90° um die y-Achse und zeigt nach links.
9	Daumen rechts	Die Hand bewegt sich nach oben, ballt sich mit abgespreiztem Daumen zur Faust, dreht sich um +90° um die y-Achse und zeigt nach rechts.
10	Enter	Die Hand betritt das Sichtfeld der Kamera.
11	Leave	Die Hand verlässt das Sichtfeld der Kamera.
12	Auflegen	Die Hand bewegt sich, als würde ein unsichtbarer Telefonhörer vom Ohr kommend aufgelegt werden.
13	Abheben	Die Hand bewegt sich, als würde ein unsichtbarer Telefonhörer abgehoben und in Richtung Ohr geführt werden.
14	Greifen	Die Hand (Handrücken nach oben) bewegt sich in Richtung des oberen Bildbereichs, ballt sich zur Faust und wird in Richtung des unteren Bildbereichs gezogen.
15	Ziehen	Die Hand dreht sich (Handfläche nach oben), bewegt sich in Richtung des oberen Bildbereichs, ballt sich zur Faust und wird in Richtung des unteren Bildbereichs gezogen.

Tabelle 6-1: Intentionsraum für dynamische Vollhandgesten

Die einzelnen Gesten unterscheiden sich sehr stark in den translatorischen Bewegungen und in der Änderung der Handform. Aus Sicht der Kamera gibt es somit sehr grobe, charakteristische Signalveränderungen.

6.4.2 Dynamische Schreibgesten zur Eingabe von Zahlen

Dieser Gestentyp gestattet die Eingabe von Zahlen mittels so genannter *Schreibgesten*. Der Benutzer zeichnet das seiner Intention entsprechende Symbol mit seinem Zeigefinger entweder direkt auf die Unterlage oder „in die Luft“, einige Zentimeter über der Unterlage. Abbildung 6.4 zeigt einige Frames einer Schreibgeste zur Eingabe der Ziffer 9. Von der Ausgangsposition aus formt sich die Hand derart, dass nur der Zeigefinger ausgestreckt bleibt. Zeitgleich begibt sich der Zeigefinger, bzw. die Hand in die Startposition zum Schreiben der Neun. Zur besseren Illustration ist der aus Sicht des Benutzers bereits gezeichnete Teil der Neun durch weiße Linienzüge visualisiert. Deutlich erkennbar ist die im Laufe der Bildfolge gemalte Ziffer.

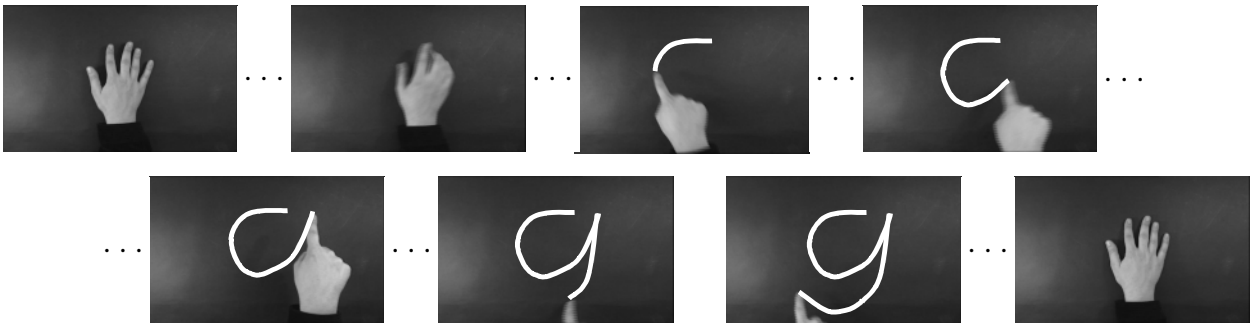


Abbildung 6.4: Auszug aus einer Bildfolge einer Schreibgeste zur Eingabe der Ziffer 9

Die Intentionsbibliothek für den Demonstrator *virtueller Taschenrechner* besteht aus den Möglichkeiten eines sehr einfachen Taschenrechners mit den Ziffern 0 bis und den vier Grundrechenarten:

Nr.	**Geste**	**Beschreibung**
1-10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Ziffern 0 bis 9
11-15	+, -, *, /	Vier Grundrechenarten
16	Clear	Displayinhalt löschen
17	Enter	Die Hand betritt das Kamerasichtfeld
18	Leave	Die Hand verlässt das Kamerasichtfeld

Tabelle 6-2: Intentionsbibliothek des Demonstrators *virtueller Taschenrechner*

Da der Benutzer während des Schreibens keine Rückmeldung über die bisher gezeichneten Linienzüge erhält, werden die Ziffern im Allgemeinen sehr unsauber und undeutlich geschrieben. Dies stellt eine Herausforderung an die Klassifikation dar und spricht für die hier vorgestellte Idee, nicht das Gezeichnete als Basis für die Klassifikation, sondern die Handdynamik während des Schreibens auszuwerten. Dadurch wirkt sich unsauberes und undeutliches Zeichnen weniger gravierend auf den Erkennungsprozess aus.

6.4.3 Dynamische Gesten zur Eingabe von Zahlen für mobile Geräte

Die in den vorhergehenden Abschnitten vorgestellten Gestentypen beziehen sich auf Desktop-Anwendungen. In diesem Abschnitt werden Gesten zur Bedienung eines Mobilfunkgerätes vorgestellt. Der Benutzer hält ein mit einer Kamera ausgestattetes Mobilfunkgerät in der linken Hand, während er mit der rechten Hand die Ziffern einer Telefonnummer „virtuell“ schreibt.

Zunächst werden kurz die technischen Aspekte einer Handy-Attrappe anhand von Abbildung 6.5 erläutert. Integriert ist eine handelsübliche CCD-Kamera, deren Objektiv mit einem Infrarot-Durchlassfilter versehen ist. Die Kamera wird durch Infrarot-Leuchtdioden ergänzt, die den kameranahen Bereich mit nicht sichtbarem Licht beleuchten, um für die Klassifikation störende Hintergrundinformationen für die Kamera nicht wahrnehmbar zu machen. Dies gelingt dadurch, dass sich die rechte Hand des Benutzers während des Schreibens in Kameranähe aufhält und somit Licht im Infrarotbereich reflektiert. Weiter entfernte Objekte werden nicht mehr durch die Infrarot-Dioden angestrahlt und deshalb von der Kamera nicht erfasst.

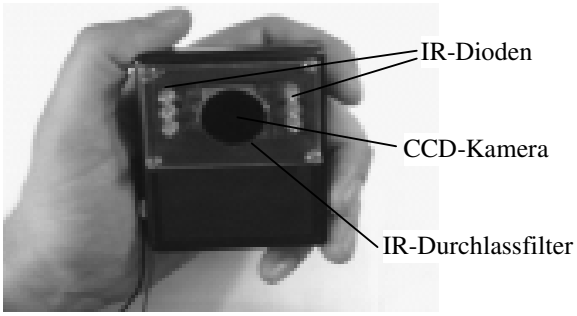


Abbildung 6.5: Technischer Aufbau des Handy-Demonstrators

Abbildung 6.6 zeigt die Eingabe der Ziffer 7 mittels einer Schreibgeste für mobile Geräte:

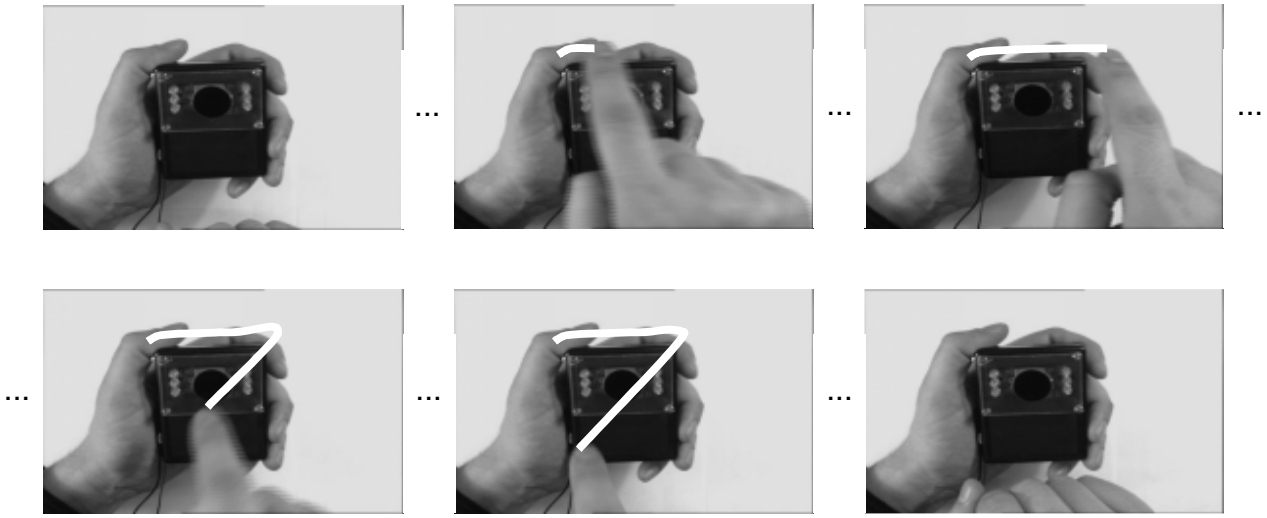


Abbildung 6.6: Schreibgeste zur Eingabe der Ziffer 7 in ein mobiles Gerät

Im ersten Frame nimmt die Infrarotkamera die rechte Hand noch nicht wahr (Abbildung 6.6). Dann bewegt sich die rechte Hand mit gestrecktem Zeigefinger in das Kamerasichtfeld, um den Startpunkt für die 7 zu markieren. Weiß eingezeichnet sind die Linienzüge, die aus Benutzersicht bereits geschrieben wurden. Nachdem die Sieben vollständig ist, zieht der Benutzer seine Hand aus dem Kamerasichtfeld zurück, um die Merkmalsextraktion und die Interpretation zu starten.

Die Intentionsbibliothek des Demonstrators besteht aus den in Tabelle 6-3 aufgezeigten Gesten. Dies sind Gesten zum Ansprechen des Funktionsumfangs eines sehr einfachen Telefons.

Nr.	**Geste**	**Beschreibung**
1-10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Ziffern 0 bis 9
11	Auflegen	Gespräch beenden
12	Abheben	Gespräch entgegennehmen / wählen
13	Clear	Löschen des Displayinhalts

Tabelle 6-3: Intentionsbibliothek für die gestenbasierte Steuerung eines mobilen Geräts

Vorteil des hier vorgestellten Gestentyps ist die berührungslose Bedienung eines mobilen, informationstechnischen Gerätes. Bei aktuellen Mobilfunksgeräten ist die Tastatur das größebestimmende Element, d.h., Handys könnten erheblich kleiner und kompakter produziert werden, wenn die Tastatur durch ein anderes, kleineres Eingabemedium ersetzt werden könnte. Im Zuge des Trends zu mobilen Multimediageräten werden in naher Zukunft alle Mobilfunkgeräte mit einer Kamera ausgestattet sein. Diese Kamera könnte genutzt werden, um die Tastatur durch eine erheblich kompaktere Eingabeschnittstelle zu ersetzen.

6.5 Merkmalsextraktion

Unabhängig von der Art und Komplexität der Gesten handelt es sich bei dynamischen Gesten um eine Überlagerung zweier dynamischer Prozesse. Der eine Prozess beschreibt die translatorischen Bewegungen der Hand, d.h., aus Kamerasicht ändert sich die Position des Handschwerpunkts. Der zweite Prozess steht für die Veränderung der Handform während der Geste und bezieht sich somit auf die Dynamik innerhalb der Hand. Da die beiden Prozesse bei intuitiven, dynamischen Gesten nicht linear unabhängig, sondern Korrelationen zwischen beiden Prozessen zu erwarten sind, die als semantische Beziehung zwischen beiden Arten der Dynamik interpretiert werden können, werden aus dem Videosignal Merkmale extrahiert, die beiden Prozessen sowie der Korrelation Rechnung tragen. Das Erfassen der Hand-Dynamik impliziert eine Zeitabhängigkeit der Bewegung, somit handelt es sich bei den Merkmalen um *spatio-temporale* Merkmale.

Eine Anforderung an den hier vorgestellten Gestikerkenner ist die Fähigkeit, sich in Echtzeit an neue Datensätze anzupassen und neue Gesten zu trainieren. Dies hat zur Folge, dass sich das Intentionsmodell durch eine möglichst einfache Struktur auszeichnen muss, die dennoch leistungsfähig genug ist, um eine hohe Klassifikationsleistung zu gewährleisten. Als Intentionsmodell wird ein Dynamisches Bayes'sches Netz verwendet, um im Gegensatz zu gängigen Bayes'schen Netzen zeit-

liche Abhängigkeiten modellieren zu können. Da dies jedoch auf Grund der Anforderungen an das Training nur in Grenzen möglich ist, modellieren die Merkmale den Hauptteil der Zeitinformation. Die Merkmale von *byHand* übernehmen somit eine der klassischen Aufgaben eines Klassifikators: die zeitlichen Modellierung. Hier müssen die spatio-temporalen Merkmale folgenden Anforderungen genügen:

- Modellierung translatorischer Dynamik, Veränderung der Handform und der Korrelation zwischen beiden Prozessen
- Handgrößeninvarianz: Die Merkmale müssen unempfindlich gegenüber unterschiedlichen Handgrößen sein, die aus Kamerasicht entweder durch verschiedene Benutzer oder durch unterschiedliche Abstände der Hand zur Kamera hervorgerufen werden können.
- Ausgangsortsinvarianz: Da es keine vorgeschriebene Ausgangsposition der Hand gibt, müssen die Merkmale von der Ruheposition der Hand unabhängig sein.
- Geschwindigkeitsinvarianz: Da jeder Mensch unterschiedlich schnell gestikuliert und ein Einfluss der Stimmung auf die Gestikuliergeschwindigkeit zu erwarten ist, dürfen die Merkmale nicht geschwindigkeitsabhängig sein.

Abbildung 6.7 zeigt die für die Merkmalsextraktion notwendigen Schritte. Die Informationen des Videosignals (Pixelebene) werden über eine Zwischenebene, der Rechteckebene, abstrahiert und schließlich durch einen Merkmalsvektor repräsentiert.

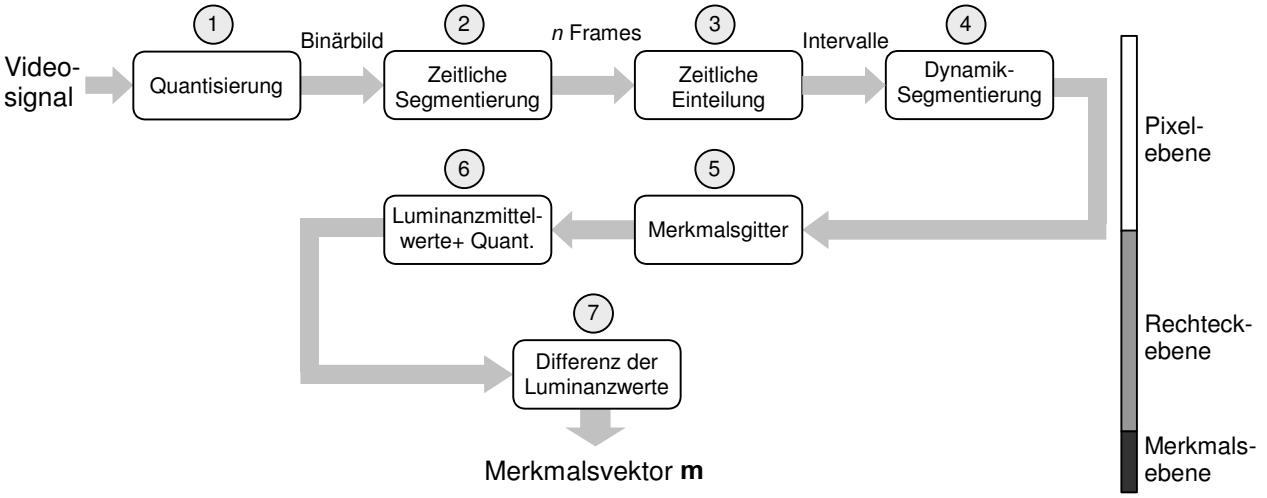


Abbildung 6.7: Bestimmung des Merkmalsvektors **m**

Die Merkmalsextraktion wird nun in den Schritten ① bis ⑦ im Detail erläutert:

① Bei dem Videosignal handelt es sich um eine Folge von Frames (Bildern) mit einer Auflösung von 356×288 Bildpunkten, deren Grauwert mit 8 Bit codiert wird, d.h., der Wertebereich der Luminanz beträgt 0 bis 255. Durch Quantisierung der Luminanzwerte werden aus den Grauwertbildern binäre Bilder erzeugt, um die Informationen zur Beschreibung der Benutzerhand von den Hintergrundinformationen zu trennen. Da davon ausgegangen wird, dass der Handhintergrund, verglichen mit der Hand, größtenteils dunkel ist, lässt sich die Hand im Allgemeinen sehr genau vom Hinter-

grund trennen. Mit einem Wertebereich von 0 bis 255 ergibt sich für den Luminanzwert $I(x, y, t)$ eines Pixels mit den Koordinaten (*x, y*) zum Zeitpunkt *t*, abhängig von dem Schwellwert 70, folgende Beziehung:

$$I(x,y,t) = \begin{cases} 0 & \text{für } I(x,y,t) < 70 \\ 255 & \text{für } I(x,y,t) \geq 70 \end{cases} \tag{6-1}$$

Gleichung (6-1) erzeugt aus jedem Frame ein binäres Bild. Weiße Bildpunkte mit einem Luminanzwert von 255 können der Hand zugeordnet werden, während schwarze Bildpunkte (Luminanz 0) dem Hintergrund entsprechen.

Ideale Bedingungen für die Merkmalsextraktion sind ein schwarzer Handhintergrund. Dass bei nicht idealen Rahmenbedingungen dennoch gute Erkennungsergebnisse erzielt werden können, wird in einem späteren Kapitel diskutiert.

② Die Aufgabe der zeitlichen Segmentierung ist die Analyse des kontinuierlichen Binär-Videosignals und die Extraktion der Frames, die eine abgeschlossene Handgeste darstellen. Hierzu werden alle binären Frames auf einen potenziellen Gestenbeginn und auf ein entsprechendes Gestenende untersucht. Da das Verfahren zur Klassifikation dynamischer Handgesten entwickelt wurde, wird lediglich geprüft, ob der aktuelle Frame bezüglich des vorhergehenden Frames ein gewisses Maß an Abweichung überschreitet, d.h. die zeitliche Segmentierung einer Bewegungsdetektion entspricht.

Für zwei aufeinander folgende Frames wird ein Differenzbild berechnet, indem der Betrag der Differenz aller Luminanzwerte des aktuellen Frames und des vorhergehenden Frames gebildet wird. Gleichung (6-2) beschreibt die Luminanzdifferenz $D(x, y, t)$ eines Pixels mit den Koordinaten (*x*,*y*) zum Zeitpunkt *t* und zum Zeitpunkt *t*-1:

$$D(x,y,t) = \left| I(x,y,t) - I(x,y,t-1) \right| \tag{6-2}$$

Diese Differenz ist für alle Pixels eines Frames zu bilden. Schließlich wird die Signalenergie des Differenzbilds I_{diff} durch Summieren aller Luminanzwerte $D(x, y, t)$ berechnet:

$$I_{\text{diff}} = \sum_{x=0}^{x_{\max}-1} \sum_{y=0}^{y_{\max}-1} D(x,y,t) \tag{6-3}$$

Die Werte $x_{\max}$ und $y_{\max}$ in Gleichung (6-3) beschreiben den rechten, unteren Bildrand.

Überschreitet die Energie des Differenzbilds I_{diff} den Schwellwert d_{motion}, so werden diese beiden Frames als Gestenbeginn und die folgenden Frames als Geste interpretiert. Entsprechend wird das Ende einer dynamischen Geste detektiert, wenn die Schwelle bei bereits erkanntem Gestenbeginn unterschritten wird. Auf diese Weise werden alle Frames ermittelt, die eine dynamische Geste als Helligkeitswerte darstellen. Folgende Abbildung veranschaulicht dies:

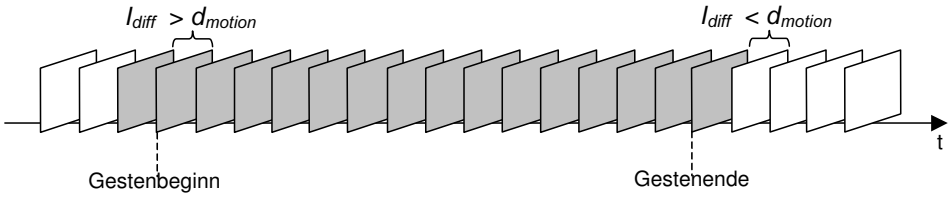


Abbildung 6.8: Bewegungsdetektion zur Ermittlung von Beginn und Ende einer Geste

Bei der Wahl der Bewegungsschwelle d_{motion} ist ein Kompromiss zu schließen. Setzt man den Schwellwert zu niedrig, ist die Bewegungsdetektion zu empfindlich und schon sehr geringfügige, unbewusste Bewegungen können die Merkmalsextraktion und damit den Klassifikationsprozess starten. Ein zu hoher Schwellwert kann bei langsamen Bewegungen zu Informationsverlust für die Klassifikation führen, weil der Gestenbeginn eventuell zu spät detektiert wird und somit einige Frames bei dem Erkennungsprozess unberücksichtigt bleiben. Für Handgesten hat sich ein Schwellwert d_{motion} von 3000 als geeignet erwiesen.

Nachdem in den ersten beiden Schritten das Videosignal quantisiert wurde und die Frames mit den für die Merkmalsextraktion interessanten Informationen bestimmt wurden, beginnt mit dem nächsten Schritt die eigentliche Merkmalsextraktion.

③ Da eine der Aufgaben der in *byHand* verwendeten Merkmale die zeitliche Modellierung ist, wird mit diesem Schritt damit begonnen, Bewegung und Zeit in Bezug zu setzen. Nach der zeitlichen Segmentierung ist die Dauer der Geste und somit die Anzahl der Frames bekannt. Diese Frames werden zunächst in einem Framespeicher zwischengespeichert, sodass alle Informationen über die Geste verfügbar sind. Aufgrund der Tatsache, dass nur binäre Bilder mit einer niedrigen Auflösung gespeichert werden müssen, führt dieser Schritt zu keinerlei Speicherproblemen.

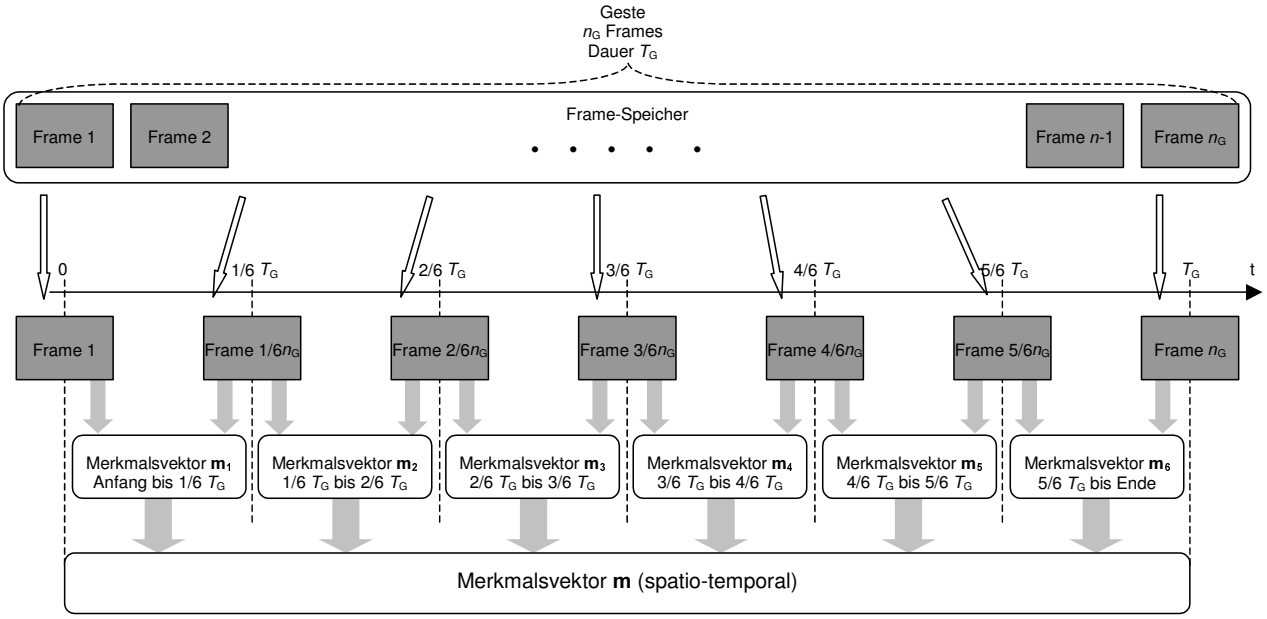


Abbildung 6.9: Aufteilung einer Geste in sechs zeitlich äquidistante Intervalle

Zur Erklärung wird von einer Geste mit n_G Frames und einer korrespondierenden Gestendauer von T_G ausgegangen. Die Geste wird in sechs äquidistante Zeitintervalle eingeteilt. Für jedes Zeitintervall wird später, unabhängig von den anderen Intervallen, ein Merkmalsvektor für die Beschreibung des entsprechenden Gestenausschnitts bestimmt. Aus dem ersten und dem letzten Frame eines solchen Intervalls werden die Merkmale berechnet. Alle sechs Merkmalsvektoren ergeben schließlich den Merkmalsvektor **m**, der die gesamte Geste repräsentiert und den Eingangsvektor für die Klassifikation bildet. Abbildung 6.9 zeigt diesen Zusammenhang in einer Übersicht.

Nach Ermittlung der relevanten sieben Frames kann der Framespeicher gelöscht werden, da alle weiteren Berechnungen nur noch auf Basis dieser sieben Bilder vollzogen werden.

④ Im Folgenden wird die Berechnung der Merkmale am Beispiel des ersten Zeitintervalls einer dynamischen Handgeste geschildert. Hierfür werden die beiden binären Frames herangezogen, die dem Anfang bzw. dem Ende des Intervalls entsprechen. Abbildung 6.10 zeigt diese beiden Frames, Frame 1 und Frame 1/6 n_G.

Abbildung 6.10: Vorverarbeitete Frames für die Merkmalsextraktion für das Zeitintervall 0 bis 1/6 T_G, d.h. Frame 1 und Frame 1/6 n_G

Die Bilder zeigen, dass sich der Handschwerpunkt auf Grund einer translatorischen Bewegung innerhalb dieses Zeitintervalls nach rechts verschoben hat. Zudem ist eine Drehung der Hand erkennbar, die sich aus Kameraperspektive durch eine Veränderung der Handform darstellt. Da dynamische Gesten klassifiziert werden sollen, ist ausschließlich der Bildbereich interessant, der auf eine Bewegung der Hand schließen lässt. Dieser Bereich wird durch Berechnung des Differenzbildes der beiden relevanten Frames nach Gleichung (6-2) bestimmt. Abbildung 6.11 zeigt das Differenzbild für die in Abbildung 6.10 dargestellten Frames.

Abbildung 6.11: Differenzbild von Frame 1 und Frame 1/6 n_G

Alle weißen Bildpunkte deuten auf Bewegung hin. Die Handkonturen beider Frames sind klar erkennbar; im Bereich der Schnittmenge beider Handformen ist wegen der Differenzbildung keine Bewegung auszumachen. Dies stellt keine Beeinträchtigung für die Merkmalsextraktion dar, da in diesem Schritt ausschließlich der Bildbereich ermittelt werden soll, in dem Bewegung herrscht.

Nun wird um die Bereiche, die auf eine Handbewegung hindeuten, ein Rechteck derart gelegt, dass es alle weißen Pixel beinhaltet, dabei aber einen möglichst kleinen Flächeninhalt einnimmt. Dieses Rechteck stellt eine Näherung an die örtliche Segmentierung der gesamten Dynamik innerhalb des betrachteten Zeitintervalls dar und bildet die Basis für die Bestimmung der spatio-temporalen Merkmale. Folgende Abbildung 6.12 zeigt das Rechteck für das in Abbildung 6.11 dargestellte Differenzbild.

Abbildung 6.12: Ortssegmentierung der Handdynamik innerhalb eines Zeitintervalls

⑤ Die Ortsinformationen über die Handdynamik werden nun verwendet, um die tatsächliche Handdynamik zu extrahieren. Hierzu werden die Koordinaten des Rechtecks zunächst in die beiden betrachteten Frames aus Abbildung 6.10 projiziert, um die relevanten Bildbereiche auszuschneiden. Diese werden in ein Gitter von 8×8 gleich großen Rechtecken aufgeteilt, wie Abbildung 6.13 zeigt.

Abbildung 6.13: Äquidistante Einteilung der Handdynamik in denen Frames, die den Beginn und das Ende eines Zeitintervalls beschreiben

Vergleicht man die korrespondierenden 64 Rechtecke der beiden Frames, so lässt sich direkt ableiten, inwiefern sich bestimmte Handausschnitte in dem betrachteten Zeitintervall verändern.

⑥ In diesem Schritt werden die Informationen der in einem Rechteck enthaltenen Pixel auf einen einzigen Wert reduziert. Für alle 64 Rechtecke, die durch das Gitter pro Frame entstehen, wird zunächst der Luminanzmittelwert $I_{\text{grid}}(r,t)$ der durch ein Rechteck r begrenzten Bildpunkte bestimmt:

$$I_{\text{grid}}(r,t) = \frac{1}{p_{\text{x}} p_{\text{y}}} \sum_{x_r=0}^{p_{\text{x}}-1} \sum_{y_r=0}^{p_y-1} I(x_{\text{or}} + x_r, y_{\text{or}} + y_r, t) \tag{6-4}$$

Die Werte x_{or} und y_{or} stehen für die Koordinaten der linken oberen Ecke des Rechtecks r; p_{x} und p_{y} geben die Anzahl der in einem Rechteck enthaltenen x- bzw. y-Werte an.

Für die folgende Quantisierung des Luminanzmittelwertes eines Rechtecks werden drei Helligkeitsstufen verwendet, d.h., für die diskretisierte Luminanz eines Rechtecks $R(r,t)$ gibt es einen Wertebereich von 0 bis 2. Die Bestimmung dieser Stufen erfolgt anhand folgender Luminanzgrenzen:

$$R(r,t) = \begin{cases} 0 & \text{für} \quad I_{\text{grid}}(r,t) < 100 \\ 1 & \text{für} \quad 100 \le I_{\text{grid}}(r,t) < 180 \\ 2 & \text{für} \quad I_{\text{grid}}(r,t) \ge 180 \end{cases} \tag{6-5}$$

Stufe 0 repräsentiert Rechtecke mit niedriger durchschnittlicher Helligkeit, Stufe 2 steht für helle Bildbereiche. Auf Grund der Tatsache, dass die Mittelung der Luminanzwerte auf Basis binärer Bilder berechnet wird, dient Stufe 1 der Modellierung von Handkanten. Abbildung 6.14 zeigt die Helligkeitsstufen $R(r,t)$ für die in Abbildung 6.10 dargestellten Frames. Stufe 0 wird durch ein schwarzes Rechteck visualisiert, Stufe 1 ist grau und Stufe 2 ist weiß dargestellt.

Abbildung 6.14: Reduktion der Bildpunkte aller 64 Rechtecke eines Bildes auf jeweils einen Wert

Mit Schritt ⑥ wird die Pixelebene verlassen und die Abstraktion der Handgeste nur noch auf Merkmalsgitterebene fortgesetzt.

⑦ Aus den Helligkeitsstufen $R(r,t)$ beider Frames werden nun direkt die Merkmale für ein Zeitintervall bestimmt. Hierfür werden die einzelnen Helligkeitsstufen des einen Frames mit den korrespondierenden Helligkeitsstufen des anderen Frames verglichen. Eine Beschreibung der Änderung der Helligkeitsstufen ist hierbei als Interpretation der Dynamik zu sehen. Die Helligkeitsänderungen werden bestimmt, indem die Helligkeitsstufen des zeitlich früheren Frames von den Helligkeitsstufen des anderen Frames subtrahiert werden. Für das u-te Zeitintervall bedeutet dies:

$$m_u(r) = R(r, \frac{u}{6} T_G) - R(r, \frac{u-1}{6} T_G) \tag{6-6}$$

Für die in Abbildung 6.10 dargestellten Frames ergibt sich der in Abbildung 6.15 gezeigte Merkmalsvektor. Der Vektor beinhaltet die Merkmale aller 64 Rechtecke des ersten Zeitintervalls. Zur besseren Übersicht wurden die einzelnen Einträge der Matrix visualisiert; die Legende rechts gibt an, in welchen Farben die Merkmale dargestellt sind.

$$\mathbf{m}_1 = \begin{bmatrix} 0 & 0 & -1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & -2 & 0 & -1 & 2 & 1 & 0 \\ -1 & 0 & -2 & 0 & 0 & 1 & 1 & 2 \\ -1 & 0 & -1 & -1 & 0 & 0 & 2 & 2 \\ 0 & -2 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & -1 & -2 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & -2 & -1 & 0 & 2 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 2 & 1 & 0 \end{bmatrix}$$

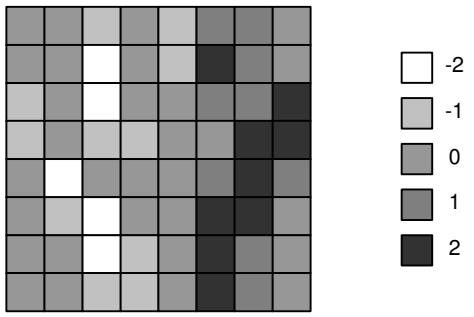


Abbildung 6.15: Merkmalsvektor $\mathbf{m}_1$ als Matrix und als grafische Darstellung

Die dunklen Rechtecke stehen für eine Steigerung der Intensität und lassen den Schluss zu, dass sich die Hand gerade in diese Bereiche hineinbewegt. Im Gegenzug repräsentieren die weißen Rechtecke Bildausschnitte, aus denen sich die Hand wegbewegt. Somit kann man aus dem Merkmalsvektor direkt eine translatorische Bewegung nach rechts ableiten. Auch die Veränderungen der Handform lassen sich ableiten, da der Merkmalsvektor keinerlei Symmetrien aufweist, im Falle einer rein translatorischen Bewegung aber ein annähernd symmetrischer Merkmalsraum zu erwarten ist.

Das beschriebene Procedere ist für die übrigen fünf Zeitintervalle analog durchzuführen. Um zu dokumentieren, wie gut die hier vorgestellte Merkmalsextraktion die charakteristischen Eigenschaften einer Geste modellieren, werden in Abbildung 6.16 die Merkmalsvektoren für drei unterschiedliche Gesten für jeweils drei verschiedene Gestentypen dargestellt.

Alle durch die Merkmalsvektoren aus Abbildung 6.16 repräsentierten Gesten wurden korrekt klassifiziert. Der Vergleich der Merkmalsvektoren der jeweils drei Gesten pro Gestentyp zeigt zum einen eine starke Korrelation der Merkmale für einen Gestentyp, zum anderen eine deutliche Diskrepanz der Merkmale der verschiedenen Gestentypen. Dies spricht für eine adäquate Repräsentation einer dynamischen Geste mittels der hier vorgestellten Merkmale.

Die Merkmalsvektoren der sechs Zeitintervalle werden zu dem Merkmalsvektor **m** zusammengefasst, der schließlich alle Informationen bezüglich einer Geste beschreibt.

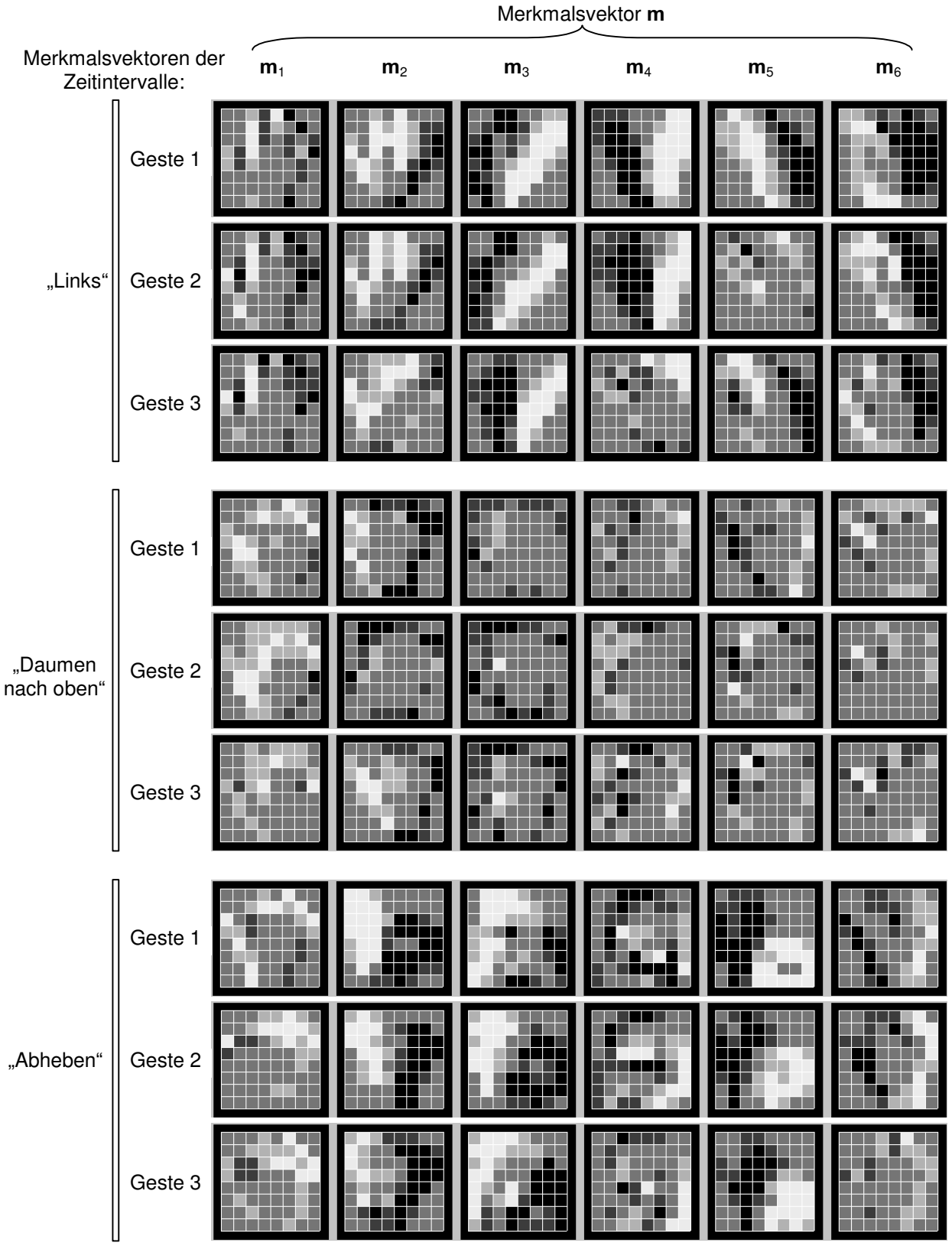


Abbildung 6.16: Merkmalsvektoren für drei verschiedene Gestentypen von jeweils drei verschiedenen Gesten

6.6 Intentionsmodell

Im Rahmen von *byHand* stellt ein Intentionsmodell den direkten Bezug zwischen allen Intentionshypothesen und dem Merkmalsvektor **m** her. Dieses Modell ist als Klassifikator des Gestikerkenners, neben der Merkmalsextraktion, die zentrale Komponente des Systems. Zunächst wird die grundliegende Idee hinter der Intentionsmodellierung für die Interpretation dynamischer Handgesten vorgestellt; schließlich wird auf die algorithmische Umsetzung eingegangen.

6.6.1 Struktur des Intentionsmodells

Die Tatsache, dass die Zeit bei der Interpretation dynamischer Handgesten eine wichtige Rolle spielt, schlägt sich auch auf die Struktur des Intentionsmodells nieder. Abbildung 6.17 zeigt die Struktur des Intentionsmodells anhand der in Kapitel 2 allgemein diskutierten drei Ebenen.

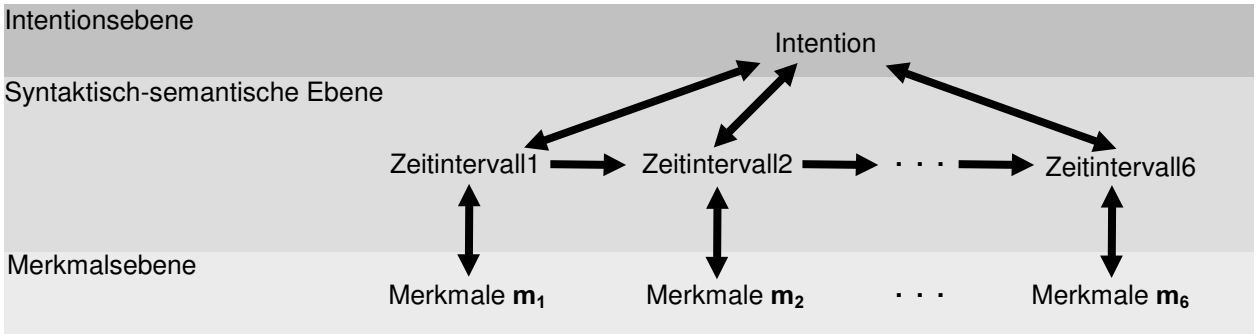


Abbildung 6.17: Struktur des Intentionsmodells von *byHand*

Das Intentionsmodell von *byHand* folgt einer Gliederung in drei Ebenen, analog zu den Intentionsmodellen der bereits vorgestellten Umsetzungen des intentionsbasierten Ansatzes.

- **Intentionsebene**
  Die Intentionsebene modelliert die komplette Intentionsbibliothek. Für die Bestimmung der Benutzerintention wird jede Intentionshypothese anhand eines Evaluierungsmaßes quantitativ bewertet. Die Interpretation erfolgt dabei diskriminativ, d.h., je wahrscheinlicher eine Intentionshypothese ist, desto unwahrscheinlicher sind die übrigen Alternativen. Die Intentionsebene fasst Klassifikationsergebnisse für alle einzelnen Zeitintervalle zusammen und erlaubt eine Gesamtaussage über den Hypothesenraum.

- **Syntaktisch-semantische Ebene**
  Die Semantikebene sorgt für korrektes Abbilden der Merkmalsvektoren $\mathbf{m_1}$ bis $\mathbf{m_6}$ auf den Intentionsraum in den jeweiligen Zeitintervallen, d.h., für jedes Zeitintervall gibt es einen eigenen Klassifikationsprozess. Dabei gehen die Ergebnisse dieser Teil-Klassifikationen quantitativ in die Klassifikationsprozesse für nachfolgende Zeitintervalle mit ein. Diese semantische Beziehung zwischen aufeinander folgenden Zeitintervallen wird durch eine entsprechende Netztopologie erzeugt.

- **Merkmalsebene**
  Die Merkmalsebene dient als Schnittstelle des Intentionsmodells zum Merkmalsvektor **m**, der gegliedert in die Vektoren $\mathbf{m_1}$ bis $\mathbf{m_6}$ direkt auf das Intentionsmodell abgebildet werden kann.

6.6.2 Realisierung des Intentionsmodells

Die in Kapitel 6.5 vorgestellten Merkmalsvektoren $\mathbf{m_1}$ bis $\mathbf{m_6}$ modellieren die durchgeführte Handgeste jeweils innerhalb eines Zeitintervalls. Eine adäquate Realisierung des Intentionsmodells für ein Zeitintervall *u* erfolgt durch das Bayes'sche Netz mit der in Abbildung 6.18 dargestellten Struktur. Diese besteht aus zwei Wurzelknoten, dem *Intentionsknoten* mit dem Namen *Intention* und dem *Zeitknoten* mit dem Namen *t*.

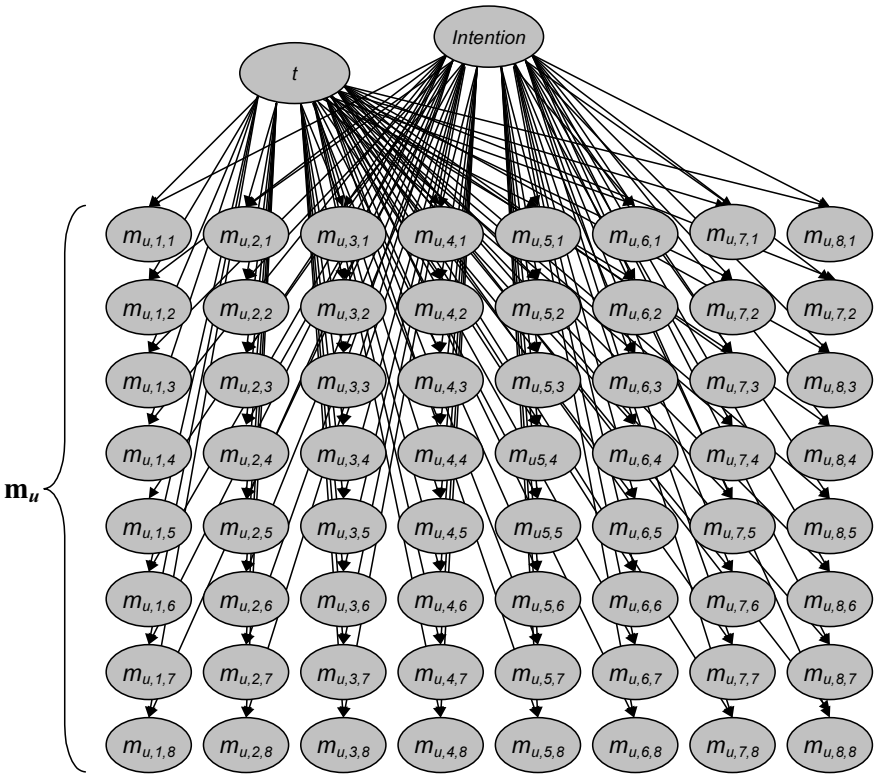


Abbildung 6.18: Topologie eines Bayes'schen Netzes zur Modellierung des *u*-ten Zeitintervalls

Alle Knoten des Netzes visualisieren diskrete Zustandsvariablen. Der Zustandsraum des Intentionsknotens berücksichtigt alle Intentionshypothesen anhand jeweils eines Zustands. Der Zeitknoten hält einen Zustand pro Zeitintervall bereit und jeder der 64 Merkmalsknoten verfügt über jeweils einen Zustand für jeden möglichen Merkmalswerte („-2", „-1", „0", „1" und „2"). Die Pfeile des Bayes'schen Netzes visualisieren die bedingten Wahrscheinlichkeiten qualitativ. Die quantitative Betrachtung erfolgt in Kapitel 6.7.

Das Netzwerk aus Abbildung 6.18 stellt zwar den Zusammenhang zwischen dem Merkmalsvektor und den Intentionshypothesen her, allerdings nur innerhalb des Zeitintervalls, das durch Setzen des Zeitknotens auf den entsprechenden Zustand angesprochen wird. Der Zeitknoten dient somit als Schalter, der in den bedingten Wahrscheinlichkeiten der Merkmalsknoten $P(m|Intention,t)$ die

Werte aktiviert, die den statistischen Zusammenhang zwischen den Merkmalen des gerade betrachteten Intervalls mit dem Hypothesenraum herstellen. Durch Abbilden des Merkmalsvektors auf die Merkmalsknoten können die Intentionshypothesen für das *u*-te Zeitintervall quantitativ evaluiert werden, indem die a-posteriori-Wahrscheinlichkeit $P(Intention|\mathbf{m}_u, t = \mathrm{t}_u)$ ausgewertet wird.

Bayes'sche Netze repräsentieren stets nur eine Momentaufnahme der zu modellierenden Problematik. Um alle Zeitintervalle in das Intentionsmodell mit einzubeziehen, ist ein konventionelles Bayes'sches Netz nicht ausreichend, da es nicht in der Lage ist, zeitlich bedingten Zustandsänderungen Rechnung zu tragen. Da die Klassifikationsergebnisse der einzelnen Zeitintervalle als eine Einheit betrachten werden müssen, wird für die Realisierung des Intentionsmodells von *byHand* ein *Dynamisches Bayes'sches Netz* [Rus94] herangezogen. Dabei handelt es sich nicht um einen neuen Netztyp, sondern lediglich um Erweiterungen in der Verwendungsweise eines Bayes'schen Netzes, die auch zeitlich bedingte Zustandsänderungen gestatten.

Das Dynamische Bayes'sche Netz wird als übergeordnete Struktur für das Ansprechen des in Abbildung 6.18 dargestellten Netzes verwendet. Folgende Abbildung 6.19 stellt die Topologie des Dynamischen Bayes'schen Netzes für alle sechs aufeinander folgenden Zeitintervalle, und damit für die Modellierung einer gesamte Geste, dar.

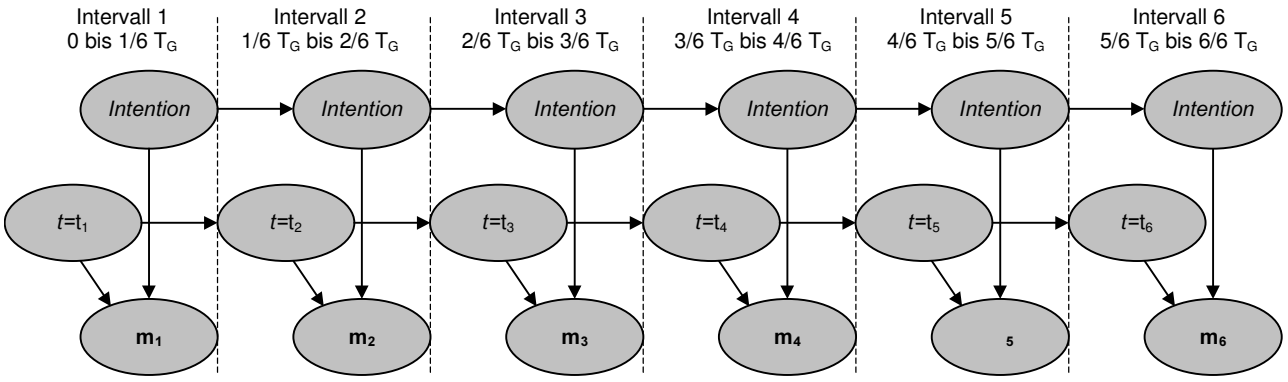


Abbildung 6.19: Realisierung des Intentionsmodells von *byHand* durch ein Dynamisches Bayes'sches Netz

Die Wahrscheinlichkeiten des Intentions- und des Zeitknotens sind nun von den Zuständen bzw. Wahrscheinlichkeiten des vorhergehenden Zeitintervalls abhängig. Die Modellierung eines Zeitintervall erfolgt anhand der in Abbildung 6.18 dargestellten Netztopologie. Die 8×8 Merkmalsknoten werden in dieser Darstellung aus Gründen der Übersichtlichkeit durch einen einzigen *Merkmalsvektorknoten* kompakt dargestellt.

Die a-priori-Wahrscheinlichkeit des Intentionsknotens *P(Intention)* wird für das erste Intervall neutral gewählt, sodass alle Intentionshypothesen zunächst als gleich wahrscheinlich gewertet werden. Die Abhängigkeit eines Intentionsknotens von dem zeitlich vorhergehenden Intentionsknoten wird derart gewählt, dass die Wahrscheinlichkeit von der zeitlich vorhergehenden Wahrscheinlichkeit des Intentionsknotens für alle Zustände exakt übernommen wird. Für den Intentionsknoten des zweiten Intervalls bedeutet dies, dass die a-posteriori-Wahrscheinlichkeit des ersten Intervalls $P(Intention|\mathbf{m} = \mathbf{m}_1, t = \mathrm{t}_1)$ als a-priori-Wahrscheinlichkeit $P(Intention)$ des zweiten Intervalls

übernommen wird. Auf diese Weise gehen die Klassifikationen aller vorhergehenden Zeitintervalle in die Klassifikation für das aktuelle Zeitintervall mit ein. Dies wird in Kapitel 6.8 detailliert erläutert. Für die einzelnen Zeitintervalle wird die Klassifikation sequenziell abgearbeitet und entspricht auf Grund der Tatsache, dass für jedes Zeitintervall ausschließlich Informationen des direkt vorhergehenden Intervalls in Betracht gezogen werden, einer Markov'schen Modellierung [Bro89].

Die bedingte Wahrscheinlichkeit der Zeitknotens sorgt dafür, dass diese Zustandsvariablen immer den Zustand zugewiesen bekommen, der das aktuelle Zeitintervall modelliert. Dadurch ist das Aktivieren der richtigen Werte der bedingten Wahrscheinlichkeit $P(\mathbf{m}|Intention, t = \mathrm{t_u})$ für das *u*-te Intervall gewährleistet.

Die für die Klassifikation entscheidenden Informationen bilden die bedingten Wahrscheinlichkeiten $P(\mathbf{m}|Intention, t = \mathrm{t_u})$ der Merkmalsknoten. Wie diese Werte bestimmt werden, wird im Folgenden Abschnitt erläutert.

6.7 Training und Adaption des Intentionsmodells

Wie jedes statistische Verfahren zur Musterkennung muss auch *byHand* Referenzmodelle erlernen. Um das Training des Intentionsmodells in Echtzeit zu erreichen, wurde bei der Topologie des Dynamischen Bayes'schen Netzes und bei der Bestimmung der Merkmale Wert darauf gelegt, dass das Training der bedingten Wahrscheinlichkeiten der Merkmalsknoten stets aus vollständigen Datensätzen erfolgt. Jeder Datensatz enthält Zuweisungen konkreter Zustände an den Intentionsknoten, den Zeitknoten und die 64 Merkmalsknoten. Dies ermöglicht die Berechnung der bedingten Wahrscheinlichkeiten anhand von Häufigkeiten und macht rechenintensive iterative Trainingsalgorithmen überflüssig.

Bei dem Gestikerkenner *byHand* werden zwei grundsätzlich verschiedene Lernmodi unterschieden:

- Training des Intentionsmodells anhand von Datensätzen verschiedener Gesten und unterschiedlicher Benutzer
- Adaption des Intentionsmodells an einen spezifischen Benutzer

Beide Verfahren werden nun im Detail vorgestellt.

6.7.1 Training des Intentionsmodells

Wie Abbildung 6.20 zeigt, werden ausgehend von *n* Beobachtungsfolgen entsprechend viele Datensätze erzeugt, die aus den Merkmalsvektoren der ausgeführten Gesten sowie aus deren Bedeutung, der Intention, bestehen. Da *byHand* einen statistischen Ansatz verfolgt, sollte die Anzahl der Datensätze möglichst groß gewählt werden, um ein repräsentatives Intentionsmodell zu erhalten. Die Datensätze wurden erzeugt, indem vier Versuchspersonen aufgefordert wurden, jeden Gestentyp der Intentionsbibliothek zwanzig Mal auszuführen.

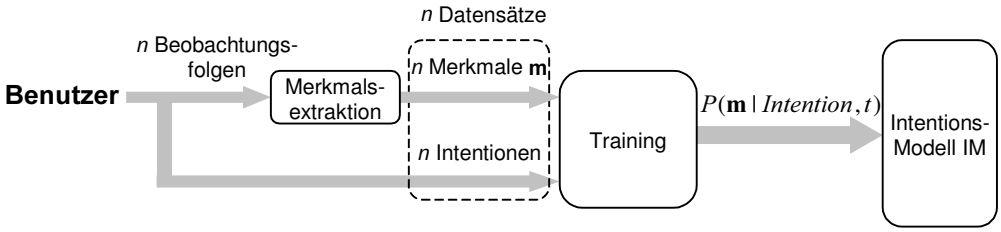


Abbildung 6.20: Überblick über das Training des Intentionsmodells

Auf Grund der einfachen Topologie des Bayes'schen Netzes können die bedingten Wahrscheinlichkeiten der Merkmalsknoten für ein Zeitintervall auf Basis von Häufigkeiten berechnet werden.

Das Training wird anhand des *r*-ten Merkmals im *u*-ten Zeitintervall erläutet. Alle Werte der bedingten Wahrscheinlichkeit $P(m_r \mid Intention = \mathrm{I_i}, t = \mathrm{t_u})$ dieses Merkmal werden bestimmt, indem zunächst die Datensätze ausgewählt werden, deren Intentions- und Zeit-Zustandsvariable mit denen des Wahrscheinlichkeitswertes übereinstimmen, d.h. (*Intention*=$\mathrm{I_i}$) und (t=$\mathrm{t_u}$). Aus den resultierenden $n_{u,i}$ Datensätzen werden die Datensätze ermittelt, deren Merkmalsknotenzustände dem Zustand $\mathrm{d_R}$ des zu berechnenden Wertes gleichen. Die Anzahl dieser Datensätze $n_{u,i.r,d}$ wird durch $n_{u,i}$ dividiert, um den gewünschten Wahrscheinlichkeitswert zu erhalten:

$$P(m_r = \mathrm{d_R} \mid Intention = \mathrm{I_i}, t = \mathrm{t_u}) = \frac{n_{u,i,r,d}}{n_{u,i}} \tag{6-7}$$

Diese Berechnung ist für alle Werte der bedingten Wahrscheinlichkeiten aller Merkmalsknoten durchzuführen. Selbst bei mehreren tausend Datensätzen lässt sich dieses Training auf einem Mittelklasse-PC in Echtzeit vollziehen.

6.7.2 Online-Adaption des Intentionsmodells an einen neuen Datensatz

Abbildung 6.21 zeigt das Schema für die Online-Adaption des Intentionsmodells an einen neuen Datensatz. Das Ziel ist die Adaption des Intentionsmodells an die für einen Benutzer typische Art zu gestikulieren. Aus der Benutzeraktion wird der Merkmalsvektor berechnet, der zusammen mit der Intention die Eingangsgröße für die Trainingskomponente bildet. Im Gegensatz zum Trainingsmodus trägt der Adaptionsmodus dem aktuellen Zustand des Intentionsmodells Rechnung, indem die aktuellen bedingten Wahrscheinlichkeiten der Merkmalsknoten in die Adaption mit einbezogen werden. Ein weiterer Unterschied liegt in der Tatsche, dass nur ein einziger Datensatz für das Training verwendet wird, nämlich der Datensatz, der die zuletzt durchgeführte dynamische Handgeste modelliert.

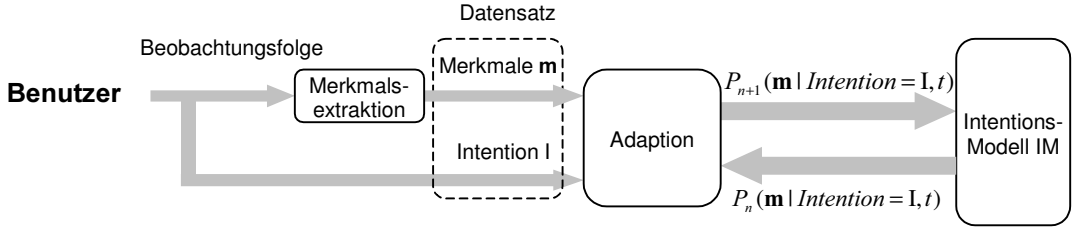


Abbildung 6.21: Überblick über die Online-Adaption des Intentionsmodells an einen neuen Datensatz

Abhängig vom aktuellen Systemzustand, dessen Wahrscheinlichkeiten $P_n(\cdot)$ mit dem Index n beschrieben werden, ist das Intentionsmodell an den neuen Datensatz anzupassen. Der resultierende Systemzustand wird durch $P_{n+1}(\cdot)$ dargestellt. In Analogie zu Gleichung (5-11) bildet folgende Berechnungsvorschrift die mathematische Grundlage für die Adaption:

$$P_{n+1}(m_r = \mathrm{d_R} \mid \mathit{Intention} = \mathrm{I_i}, t = \mathrm{t_u}) = \\ = \frac{P_{\mathit{Data}}(m_r = \mathrm{d_R} \mid \mathit{Intention} = \mathrm{I_i}, t = \mathrm{t_u}) + w_{\mathit{Data}} \cdot P_n(m_r = \mathrm{d_R} \mid \mathit{Intention} = \mathrm{I_i}, t = \mathrm{t_u})}{1 + w_{\mathit{Data}}} \quad (6\text{-}8)$$

Der Wert $P_{\mathit{Data}}(m_r = \mathrm{d_R} \mid \mathit{Intention} = \mathrm{I_i}, t = \mathrm{t_u})$ wird direkt dem Datensatz entnommen und ist entweder gleich 0 oder 1. Mit dem Gewichtungsfaktor w_{Data} lässt sich der Einfluss des neuen Datensatzes auf den aktuellen Zustand des Intentionsmodells festlegen. Ein Gewichtungsfaktor von 10 hat sich als geeignet erwiesen, um eine sinnvolle Lernkurve zu gewährleisten. Gleichung (6-8) ist auf alle Merkmalsknoten aller Merkmalsvektoren ($\mathbf{m_1}$ bis $\mathbf{m_6}$) anzuwenden.

6.8 Intentionsbasierte Interpretation

Die intentionsbasierte Interpretation der Merkmale stellt den eigentlichen Prozess zur Klassifikation einer dynamischen Handgeste dar. Hierzu wird der Merkmalsvektor auf das trainierte Intentionsmodell abgebildet um somit eine quantitative Evaluierung der Intentionsbibliothek bzw. des Intentionshypothesenraums zu ermöglichen. Der Merkmalsvektor **m** wird in die sechs Merkmalsvektoren ($\mathbf{m_1}$ bis $\mathbf{m_6}$) aufgeteilt, die den einzelnen Zeitintervallen zugeordnet sind. Diese Merkmalsvektoren werden nun sequenziell entsprechend ihrer zeitlichen Reihenfolge auf die 8×8 Merkmalsknoten des Intentionsmodells abgebildet, indem den Zustandvariablen ein konkreter Zustand zugewiesen wird.

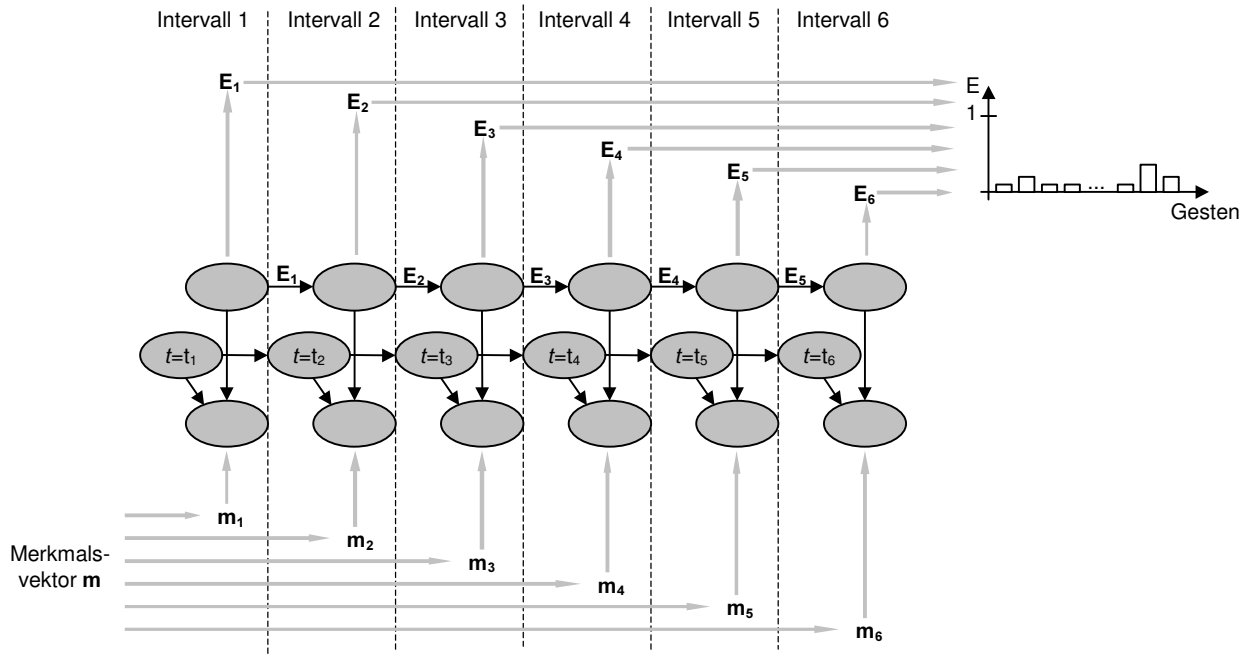


Abbildung 6.22: Klassifikation von dynamischen Handgesten anhand des Intentionsmodells

Abbildung 6.22 zeigt die Verwendung des Intentionsmodells für die intentionsbasierte Interpretation. Zur Modellierung des ersten Zeitintervalls werden die Merkmalsknoten entsprechend des Merkmalsvektors $\mathbf{m_1}$ instantiiert, d.h., den Merkmalsknoten werden die Zustände des Merkmalsvektors zugewiesen. Der Zeitknoten gewährleistet die Aktivierung der Werte der bedingten Wahrscheinlichkeiten $P(m_r \mid Intention, t = \mathrm{t}_1)$, die das erste Zeitintervall repräsentieren. Mittels dieser Werte erfolgt der Rückschluss auf den Intentionsknoten. Die a-posteriori-Wahrscheinlichkeit $P(Intention \mid \mathbf{m}_1, t = \mathrm{t}_1)$ ist das Ergebnis des Klassifikationsprozesses für das erste Intervall. Sie entspricht dem Evaluierungswert für das erste Intervall:

$$\mathbf{E_1} = P(Intention \mid \mathbf{m}_1, t = \mathrm{t}_1) \tag{6-9}$$

Bei $\mathbf{E_1}$ handelt es sich um einen Vektor, da jede Intentionshypothese einen Wert zugewiesen bekommt. Die Hypothese mit dem größten Evaluierungswert entspricht der wahrscheinlichsten Geste im ersten Zeitintervall.

Im nächsten Schritt wird $\mathbf{E_1}$ als a-priori-Wahrscheinlichkeit $P(Intention)$ des folgenden Intervalls weiterverarbeitet. Der Zeitknoten erhält den Zustand des zweiten Intervalls. Mit Abbildung des Merkmalsvektors $\mathbf{m_2}$ auf die Merkmalsknoten verändert sich die a-posteriori-Wahrscheinlichkeit $P(Intention \mid \mathbf{m}_2, t = \mathrm{t}_2)$. Zu betonen ist, dass diese Wahrscheinlichkeit sowohl das erste als auch das zweite Intervall berücksichtigt. Dieses Procedere ist nun für die weiteren Intervalle ebenfalls durchzuführen. Generell gilt für die Berechnung des Evaluierungsvektors für das *u*-te Zeitintervall:

$$\mathbf{E_u} = P(Intention \mid \mathbf{m}_u, t = \mathrm{t}_u) \tag{6-10}$$

Schließlich werden die Evaluierungswerte für alle Intentionshypothesen über die Zeitintervalle gemittelt:

$$\mathbf{E} = \frac{1}{6}\sum_{u=1}^{6} \mathbf{E_u} \tag{6-11}$$

Die Geste mit dem maximalen Evaluierungswert ist die wahrscheinlichste Intention und somit das Ergebnis des Erkennungsprozesses.

6.9 Ergebnisse und Diskussion

Dieser Abschnitt dokumentiert die Klassifikationsraten von *byHand* für die verschiedenen Intentionsbibliotheken. Für bestimmte Gestentypen wurden zudem weitere Untersuchungen durchgeführt, die in den folgenden Unterkapiteln diskutiert werden.

Die Erkennungsleistung von *byHand* für dynamische Handgesten wurde zunächst mit benutzerspezifisch trainiertem Intentionsmodell bestimmt. Der Trainingskorpus bestand hierfür aus 20 Gesten pro Intentionshypothese, wobei alle Datensätze von einer Person stammten, um das Intentionsmodell für eine bestmögliche Erkennungsleistung zu personalisieren. Mit jeweils 15 Gesten pro Intentionshypothese als Testdaten ergab sich eine Klassifikationsrate von 97,3 Prozent. Zu berücksichti-

gen ist, dass sowohl Trainingsdaten und Testdaten unter identischen Voraussetzungen aufgenommen wurden, d.h., sowohl die Lichtverhältnisse als auch die Kleidung des Benutzers waren in beiden Situationen identisch.

Neben der Evaluierung von *byHand* anhand eines personalisierten Intentionsmodells wurden die bedingten Wahrscheinlichkeiten des Verfahrens zudem auf vier Personen mit unterschiedlicher Kleidung bei geringfügig unterschiedlichen Lichtverhältnissen erlernt. Da zwei der Personen Kleidung mit hellem Ärmel trugen, wird neben der Hand auch der Unterarm durch den Merkmalsvektor modelliert. Die Kleidung der anderen beiden Personen war zu dunkel, um im Merkmalsraum berücksichtigt zu werden. Somit wurde das Intentionsmodell auf vier verschiedene Personen für unterschiedliche Rahmenbedingungen trainiert. Das auf diese Weise trainierte System wird als *unpersonalisiert* bezeichnet. In Abbildung 6.23 wird die Erkennungsrate des unpersonalisierten Intentionsmodells mit der Klassifikationsleistung des personalisierten Systems verglichen.

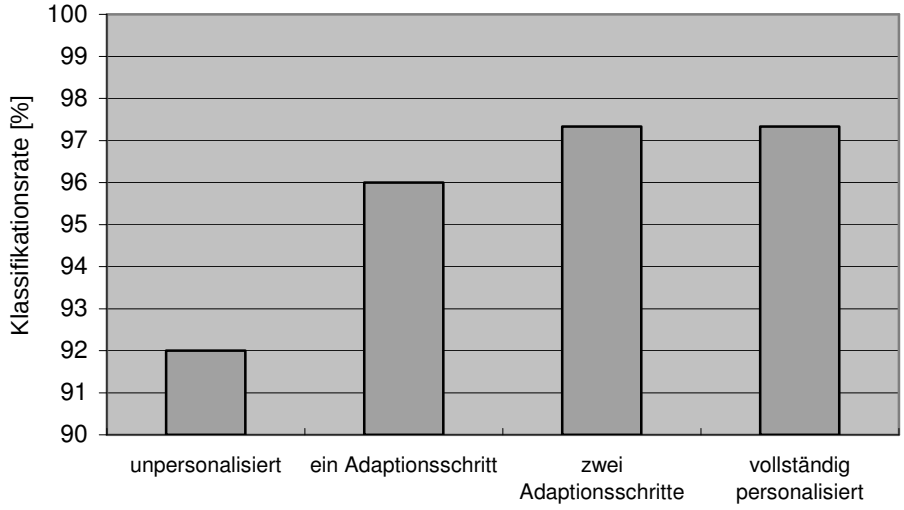


Abbildung 6.23: Vergleich der Klassifikationsleistung eines unpersonalisierten Intentionsmodells mit adaptierten Intentionsmodellen

Die linke Säule gibt die Erkennungsrate für das auf vier Personen trainierte Verfahren an. Mit 92 % liegt sie zwar erwartungsgemäß niedriger als für das personalisierte System (rechte Säule, 97,3 %), befindet sich aber dennoch auf einem sehr hohen Niveau. 42 % der Fehlerkennungen des Systems sind auf eine bestimmte Geste zurückzuführen. Dies lässt darauf schließen, dass der Benutzer, dessen Aktionen zur Evaluierung von *byHand* herangezogen wurden, eine eigene, charakteristische Art hat, diese spezielle Geste durchzuführen. Gerade in diesem Fall ist ein Gewinn bringender Einsatz der Adaption des Intentionsmodells an den aktuellen Benutzer zu erwarten.

Die beiden mittleren Säulen der Abbildung 6.23 stellen die Klassifikationsraten des personalisierten Systems an den aktuellen Benutzer nach einem bzw. zwei Adaptionsschritten dar. Für die Adaption wurde für den Gewichtungsfaktor w_{Data} der Wert 3 gewählt, um eine schnelle Anpassung der bedingten Wahrscheinlichkeiten an den aktuellen zu gewährleisten. Ein Adaptionsschritt entspricht dabei der Anpassung des Intentionsmodells an jede einzelne Geste der Intentionsbibliothek. Mit dem ersten Adaptionsschritt kommt es bereits zu einer erheblichen Steigerung der Erkennungsrate

von 92 auf 96 Prozent. Durch einen weiteren Lernschritt erhöht sich die Klassifikationsrate schließlich auf 97,3 Prozent. Dieser Wert entspricht gerade der Erkennungsrate des vollständig personalisierten Systems und ist damit die theoretische, obere Grenze für eine Adaption. Dies lässt darauf schließen, da *byHand*, ausgehend von einem auf mehrere Personen trainierten Intentionsmodells, innerhalb weniger Adaptionsschritte vollständig an den aktuellen Benutzer angepasst werden kann. Diese Eigenschaft ist vor allem für den praktischen Einsatz des Gestikerkenners zur Steuerung einer Applikation von entscheidender Bedeutung. Die Tatsache, dass die Adaption an einen spezifischen Benutzer bereits nach wenigen Schritten erfolgt, ist auf die Struktur des Intentionsmodells zurückzuführen. Die Merkmalsknoten, deren bedingte Wahrscheinlichkeiten für die Klassifikation maßgeblich sind, sind nicht voneinander, sondern lediglich vom Intentionsknoten und vom Zeitknoten abhängig. Aus diesem Grund setzen sich diese Wahrscheinlichkeiten nur aus wenigen Werten zusammen, da die Zustandsräume des Intentionsknotens und des Zeitknotens nur eine überschaubare Anzahl an Zustandskonstellationen zulassen. Um für diese Konstellationen repräsentative Wahrscheinlichkeiten zu berechnen, sind somit nicht viele Datensätze möglich. Wären alle Merkmalsknoten miteinander vernetzt, so wären schätzungsweise einige hundert Adaptionsschritte nötig, um das Klassifikationsergebnis so deutlich wie in Abbildung 6.23 zu verbessern. Die hohe Klassifikationsleistung lässt auf Grund des einfach strukturierten Intentionsmodells auch auf sehr aussagekräftige Merkmale schließen.

Wegen der verwendeten Merkmale ist eine starke Abhängigkeit der Erkennungsrate von der Kleidung des Benutzers zu erwarten, da im Falle heller Kleidung der Unterarm modelliert wird, bei dunkler Kleidung aber unberücksichtigt bleibt. Die Ergebnisse einer Untersuchung des Einflusses der Kleidung auf die Erkennungsleistung von *byHand* werden im Folgenden dargestellt.

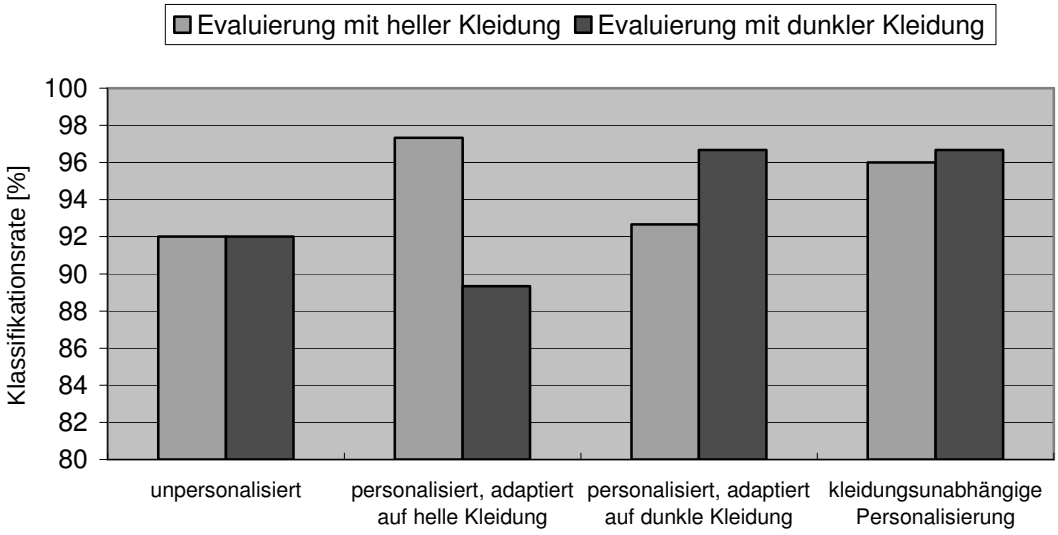


Abbildung 6.24: Untersuchung des Einflusses von Kleidung auf die Klassifikation

Da die Merkmale sehr stark mit der Kleidungsfarbe korrelieren, wurde ihr Einfluss auf die Erkennungsrate für den Fall, dass das Intentionsmodell auf Basis von Gesten, die mit heller bzw. dunkler Kleidung durchgeführt wurden, untersucht. Die Klassifikationsraten sind in Abbildung 6.24 dargestellt. Wie zu erwarten war, liegen im unpersonalisierten Fall die Erkennungsraten für beide Kleidungsarten auf dem gleichen Niveau, da die Datensätze zu gleichen Anteilen mit heller bzw. dunk-

ler Armbekleidung erzeugt wurden. Die Adaption des Intentionsmodells an einen bestimmten Benutzer birgt immer die Gefahr einer ausschließlichen Berücksichtigung der zum Zeitpunkt der Adaption getragenen Kleidung. Aus diesem Grund wurde die Korrelation von Kleidung und Erkennungsrate für drei unterschiedlich personalisierte Intentionsmodelle untersucht.

Das zweite Säulenpaar in Abbildung 6.24 visualisiert die Klassifikationsraten von *byHand* mit einem personalisierten Intentionsmodell, dessen Adaption auf Daten heller Kleidung basiert. Die Evaluierung zeigt einen deutlichen Abfall der Erkennungsleistung im Falle dunkler Kleidung. Auch das mit dunkler Kleidung adaptierte System erweist sich als kleidungsabhängig. Erwartungsgemäß lässt sich der Einfluss der Kleidungsfarbe in Grenzen halten, wenn die Adaption zu gleichen Anteilen auf Basis heller sowie dunkler Armbekleidung erfolgt. In diesem Fall zeichnet sich *byHand* durch eine robuste Klassifikation mit einer gemittelten Erkennungsrate von 96,3 Prozent bei weitgehender Kleidungsunabhängigkeit aus.

Generell liegen die Erkennungsraten für dynamische Vollhandgesten auf einem sehr hohen Niveau, was unterstreicht, dass es sich bei dem Evaluierungsmaß der Intentionshypothese um ein genügend aussagekräftiges Maß für die Klassifikation handelt. Belegt wird dies auch durch Abbildung 6.25, in der die Evaluierungsmaße aller Intentionshypothesen für den Fall von 10 ausgeführten *Daumen-oben*-Gesten dargestellt sind. In jedem Fall wurde die Geste korrekt klassifiziert. Das Evaluierungsmaß der tatsächlich ausgeführten Geste ist jeweils deutlich höher als die Werte der konkurrierenden Hypothesen, was schließlich zu einer klaren Entscheidung für die richtige Geste führt.

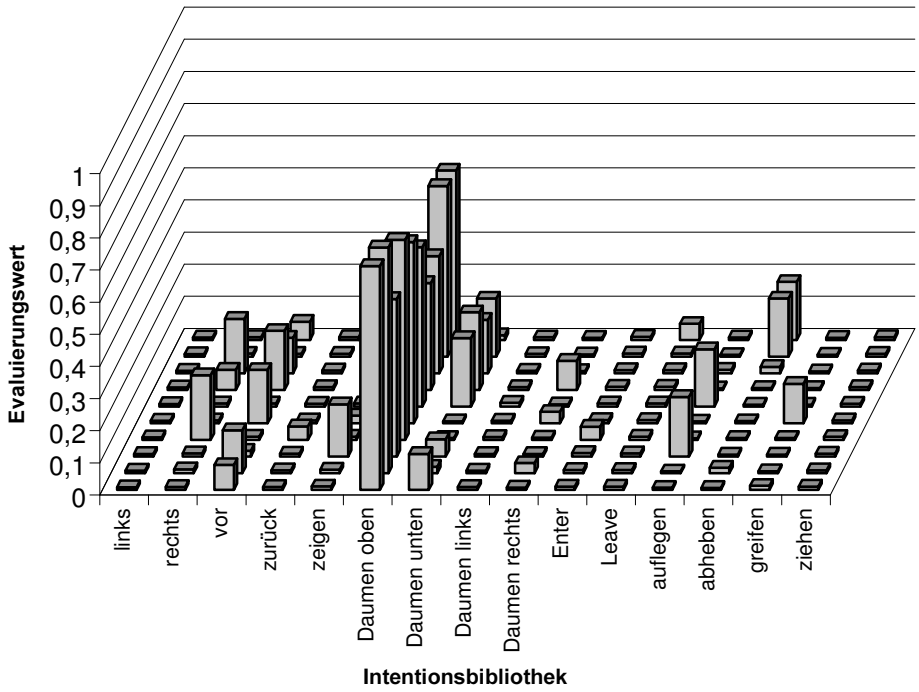


Abbildung 6.25: Evaluierungsmaße für 10 *Daumen-oben*-Gesten

Die Schreibgesten wurden ausschließlich anhand eines personalisierten Intentionsmodells evaluiert. Abbildung 6.26 zeigt die Klassifikationsraten für unterschiedlich optimierte Intentionsmodelle. Das erste Säulenpaar bezieht sich auf ein personalisiertes und auf helle Kleidung adaptiertes System. Anhand von 180 Testdatensätzen, Schreibgesten eines Benutzers mit heller Kleidung, ergab sich

eine Erkennungsrate von 88,9 Prozent. Testdaten für dunkle Armbekleidung führen zu einem Einbruch der Klassifikationsrate um 13,3 Prozent. Entsprechende Tests mit einem auf dunkle Kleidung optimierten System ergaben ebenfalls eine starke Korrelation der Erkennungsleistung mit der Helligkeit der Armbekleidung.

Generell erweisen sich Schreibgesten somit als erheblich kleidungsabhängiger als Vollhandgesten. Dies ist darauf zurückzuführen, dass sich einige Schreibgesten zum Teil nur gering voneinander unterscheiden. Das Zeichnen der „6“ kann zum Beispiel auch als ein unsauber geschriebenes „C“ (Clear) interpretiert werden. Eine andere typische „Verwechslung“ kann bei der „0“ und einer sehr schnell, unsauber geschriebenen „8“ auftreten. Da die Intentionshypothesen teilweise sehr ähnlich sind, führt gerade für das Intentionsmodell unbekannte Kleidung häufig zu Fehlklassifikationen.

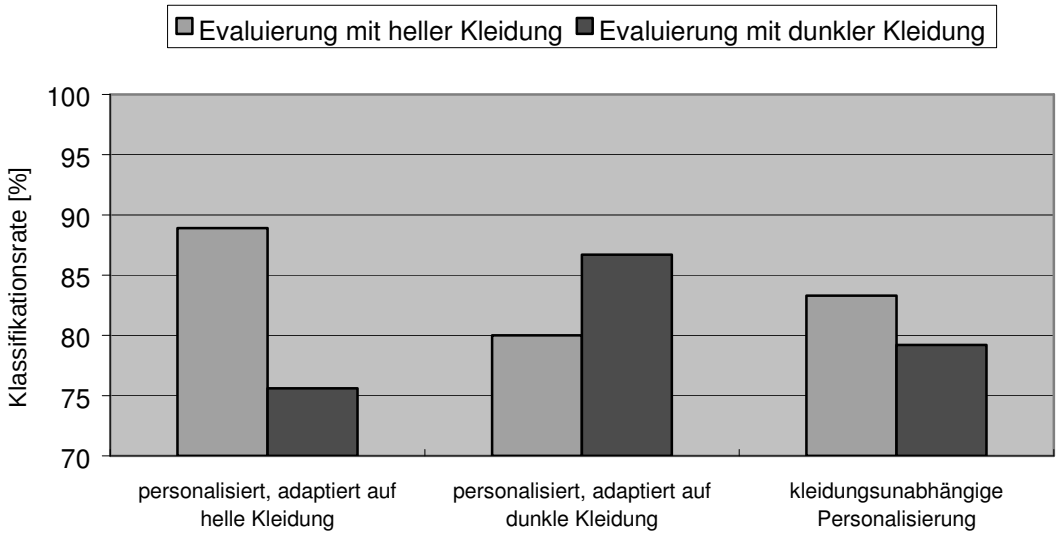


Abbildung 6.26: Klassifikationsraten für Schreibegesten unter unterschiedlichen Rahmenbedingungen

Das dritte Säulenpaar in Abbildung 6.26 zeigt die Erkennungsraten für ein personalisiertes, auf helle und im gleichen Maße dunkle Kleidung trainiertes System. Die Erkennungsraten beider Kleidungsarten zeigen die zu erwartende Tendenz, dass sie in etwa den Mittelwerten der Erkennungsraten der optimierten Intentionsmodelle entsprechen. Gemittelt über beide Kleidungsarten ergibt sich für das kleidungsunabhängige System eine Erkennungsrate von 81,25 Prozent.

Generell liegen die Erkennungsraten für dunkle Kleidung auf einem etwas niedrigeren Niveau als für helle Kleidung. Dies ist darauf zurückzuführen, dass der dunkle Ärmel im Laufe einer dynamischen Geste, zum Beispiel bei sich nach vorne streckender Hand, verglichen mit der Hand nach hinten ziehen kann. Dieses Phänomen kann aus Kamerasicht zu einer Verformung der Hand führen und somit die Klassifikation beeinträchtigen. Vermeidet man dieses Phänomen, zum Beispiel durch Fixieren der Kleidung, sind Erkennungsraten wie bei heller Kleidung zu erwarten.

Verglichen mit einem auf eine bestimmte Kleidungsart optimiertes Intentionsmodell, stellt eine kleidungsunabhängige Modellierung einen Kompromiss dar. Auf Basis der Klassifikationsraten aus Abbildung 6.26 ist jedoch eine explizite Segmentierung der Hand empfehlenswert, um eine tatsächliche Unabhängigkeit der Klassifikation von der Armbekleidung zu gewährleisten. In diesem Fall

beziehen sich die Merkmale ausschließlich auf die Hand, da alle anderen Informationen aus den Frames herausgefiltert werden. Dies trifft dann auch auf karierte Kleidung zu, die hier nicht untersucht wurde. Die durch Handsegmentierung zu erwartende Klassifikationsrate entspricht den 88,9 Prozent des auf helle Kleidung optimierten Systems.

Für die Evaluierung der dynamischen Schreibgesten für Mobilgeräte wurde das Intentionsmodell personalisiert und schließlich anhand von 130 Testdatensätzen getestet. Die resultierende Klassifikationsrate liegt bei 86,9 Prozent. Dies ist ein sehr hohes Niveau, gemessen an dem starken Einfluss der Handhaltung und des Abstands der Hand zur Kamera. Abbildung 6.27 dokumentiert dies. Dargestellt sind Ausschnitte zweier dynamischer Gesten zum Schreiben einer „1“. Im oberen Fall zeichnet der Benutzer mit ausgestrecktem Zeigefinger die Trajektorie einer „1“, wobei ausschließlich die Dynamik des Zeigefingers durch die Merkmale repräsentiert wird, da die übrigen Handbereiche auf Grund ihrer Entfernung weniger stark durch die Infrarotdioden beleuchtet werden und somit nach Infrarotfilterung und Quantisierung nicht in den binären Bildern visualisiert werden. Im unteren Fall ist der Zeigefinger angewinkelt, was eine kürzere Entfernung der übrigen Finger zur Kamera bewirkt. Deshalb werden neben dem Zeigefinger weitere Bereiche der Hand von den Infrarot-Dioden stark genug beleuchtet, um nach der Quantisierung der Frames als helle Bildbereiche in der Merkmalsextraktion berücksichtigt zu werden.

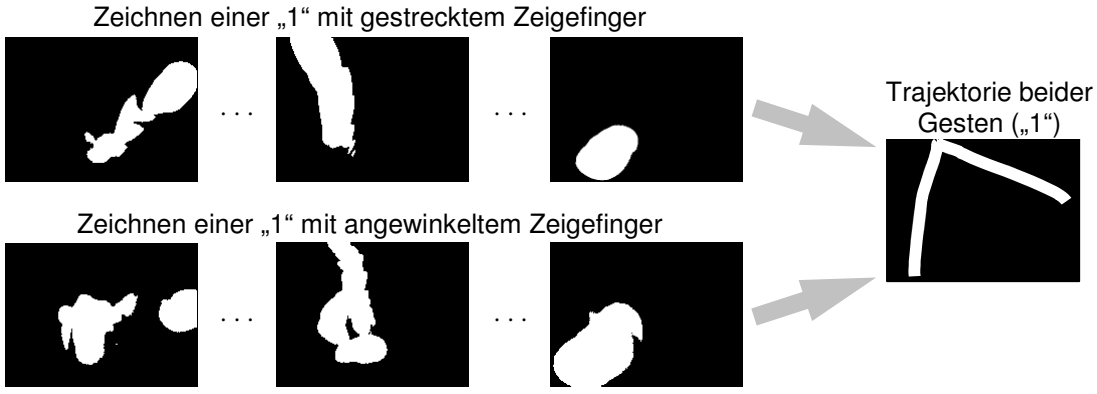


Abbildung 6.27: Einfluss der Handhaltung auf die Bildsequenz

Abbildung 6.27 zeigt in der rechten Darstellung eine typische Trajektorie der gezeichneten „1“. Da das Bild aus Kameraperspektive zu interpretieren ist, handelt es sich um eine gespiegelte „1“. Auf Grund des fehlenden haptischen und visuellen Feedbacks sind die Trajektorien im Allgemeinen sehr undeutlich und können sehr stark vom Referenzmuster des zu zeichnenden Symbols abweichen. Ein weiteres Problem ist die Gefahr, während des Schreibens das Kamerasichtfeld zu verlassen. Diese beiden Phänomene machen eine Einweisung in die Benutzung eines mobilen Gerätes mittels Schreibgesten sinnvoll, da der Benutzer weder das Kamerasichtfeld verlassen noch in zu großer oder zu kleiner Entfernung von der Kamera gestikulieren sollte.

6.10 Implementierung von *byHand*

Implementiert wurde *byHand* unter Linux in C++ [Str92] auf einem Pentium-II-PC mit 400 Mhz Taktfrequenz. Abbildung 6.28 zeigt die dem implementierten System zugrunde liegende Architektur.

Ein Framegrabber erzeugt aus dem Kamerasignal eine Folge von Frames mit einer Auflösung von 384×288 Bildpunkten, die von der Merkmalsextraktion analysiert wird. Die Merkmalsextraktion wurde als Client realisiert und sendet die resultierenden Merkmalsvektoren an einen Server, in dem das Intentionsmodell, die Trainings- und Adaptionskomponente sowie die intentionsbasierte Interpretation umgesetzt sind.

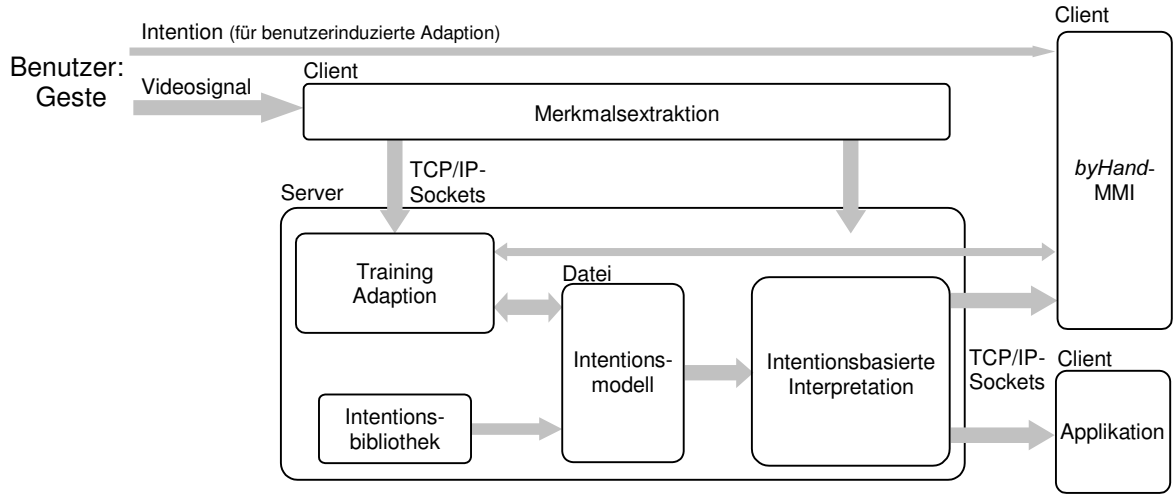


Abbildung 6.28: Systemarchitektur der Implementierung von *byHand*

Zur Steuerung und Überwachung von *byHand* wurde eine Schnittstelle in Tcl/Tk [Wel99] als Client implementiert. Abbildung 6.29 zeigt einen Screenshot dieser Komponente.

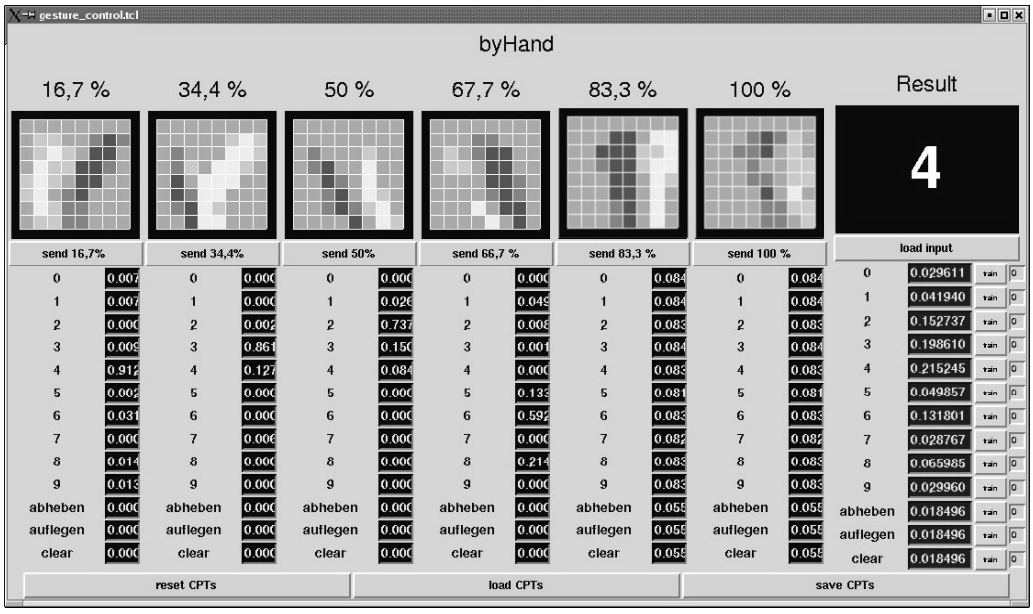


Abbildung 6.29: MMI zur Steuerung von *byHand*

Diese Oberfläche stellt die Merkmalsvektoren grafisch dar und gibt Auskunft über alle Teilevaluierungswerte sowie über die Gesamtevaluierungswerte aller Intentionshypothesen. Darüber hinaus können Adaptionsprozesse getriggert, die dazu benötigte Intention angegeben und die Adaptionsgeschwindigkeit eingestellt werden.

Ebenfalls als Client in Tcl/Tk wurden diverse Simulationen von Endapplikationen realisiert. Dabei dienen diese Komponenten eher der anschaulichen Visualisierung einer Systemreaktion auf eine Geste als einer sinnvollen Anwendung. Generell lassen sich diese Clients aber durch reale Softwaresysteme ersetzen, die mit Gesten gesteuert werden sollen. Abbildung 6.30 zeigt beispielsweise eine Applikation für dynamische Schreibgesten, den virtuellen Taschenrechner. Der Benutzer kann alle Ziffern und die Symbole der Grundrechenarten mit dem Zeigefinger schreiben. Wird ein Symbol erkannt, so wird es im Display angezeigt und dessen Taste farblich hervorgehoben, um dem Benutzer eine unmittelbare Rückkopplung zu geben.

Abbildung 6.30: Virtueller Taschenrechner als Applikation für dynamische Schreibegesten

7

Diskussion und Ausblick

Gegenstand dieser Arbeit ist ein allgemeiner Ansatz zur intentionsbasierten Interpretation von Benutzeraktionen, der anhand konkreter Systeme für verschiedene Aufgabenstellungen der Mensch-Maschine-Kommunikation umgesetzt und evaluiert wurde. Die vier resultierenden Systeme betrachten dabei die Benutzerintention aus unterschiedlichen Blickwinkeln, so dass eine große Bandbreite an unterschiedlichen Intentionsbegriffen behandelt werden konnte. Eine ausführliche Diskussion der einzelnen Systeme wurde bereits in den Kapiteln 3 bis 6 vorweggenommen, so dass an dieser Stelle nur auf Aspekte, die für den allgemeinen Ansatz zur intentionsbasierten Interpretation von Benutzeraktionen relevant sind, eingegangen wird. Darüber hinaus wird ein konkreter Ausblick darauf gegeben, wie der vorgestellte Ansatz für eine multimodale Eingabeschnittstelle weiterentwickelt werden könnte.

Die hohen Klassifikationsraten der entwickelten Systeme belegen, dass eine intentionsbasierte Analyse von Aktionen eine äußerst robuste Klassifikation der Benutzerabsicht ermöglicht. Vor allem durch das Vermeiden einer exakten Rekonstruktion der Benutzereingabe wird ein benutzeradäquater Mensch-Maschine-Dialog möglich, da selbst fehlerhaftes Agieren zu einer korrekten Bestimmung des Benutzerziels führen kann. Die Erkennungsraten des sprachverstehenden Systems *Insense* (Kapitel 3), die sowohl für Kommandosprache als auch für natürliche Spontansprache auf einem sehr hohen Niveau liegen, dokumentieren dies. Der Planerkenner *AMPlan* (Kapitel 4) geht noch einen Schritt weiter und nutzt suboptimale Aktionen konsequent für eine robustere Klassifikation des übergeordneten Ziels. Beide Systeme folgen der ersten von zwei entwickelten Ausprägungen des intentionsbasierten Ansatzes und nutzen konsequent die Möglichkeit der Merkmalsselektion. Für die Klassifikation werden somit nur die Worte bzw. Kommandos herangezogen, die für bestimmte Intentionshypothesen charakteristisch sind.

Die zweite Ausprägung des intentionsbasierten Ansatzes richtet sich an Intentionen, die sich durch eine einzige Aktion vermitteln lassen. Anders als bei *Insense* oder *AMPlan* gibt es in diesen Fällen keine aufeinanderfolgenden Worte bzw. Einzelaktionen, zwischen denen sich syntaktisch-semantische Bezüge herstellen lassen. Um dennoch diese Benutzerabsicht verstehen zu können, lernt das Intentionsmodell von *Adaptive Compass* Hintergrundwissen über den Benutzer und seine Präferenzen in Form von semantischen Beziehungen zwischen Aktionsmerkmalen und vorherigen

Intentionen. Darüber hinaus erwies sich die Struktur des Ansatzes als flexibel genug, um den Gestikerkenner *byHand* umzusetzen. Dieses System orientiert sich an der Idee von *Adaptive Compass*, einfache Intentionsmodelle mit aussagekräftigen Merkmalen zu kombinieren um eine Adaption des Bayes'schen Netzes in Echtzeit zu ermöglichen.

Dass der Ansatz unterschiedliche Interpretationen des Intentionsbegriffs erlaubt, ist auf die Realisierung der Intentionsmodelle durch Bayes'sche Netze zurückzuführen. Die Intentionsmodelle ermöglichen für jede Aufgabenstellung eine adäquate Repräsentation von Syntax und Semantik von Merkmalsbeobachtungen. Somit modellieren sie das gesamte Wissen über mögliche Interaktionsinhalte bzw. Intentionen, wodurch das *Verstehen* einer Aktion erst realisierbar wird. Auf Grund der vielfältigen Möglichkeiten bei dem Entwurf einer Netztopologie können gewünschte Eigenschaften eines Intentionsmodells gezielt betont werden. Wie gezeigt, kann das Spektrum von einer komplexen Verknüpfung von Merkmalen zur Modellierung von Syntax und Semantik bis hin zu einer sehr einfachen Netzstruktur für die Echtzeitadaption reichen.

Auf Grund des statistischen Charakters des Ansatzes ist die Interpretation von Erkennungsergebnissen sehr einfach. *Adaptive Compass* belegt dies, da innerhalb eines Klassifikationsprozesses die Ergebnisse von Teilinterpretationen mehrmals interpretiert werden. Diese Eigenschaft ist vor allem für Komponenten zur Dialogmodellierung interessant, die quantitative Aussagen über die Ziele eines Benutzers für eine adaptive Benutzerführung auswerten. Zudem kann durch Veränderung der a-priori-Wahrscheinlichkeit des Intentionsknotens eines Intentionsmodells der aktuelle Dialogkontext auf sehr einfache Weise in die Klassifikation von Aktionen mit einbezogen werden.

Nicht zuletzt soll betont werden, dass es sich den Forschungsgebieten Sprachverstehen, Planerkennung, Benutzermodellierung und Gestikerkennung um sehr anspruchsvolle Aufgabenstellung der Mustererkennung, die in der Regel ganze Forschergruppen beschäftigen. Die Tatsache, dass im Rahmen dieser Arbeit leistungsfähige Systeme aus diesen vier Bereichen entwickelt wurden, zeigt, dass der intentionsbasierte Ansatz eine pragmatische Vorgehensweise bei der Entwicklung neuer Verfahren zur Klassifikation des Benutzerverhaltens ermöglicht.

Bedienkonzepte der Zukunft werden ein großes Spektrum ein Eingabemodalitäten, wie z.B. Sprache, Gestik, Mimik, Tastatur oder Maus, bereitstellen, um eine möglichst intuitive und natürliche Interaktion mit einem Rechner zu ermöglichen. Multimodale Systeme bergen aber auch ein großes Risiko einer falschen Klassifikation der Benutzerabsicht, da für eine Aktion unter Umständen mehrere Modalitäten parallel genutzt werden. Der Benutzer kann beispielsweise eine spontansprachliche und eine gestenbasierte Eingabe zeitgleich oder zeitlich versetzt tätigen. Die einzelnen Komponenten einer Aktion könnten zudem redundant oder widersprüchlich sein. Die Interpretation derartiger Eingaben ist ein ideales Einsatzgebiet für den hier vorgestellten intentionsbasierten Ansatz. Eine Umsetzung des Verfahrens wäre in der Lage, den Benutzer über alle Modalitäten hinweg zu *verstehen*. Die Intentionsmodelle könnten syntaktisch-semantisch verwandte oder redundante Beobachtungen nutzen, um Aussagen über Intentionshypothesen zu konsolidieren. Analog zu *Insense*, kann bei widersprüchlichen Beobachtungen der plausibelsten Aktion der Vorzug gegeben werden. Allerdings sind in einem derartigen System erheblich komplexere Intentionsmodelle zu erwarten, deren Beherrschung eine Weiterentwicklung der intentionsbasierten Interpretation und der Lernverfahren voraussetzen würde, um Echtzeitfähigkeit zu gewährleisten.

A

Anhang

A.1 Glossar

Adaptive Compass
Adaptive Compass ist die Umsetzung des intentionsbasierten Ansatzes für die Interpretation von Aktionen und Situationen im Rahmen eines Fahrzeugnavigationssystems. Indem das System Fahrerpräferenzen und Situationen in Bezug setzt, ist es in der Lage, Zielorteingaben zu verstehen und beispielsweise mehrdeutige Zielorteingaben dem tatsächlichen Wunschziel des Fahrers zuzuordnen.

AMPlan
AMPlan ist ein System zur Planerkennung, basierend auf dem intentionsbasierten Ansatz. Dieses Verfahren hat die Fähigkeit übergeordnete Benutzerintentionen zu ermitteln indem neben dem optimalen Benutzerverhalten auch suboptimale Aktionen berücksichtigt werden.

Basisphrase
Eine Basisphrase des sprachverstehenden Systems *Insense* stellt die unterste Ebene einer Operator- oder Parameter-Phrase dar.

Bayes'sche Netze
Ein probabilistischer Formalimus zur Repräsentation unsicherer und unvollständiger Information. Ein Bayes'sches Netz besteht aus Knoten (Zustandsvariablen), die durch Kanten in einen semantischen Zusammenhang gestellt werden können. Die Knoten repräsentieren Ereignisse der zu lösenden Aufgabestellung, die Kanten entsprechen statistischen Abhängigkeiten, die durch bedingte Wahrscheinlichkeiten quantifiziert werden können. Im Rahmen dieser Arbeit bilden sie die Grundlage für die Realisierung der Intentionsmodelle.

Benutzermodellierung
Dieser Begriff steht für das Erfassen bestimmter Benutzereigenschaften und -interessen. Dieses Wissen kann schließlich zur benutzeradäquaten Dialogführung herangezogen werden oder, wie in Adaptive Compass dazu dienen den Benutzer zu verstehen.

Disambiguierung
Dieser Begriff beschreibt die Fähigkeit von *Adaptive Compass*, im Falle einer mehrdeutigen Zielorteingabe den tatsächlichen Wunschzielort des Fahrers zu ermitteln.

Default-Zielort
Der Default-Zielort ist der Ortsname, der bei Aktivierung eines Fahrzeugnavigationssystems automatisch als Zielort voreingestellt ist.

byHand
byHand ist die Umsetzung des intentionsbasierten Ansatzes zur Interpretation dynamischer Handgesten. Charakteristische Merkmale und ein einfach strukturiertes Intentionsmodell ermöglichen Echtzeittraining und –adaption an den Benutzer und an neue Gesten.

Evaluierungswert
Jede Intentionshypothese wird quantitativ anhand eines Evaluierungswertes bewertet. Die Intention mit dem maximalen Wert ist das Ergebnis der Interpretation.

Gestikerkennung
Dieser Begriff umschreibt die videobasierte Klassifikation dynamischer Handgesten.

Insense
Diese Umsetzung des intentionsbasierten Ansatzes ist in der Lage, sowohl kommandosprachliche als auch spontansprachliche Äußerungen zu interpretieren und den Inhalt rechnergerecht zu erfassen. Selbst bei nicht idealen Bedingungen und grammatikalisch inkorrekten Äußerungen kann die Benutzerabsicht robust erfasst werden.

Instanziierung
Wird einer Zustandsvariable ein konkreter Zustand zugewiesen, so wird von Instanziierung gesprochen. Eine instanziierte Zustandsvariable modelliert sichere Information.

Intention
Im Rahmen dieser Arbeit steht der Begriff der Intention für die Benutzerabsicht, die die eigentliche Motivation für einen Benutzer darstellt, sich mit einer Applikation auseinander zu setzen. Da Intentionen alle potenziellen Interaktionsinhalte abdecken, kann Wissen über die möglichen Benutzerziele für das Verstehen von Aktionen herangezogen werden.

Intentionsbibliothek
Dieser Begriff umschreibt den Hypothesenraum des intentionsbasierten Ansatzes. Jedes zu berücksichtigende potenzielle Benutzerziel ist durch eine Intentionshypothese vermerkt.

Intentionsdecodierung
Dieser Begriff umschreibt die Analyse einer rekonstruierten Aktion mit dem Ziel, deren Inhalt bzw. Aussage zu erfassen.

Intentionshypothese
Jedes potenzielle Benutzerziel wird als Intentionshypothese in der Intentionsbibliothek berücksichtigt.

Intentionsmodelle
Intentionsmodelle dienen der Darstellung syntaktisch-semantischer Beziehungen zwischen Merkmalen und Intentionshypothesen. Sie repräsentieren das gesamte Wissen über potenzielle Interaktionsinhalte und werden anhand Bayes'scher Netze realisiert. Im intentionsbasierten Ansatze dienen sie als Klassifikator.

Kenngröße
Für die Zielort-Prädiktion werden alle Kreise Deutschlands anhand von 28 Kenngrößen charakterisiert. Diese umfassen wirtschaftliche, industrielle Parameter und Größen, die den Freizeitwert einer Gegend dokumentieren.

Merkmalsselektion
Merkmalsselektion ist die Fähigkeit, nur die Merkmale für eine intentionsbasierte Klassifikation heranzuziehen, die für eine Intentionshypothese charakteristisch ist. Das fehlerhafte Komponenten des Merkmalsvektors somit ignoriert werden können, ist eine robuste Erkennung der Benutzerabsicht auf unter nicht idealen Bedingungen möglich.

Merkmalsvektor
Dieser Vektor reduziert eine Beobachtungsfolge auf die charakteristischer Eigenschaften einer Aktion.

Objekt
In *AMPlan* werden alle Verzeichnisse und Dateien als Objekte bezeichnet.

Operator
In *AMPlan* werden Teilpläne bzw. Aktionen zur Manipulation von Objekten als Operator definiert. In *Insense* entspricht ein Operator einer Phrase, die dazu dient, der Applikation mitzuteilen, welche Systemfunktion durch eine sprachliche Äußerung adressiert wird.

Parameter
In *Insense* entspricht ein Parameter einer Phrase, die dazu dient, der Applikation mitzuteilen, inwiefern eine adressierte Systemfunktion beeinflusst werden soll.

Plan
Ein Plan ist eine Abfolge von Aktionen, die dazu dient, das Benutzerziel zu erreichen bzw. umzusetzen.

Planerkennung
Ziel der Planerkennung ist es, auf Basis von beobachteten Aktionen des Benutzers auf dessen Ziel zu schließen. Die Herausforderung besteht darin, möglichst lange vor Erreichen des Ziels Aussagen über die Benutzerabsicht treffen zu können. Es handelt sich somit um Mustererkennung anhand unvollständiger Beobachtungsfolgen.

Situation
In *Adaptive Compass* wird eine Nutzungssituation anhand verschiedener Situationsparameter derart beschrieben, dass sich semantische Beziehungen zwischen einer Situation und der Benutzerintentionen erfassen lassen. Situationsparameter, wie zum Beispiel Zeit oder Tag, könnten auf Ziel-Präferenzen des Fahrers während der Arbeitszeit oder in der Freizeit schließen lassen. Informationen über das Wetter könnten beispielsweise Aussagen über Fahrerinteressen ermöglichen.

Sprachverstehen
Sprachverstehen bezeichnet die Analyse einer sprachlichen Äußerung mit dem Ziel, deren Inhalt bzw. Aussage zu ermitteln. Diese Informationen müssen schließlich so dargestellt werden, dass sie von einem Rechner interpretiert werden können.

Syntaktisch-semantische Ebene
Diese Ebene eines Intentionsmodells realisiert den statistischen Klassifikationsmechanismus, mit dem Merkmale auf Intentionshypothesen abgebildet werden können.

Verstehen
Der Begriff Verstehen bezieht sich in dieser Arbeit auf die Interpretation einer Beobachtungsfolge mit dem Ziel, den Inhalt einer Benutzeraktion zu erfassen. Hierfür ist Wissen über potenzielle Interaktionsinhalte bzw. Intentionen nötig.

Zielort-Prädiktion
Adaptive Compass kann situationsbezogene Aussagen über Benutzerpräferenzen bzgl. möglicher Wunschziele tätigen und dadurch beispielsweise mehrdeutige Zielorteingaben interpretieren oder die sprachgesteuerte Eingabe von Navigationszielen robuster gestalten. Dies wird als Zielort-Prädiktion bezeichnet.

A.2 Symbolverzeichnis

Intentionsbasierte Interpretation von Benutzeraktionen (S. 9 ff.):

$\mathbf{o}$	Beobachtungsfolge
$\mathbf{m}$	Merkmalsvektor
$\mathbf{I}$	Intentionsbibliothek, Vektor aller Intentionshypothesen
IM	Intentionsmodell
E	Evaluierungswert

Intentionsbasierte Interpretation spontansprachlicher Äußerungen (S. 25 ff.):

c	Konfidenzmaß eines Wortes
m_{O}	Zustandsvariable zur Repräsentation optionaler Wörter
m_{S}	Zustandsvariable zur Repräsentation von Schlüsselwörtern

$P_{\mathrm{I}}(\cdot)$ Wahrscheinlichkeit einer Zustandsvariable eines Intentionsnetzes

$P_{\mathrm{Op}}(\cdot)$ Wahrscheinlichkeit einer Zustandsvariable eines Operatornetzes

$P_{\mathrm{P}}(\cdot)$ Wahrscheinlichkeit einer Zustandsvariable eines Parameterphrasennetzes

$P_{\mathrm{PB}}(\cdot)$ Wahrscheinlichkeit einer Zustandsvariable eines Parameter-Basisphrasennetzes

$P^{*}(\cdot)$ durch unsichere Wortbeobachtungen veränderte Wahrscheinlichkeit

$P_{\mathrm{m}}(\cdot)$ durch unsichere, externe Beobachtung veränderte Wahrscheinlichkeit

P_{Operator} maximale Randwahrscheinlichkeit eines Intentionsknotens über alle Operatornetze

$P_{\mathrm{PB}}^{**}(\cdot)$ Wahrscheinlichkeit nach Abbilden einer Wortkette auf die erste Basisphrase

$C_{\mathrm{Parameter}}$ Vollständigkeit einer Parameterphrasen-Beobachtung

n_{P} Anzahl der Parameter einer Parameter-Phrase

n_{iP} Anzahl der beobachteten Parameter einer Phrase

E_{P} Quantitative Bewertung der betrachteten Parameter-Basisphrasen-Kombination

Intentionsbasierte Interpretation unvollständiger Aktionssequenzen (S. 57 ff.):

$\mathbf{m}_{\mathrm{t=0}}$ Merkmale der aktuellen Beobachtung

$\mathbf{m}_{\mathrm{t<0}}$ Merkmale aller vorherigen Beobachtungen

$\mathbf{m}_{\mathrm{t\leq 0}}$ Merkmale aller vorherigen und der aktuellen Beobachtung

$m_{\mathrm{t=0;}k}$ k-tes Merkmal der aktuellen Beobachtung

n_{Objekte} Anzahl der zu manipulierenden Objekte in einem Plan

n_{Aktion} Häufigkeit eines betrachteten Befehls in den Teilplänen eines bestimmten Operators

n_{Operator} Häufigkeit eines bestimmten Operators in den Testplänen

Intentionsbasierte Interpretation von Situationen und Aktionen (S. 85 ff.):

$\mathbf{m}_{\mathbf{Sit}}$ Merkmalsvektor zur Beschreibung einer Situation

$m_{\mathrm{Sit}k}$ k-te Zustandsvariable des Situationsvektors

l_{aktuell} aktueller Ort

k_{aktuell} aktueller Kreis

l_{ein} eingegebener Ort

k_{ein} Kreis des eingegebenen Ortes

$n_{\mathrm{KG}_{\mathrm{Kreis}}}$	Anzahl der Erwerbstätigen in der Branche *KG* in einem bestimmten Kreis
$n_{\mathrm{Gesamt}_{\mathrm{Kreis}}}$	Anzahl der Erwerbstätigen in einem bestimmten Kreis
$\mu_{\mathrm{KG}_{\mathrm{Kreis}}}$	Anteil der Erwerbstätigen in der Branche *KG* in einem bestimmten Kreis
$n_{\mathrm{KG}_{\mathrm{Bund}}}$	Anzahl der Erwerbstätigen in einer Branche *KG* bundesweit
$n_{\mathrm{Gesamt}_{\mathrm{Bund}}}$	Anzahl der Erwerbstätigen bundesweit
$\mu_{\mathrm{KG}_{\mathrm{Bund}}}$	Anteil der Erwerbstätigen in der Branche *KG* bundesweit
$\sigma_{\mathrm{KG}_{\mathrm{Kreis}}}$	Maß dafür, wie charakteristisch eine Branche *KG* für einen Kreis ist
$n_{\sigma>0}$	Anzahl der Kreise mit positiver Streuung
$\mathbf{m}_{\mathbf{KG}}$	Kenngrößenvektor zur Charakterisierung eines Kreises
d_{Ziel}	Entfernung eines aktuellen Ortes (Kreises) zu einem eingegebenen Zielort (Zielkreis)
d	Kreisringe zur Modellierung des Aktionsradius
$\mathbf{s}$	Schwerpunktvektor zur Modellierung der bevorzugten Gegend
$\mathbf{k}_{\mathbf{IT}}$	Einträge der Telefon-History, des Adressbuchs und des Kalenders
$P_n(\cdot)$	Wahrscheinlichkeit, die alle bereits beobachteten *n* Datensätze modelliert
$P_{n+1}(\cdot)$	Wahrscheinlichkeit, die alle vorherigen und den neuen Datensatz modelliert
$P_{n,i}(\cdot)$	Wahrscheinlichkeit, die durch *i* Iterationen verändert wurde
ΔP	Differenz zwischen Ist- und Sollzustand
D_{konv}	Toleranzschwelle für die Konvergenz des iterativen Adaptionsverfahrens
w_{Sit}	Faktor zur iterativen Veränderung bedingter Wahrscheinlichkeiten
w_{Data}	Gewichtungsfaktor zur Steuerung der Adaption an einen neuen Datensatz
$E_{\mathrm{Kreis}}(\mathbf{m}_{\mathbf{Sit}})$	Evaluierungswert eines Kreises in einer Situation $\mathbf{m}_{\mathbf{Sit}}$
$E_{\mathrm{KG}}(\mathbf{m}_{\mathbf{Sit}})$	Evaluierung eines Kreises auf Basis von Fahrerpräferenzen
$E_{\mathrm{Distanz}}(\mathbf{m}_{\mathbf{Sit}})$	Evaluierung eines Kreises auf Basis typischer Aktionsradien
E_{Global}	Evaluierung eines Kreises auf Basis globaler Benutzerpräferenzen
s_{Kreis}	Entfernung eines Kreis zum Schwerpunkt des Benutzerinteresses
$s_{\max}$	maximale Entfernung eines Kreises vom Schwerpunkt
c_{I}	a-priori-Konfidenzmaß einer Intentionshypothese

$c_{\max}$ maximales a-priori-Konfidenzmaß einer Intentionshypothese

$E_{\mathrm{I}}(\mathbf{m}_{\mathbf{Sit}})$ Evaluierungswert für eine Intentionshypothese, der sich durch Zielortprädiktion und a-priori-Gewichtung durch einen Spracherkenner ergibt

$E_{\mathrm{I}_t,\mathrm{def}}$ Evaluierung der *t*-ten Intentionshypothese für die Bestimmung des Default-Zielorts

$\tilde{z}_{\mathrm{def}}$ von *Adaptive Compass* bestimmter Default-Zielort

Intentionsbasierte Interpretation dynamischer Handgesten (S. 123 ff.):

$I(x,y,t)$ Luminanzwert des Pixels mit den Koordinaten (*x*,*y*) zum Zeitpunkt *t*

$D(x,y,t)$ Differenz der Luminanzwerte von Pixel (*x*,*y*,*t*) und (*x*,*y*,*t*-1)

I_{diff} Energie des Differenzbilds $D(x,y,t)$

d_{motion} Schwellwert zur Detektion einer Handbewegung

n_{G} Anzahl der Frames einer dynamischen Handgeste

T_{G} Dauer einer dynamischen Handgeste

$I_{\mathrm{grid}}(r,t)$ Luminanzwerte der Bildpunkte des Rechtecks *r*

$R(r,t)$ diskretisierte Luminanzwerte eines Rechtecks *r*

$m_u(r)$ Merkmal des Rechtecks *r* für das *u*-te Zeitintervall

$\mathbf{m}_{\mathrm{u}}$ Merkmalsvektor des *u*-ten Zeitintervalls

$P_n(\cdot)$ Wahrscheinlichkeit, die alle bisher beobachteten Datensätze modelliert

$P_{n+1}(\cdot)$ Wahrscheinlichkeit, die alle vorherigen und den neuen Datensatz modelliert

Literatur

[Adv98] Advani, A.; Lo K.; Shahar Y.: *Intention-Based Critiquing of Guideline-Oriented Medical Care*. Proceedings of the AMIA ´98 Symposium, 1998.

[Alb97] Albrecht, D.; Zukerman, I.; Nicholson, A.; Bud, A.: *Towards a Bayesian Model for Keyhole Plan Recognition in large Domains*. In Proceedings of the Sixth International Conference on User-Modeling, Sardinien, Italien, Springer-Verlag, 1997, S.365-376.

[All80] Allen, J.F.; Perrault, C.R.: *Analyzing Intention in Utterances*. Artificial Intelligence, 15(3), 1980, S.143-178.

[Ale95] Alexandersson, J.: *Plan Recognition in VERBMOBIL*. In M. Bauer, S. Carberry, D. Litman (eds.), Proceedings of the IJCAI `95-Workshop: The Next Generation of Plan Recognition Systems, Montreal, 1995, S. 2-7.

[Aus98] Aust, H.: *Sprachverstehen und Dialogmodellierung in natürlichsprachlichen Informationssystemen*. Dissertation, Institut für Informatik, RWTH Aachen, 1998.

[Bee00] Beetz, M.; Grosskreutz, H.: *Probabilistic Hybrid Action Models for Predicting Concurrent Percept-Driven Robot Behavior*. Proceedings of the Fifth International Conference on AI Planning Systems, AAAI Press, Menlo Park, Californien, 2000, S. 42-61.

[Bra90] Bratman, M.: *What is Intention ?*. In Intentions in Communication. Philip Cohen, Jerry Morgan & Martha Pollack (Eds.), MIT Press, 1990.

[Bro89] Bronstein, I.N; Semandjajew, K.A.: *Taschenbuch der Mathematik*. Gemeinschaftsausgabe Verlag Nauka, Moskau BSB. Teuber Verlagsgesellschaft 1989.

[Bub96] Bub, T; Schwinn, J.: *VERBMOBIL: The Evolution of a complex large Speech-to-Speech Translation System*. Proceedings of ICSLP `96. Philadelphia, 1996, S. 2371-2374.

[Cha93] Charniak, E., Goldman. R.: *A Bayesian Model of Plan Recognition*. Artificial Intelligence, 1993, S. 53- 79.

[Coh79] Cohen, P.R.; Perrault, C.R.: *Elements of a plan-based Theory of Speech Acts*. Cognitive Science, 3(3), 1979, S. 245- 274.

[Coh81] Cohen, P.R.; Perrault, C.R.; Allen , J.F.: *Beyond Question Answering*. In Lehnert and Ringle: Strategies für Natural Language Processing. Hillsdale , New York, 1981, S. 245- 274.

[Coh87] Cohen, P.R.; Levesque, H.: *Intention – Choice – Commitment*. Proceedings of AAAI ’87, 1987.

[Coh90a] Cohen, P.R.; Levesque, H.: *Persistance, Intention and Commitment*. In Intentions in Communication. Philip Cohen, Jerry Morgan & Martha Pollack (Eds.), MIT Press, 1990.

[Coh90b] Cohen, P.R.; Levesque, H.: *Rational Interaction as the Basis for Communication*. In Intentions in Communication. Philip Cohen, Jerry Morgan & Martha Pollack (Eds.), MIT Press, 1990.

[Con97] Conati, C.; Gertner, A.; VanLehn, K.; Druzdzel, M.: *Online Student Modeling for coached Problem Solving using Bayesian Networks.* Proc. of the Sixth International Conference on User Modeling, Sardinien, Italien, 1997, S. 231-242.

[Cor93] Correa, M.; Coelho, H.: *Around the Architectural Agent Approach to Model Conversations*. Proceedings of the Fifth European Workshop on Modelling Autonomous Agents in Multi-Agent-Worlds, Neuchatel (Schweiz), 1993.

[Dag92] Dagum, P.; Galper, A.; Horvitz, E.: *Dynamic Network Models for Forecasting.* Proc. of the Eighth Workshop on Uncertainty in Artificial Intelligence. Stanford, USA, 1992, S. 41-48.

[Fre00] Freedman, R.: *Plan-Based Dialogue Management in a Physics Tutor*. Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000), Seattle, 2000.

[Gei02] Geiger, M.; Nieschulz, R.; Zobl, M.; Lang, M.: *Bedienkonzept zur Gestenbasierten Interaktion mit Geräten im Automobil - Gesture-Based Control Concept for In-Car Devices.* Tagungsband VDI/VDE - GMA Fachtagung USEWARE 2002, Darmstadt, 11.-12.06.2002. Düsseldorf: VDI-Verlag, 2002, Hrsg.: VDI. VDI-Berichte; 1678 "USEWARE 2002 Mensch-Maschine-Kommunikation/Design", S. 299-303

[Gof01] Gofuku, A.; Tanaka Y.: *Intention-Based Display of Diagnostic Information*. Proceedings of the International Conference on Information Systems, Analysis and Synthesis `2001, Volume 8, 2001.

[Hof00] Hofmann, M.; Lang, M.: *Belief Networks for a Syntactic and Semantic Analysis of Spoken Utterances*. Proc ICSLP 2000, Peking, China, 2000, China Military Friendship Publish, Vol. 2, S.875-878.

[Hof01a] Hofmann, M.; Lang, M.: *Intention-based Probabilistic Phrase Spotting for Speech Understanding.* Proc of the Int. Symp. On Intelligent Multimedia, Video and Speech Processing, ISIMP 2001, Hong Kong, China, 2001. Ed.: IEEE Hong Kong Chapter of Signal Processing, S. 99-102.

[Hof01b] Hofmann, M.; Lang, M.: *User Appropriate Plan Recognition for Adaptive Interfaces.* Proc. of the 9th Int. Conf. on Human-Computer Interaction (HCI International 2001), New Orleans, Louisiana, USA, 2001. Ed.: Lawrence Erlbaum Ass., New Jersey, 2001. Vol. 1 "Usability Evaluation and Interface Design", S. 1130-1134.

[Hof01c] Hofmann, M.; Lang, M.: *A Dialog Model for Offering Task Completion for Complex Domains.* Proc. of the 9th Int. Conf. on Human-Computer Interaction (HCI International 2001), New Orleans, Louisiana, USA, 2001. Ed.: Lawrence Erlbaum Ass., New Jersey, 2001. Poster Sessions: Abridged Proceedings, S. 305-307.

[Hof01d] Hofmann, M.; Bengler, K.; Lang, M.: *Ein Assistenzsystem zur fahrer- und situationsadaptiven Prädiktion potentieller Zielorte für eine robuste Interaktion mit sprachgesteuerten Navigationssystemen.* Tagungsband "Elektronik im Fahrzeug", Baden-Baden, 2001. VDI-Bericht 1646, S. 979-996.

[Hof02] Hofmann, M.: *Erkennung dynamischer Handgesten zur berührungslosen Interaktion mit informationstechnischen Systemen*. Patentanmeldung bei Deutschen Patentamt, Nr. 10233233.9-53.

[Hor97] Horwitz, E.: *Agents with Beliefs: Reflections on Bayesian methods for User Modeling.* Proc. of the Sixth International Conference on User Modeling, Sardinien, Italien, 1997, S. 441-442.

[Hor98] Horwitz, E.; Breese, J.; Heckermann, D.; Hovel, D.; Rommelse, K.: *The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users*. Proceedings of the Fourthteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, Juli 1998, Morgan Kaufman Publishers, S. 256-266.

[Hub94] Huber, M. J.; Durfee, E.H.: *Observational Uncertainty in Plan Recognition among interacting Robots*. In Working Notes of the Workshop on Dynamically Interasting Robots, Chambery, Frankreich, 1993, S. 68-75.

[Hub94] Huber, M. J.; Durfee, E.D; Wellman, M.P.: *The automated Mapping of Plans for Plan Recognition*. In Proceedings of the Tenth Conference on Uncertainty on Artificial Intelligence, 1994, S. 344-351.

[Hun03] Hunsinger, J.: *Multimodale Erfassung mathematischer Formeln durch einstufig-probabilistische semantische Decodierung*. Dissertation, Fakultät für Elektro- und Informationstechnik, Technische Universität München, 2003.

[Jam96] Jameson, A.: *Numerical Uncertainty Management in User and Student Modeling.: An Overview of Systems and Issues.* User Modeling and User-Adapted Interaction, 1996, Ausgabe 5, S.193-251.

[Jen96] Jenson, F.: *An Introduction to Bayesian Networks*. UCL (University London College). 1996.

[Jin99] Jin Qian, J.: *Implementierung eines Clustering-Verfahrens zur Inferenz in Bayes'schen Netzen*. Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1999.

[Kit96] Kitano, H.; Tambe, M; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara H.; Node, I; Asada, M.: *The Robocup synthetic Agent Challenge*, 1997.

[Kon93] Konolige, K.; Pollack, M.: *A Representationalist Theory of Intention*. Proceedings of IJCAI ´93. 1993.

[Lam91] Lambert, L.: *Modifying Beliefs in a Plan-Based Dialogue Model*. Proc. of the 29th Annual Meeting of the Association for Computational Linguistics, 1991.

[Lan02] Lang, M.: 1. Sinnesorgane und Sinnesmodalitäten. 2. Interaktionsmodelle und Dialogformen. 28 S. In: Handbuch der Ergonomie, Bd.4, C-8.1.: Interaktion zwischen Mensch und Computer. Hrsg.: Bundesamt für Wehrtechnik und Beschaffung, Koblenz. Carl Hanser Verlag, München, 2002, 6. Ergänzungslieferung.

[L&H98] Lernout & Hauspie Speech Products N.V.: *Lernout & Hauspie-Software Developers' Kit*. Ieper, Lernout & Hauspie Speech Products E.V.,1998.

[Les99] Lesh, N.; Allen, J.: *Simulation-based Inference for Plan Monitoring*. In Proc. of the National Conference on Artificial Intelligence, 1999, S. 280-285.

[Lev95] Levin, E.; Pieraccini, R.: *Concept-Based Spontaneous Speech Understanding System*. In Proceedings of Eurospeech`95, 1994, Vol.2, S. 555-558.

[Mat02] Matsubara, S.; Kimura, S.; Kawaguchi, N; Yamaguchi, Y.; Inagaki, Y.: *Example-based Speech Intention Understanding and Its Application in In-Car Spoken Dialogue System*. Proceedings of the 17th International Conference on Computational Linguistics (COLING-2002), Taipei, 2002. Vol 2, S. 633-639.

[Mil94] Miller, S.; Bobrow, R.; Ingria, R.; Schwartz, R.: *Hidden Understanding Models of Natural Language*. In Proceedings of the Association of Computational Linguistics, 1994, S. 555-558.

[Mik97] Miksch, S.; Shahar Y.; Johnson P.: *A Task-Specific, Intention-Based, and Time-Oriented Language for Representing Skeletal Plans*. In Motta, E.; Harmelen, F. v.; Pierret-Golbreich, C.; Filby, I.; Wijngaards, N. (eds.), 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97), Milton Keynes, UK, 1997.

[Mor00] Morguet, P.: *Stochastische Modellierung von Bildsequenzen zur Segmentierung und Erkennung dynamischer Gesten*. Dissertation, Fakultät für Elektro- und Informationstechnik, Technische Universität München, 2000.

[Mue97] Müller, J.: Die semantische Gliederung als Repräsentation des Bedeutungsinhalts innerhalb sprachverstehender Systeme. Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, 1997

[Neu00] Neuss, R.; Hunsinger, J.; Stenzel, R.; Lang, M.: *Sprachgesteuerte Fahrerassistenz durch einstufig-probabilistisches Verstehen natürlich gesprochener Sprache - Voice Controlles Driver Assistance by Single-Stage Probabilistic Natural Language Understanding.* Tagungsband VDI/VDE - GMA Fachtagung USEWARE 2002, Darmstadt, Düsseldorf: VDI-Verlag, 2002, Hrsg.: VDI. VDI-Berichte; 1678 "USEWARE 2002 Mensch-Maschine-Kommunikation/Design", S. 55-60.

[Neu01] Neuss, R.: *Usability Engineering als Ansatz zum multimodalen Mensch-Maschine Dialog*. Dissertation, Fakultät für Elektro- und Informationstechnik, Technische Universität München, 2001.

[Nie94] Nielsen, J.: *Usability Engineering*. Morgan Kaufmann Publishers, San Francisco, CA, USA, Oktober 1994.

[Nie01] Nieschulz, R.; Geiger, M.; Bengler, K.; Lang, M.: *An Automatic, Adaptive Help System to Support Gestural Operation of an Automotive MMI.* Proc. of the 9th Int. Conf. on Human-Computer Interaction (HCI International 2001), New Orleans, Louisiana, USA, 2001. Ed.: Lawrence Erlbaum Ass., New Jersey, 2001. Vol. 1 "Usability Evaluation and Interface Design", S. 272-276.

[Nie02] Nieschulz, R.; Geiger, M.; Zobl, M; Lang, M.: *Informationsbedarf bei gestischer Interaktion im Fahrzeug - Need for Assistance in Automotive Gestural Interaction.* Tagungsband VDI/VDE - GMA Fachtagung USEWARE 2002, Düsseldorf: VDI-Verlag, 2002, Hrsg.: VDI. VDI-Berichte; 1678 "USEWARE 2002 Mensch-Maschine-Kommunikation/Design", S. 293-297.

[Nil99] Nillson, N.: *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers, Inc. San Francisco, California, 1998.

[Oka02] Oka, K.; Sato, Y.; Koike, H.: *Real-time Tracking of Multiple Fingertips and Gesture Recognition für Augmented Desk Interface Systems*. Proc. of the IEEE Int'l Conf. Automatic Face and Gesture Recognition (FG 2002), 2002, S. 429-434.

[Pea88] Pearl, J.: *Probabilistic Reasoning in Expert Systems*. Morgan Kaufmann Publishers, San Mateo, CA, USA, 1988.

[Per80] Perrault, C.R.; Allen, J.F.: *A plan-based Analysis of indirect Speech Acts*. American Journal of Computational Liguistics, 6(3-4), 1980, S. 167-182.

[Pyn95] Pynadath, D.V.; Wellmann M.P.: *Accounting for Context in Plan recognition, with application to traffic monitoring*. In Proc. of the Conference on Uncertainty in Artificial Intelligence, 1995, S. 472-481.

[Pyn99] Pynadath, D.V.: *Probabilistic grammars for plan recognition*. PhD thesis, University of Michigan. 1999.

[Rao93] Rao, A; Georgeff M.: *Intentions and Rational Commitment*, Tech. Rep. 08, Australian Artificial Intelligence Institute, Melbourne, Australien, 1990.

[Rig97] Rigoll, G.; Kosmala, A.; Eickeler, S.: *High Performace Real-time Gesture Recognition using Hidden Markov Models*. In Gesture Workshop, Bielefeld, 1997.

[Rus95] Russel, S.; Norvig, P.: *Artificial Intelligence – A Modern Approach*. Prentice Hall, 1995.

[Sch00] Schiller, R.: *Entwicklung und Evaluierung eines adaptiven Benutzerassistenzsystems*. Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 2000.

[Sch01] Schmidt, T.; Buck, S.; Beetz, M.: *AGILO RoboCuppers 2002: Utility- and Plan-based Action Selection based on Probabilistically Estimated Game Situations*. Proceedings of RoboCup'2001. 2001, S. 611-615.

[Sen92] Seneff, S.: *TINA: A Natural Language System for Spoken Language Applications*. Computational Liguistics, Vol. 18, No. 1, 1992, S. 61-86.

[Sta95] Starner, T.; Pentland, A.: *Visual Recognition of American Sign Language using Hodden Markov Model*. In Proc. of the International Workshop on Automatic Face- and Gesture-Recognition, Killington, VT, Oktober 1996.

[Sta97] Stahl, H.: *Konsistente Integration stochastischer Wissensquellen zur semantischen Decodierung gesprochener Äußerungen*. Dissertation, Fakultät für Elektro- und Informationstechnik, Technische Universität München, 1997.

[Str92] Stroustrup, B.: *Die C++ Programmiersprache*. Addison-Wesley, Bonn, München, Paris [u.a.], 1992.

[Uck99] Ucke, A.: *Entwicklung eines Algorithmus zur automatischen Generierung einer Planbibliothek*. Diplomarbeit, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, 1999.

[Wah00] Wahlster, W.: *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer-Verlag, Berlin/Heidelberg, 2000.

[Wah01] Wahlster, W.; Reithinger, N.; Blocher, A.: *SMARTKOM: Multimodal Communication with a life-like Character*. Proceedings of Eurospeech 2001, Aalborg, Dänemark, 2001.

[Wan02] Wang, Y.; Acero, A.; Chelba, C; Frey, B.; Wong, L.: *Combination of Statistical and Rule-Based Approaches for spoken Language Understanding*. In Proceedings of the International Conference on Spoken Language Processing (ICSLP'02), Denver, Colorado, 2002.

[Wel99] Welch, B. B.: *Practical Programming in Tcl and Tk*. Prentice Hall, New Jersey, USA, 1999.

[Yam92] Yamato, J.; Phya, J.; Ishii, K.: *Recognizing Human Action in Time-Sequential Images using Hidden Markov Model*. Proc of IEEE Computer Vision and Pattern Recognition ´92, S. 379-385, 1992.

[Zho00] Zhou, Y.; Freedman, R.; Glass, M.; Michael, J.A.; Rovick, A.A.; Evens, A.W.: *Delivering Hints in a Dialogue-Based Intelligent Tutoring System*. Proceedings of the Sixteenth National Conference of Artificial Intelligence (AAAI ´99), Orlando, 1999.

[Zob02] Zobl, M.; Geiger, M.; Morguet P.; Nieschulz, R.; Lang, M.: *Gestenbasierte Interaktion mit Geräten im Automobil - Gesture-Based Control of In-Car Devices.* Tagungsband VDI/VDE - GMA Fachtagung USEWARE 2002, Düsseldorf: VDI-Verlag, 2002, Hrsg.: VDI. VDI-Berichte; 1678 "USEWARE 2002 Mensch-Maschine-Kommunikation/Design", S. 305-309.