

Klassifizierung und Weiterentwicklung adaptiver Methoden

Istvan-Tibor Nebel, Ralf Paschke

Medizinische Klinik und Poliklinik III des Universitätsklinikums Leipzig, AG Medizinische Lern- und Informationssysteme, Ph.-Rosenthal-Str. 27, D-04103 Leipzig, Email: nebt@medizin.uni-leipzig.de

Die Anforderungen an adaptive tutorielle Systeme können mittels unterschiedlicher Methoden realisiert werden. Der Beitrag versucht die am häufigsten implementierten Algorithmen adaptiver Methoden nach ihrer Art Informationen zu verarbeiten, zu klassifizieren. Jede adaptive Methode besitzt verschiedene Vor- und Nachteile hinsichtlich der Verarbeitungsgeschwindigkeit und der Inferenzfähigkeit. Ausgehend von den spezifischen Eigenschaften sollen in einer hybriden Modellierung die Vorteile der einzelnen Methoden in einer Benutzermodellierungskomponente – den Controltypes – zur Anwendung gebracht werden.

1 Methoden und Algorithmen der Benutzermodellierung

Die Anforderungen an adaptive tutorielle Systeme können mittels unterschiedlicher Methoden realisiert werden. Im Folgenden sind die am häufigsten implementierten Algorithmen adaptiver Methoden aufgeführt, und nach ihrer Art Informationen zu verarbeiten, klassifiziert.

Eine Möglichkeit verschiedene Algorithmen zu klassifizieren ist es, sie unterschiedlichen Repräsentationsformen zuzuordnen. Diese Art der Klassifizierung ist in Tabelle 1 dargestellt. Dabei ist festzustellen, dass, neben der Auswertung von eindeutigen Benutzerdaten, Verfahren existieren, die Annahmen mittels verschiedener Methoden treffen. Diese Methoden können auf explizit aufgestellten Regelwerken, logikbasierten Methoden oder auf mathematischen Modellen basieren. Letztere verwenden wiederum Methoden aus der Probabilistik und Statistik, um Annahmen unter Unsicherheit zu treffen.

ad hoc	regelbasiert		mathematisch			
	logisch	symbolisch	heuristisch	logisch	stochastisch	probabilistisch
Primäre Erwerbsheuristik	Overly-Modelle	Symbolische KI	Bewerkungskriterien	Stereotypisierung	Fuzzy-Logik	Bayes'sche Netze
Präferenzen als Ziele	Kollaboratives Filtern		Evolvierende Agenten	Doppelte Stereotypisierung	Neuronale Netze	Dempster-ShaferTheorie
			Genetische Algorithmen	Kandidat-Kritik-Modell		
Hybridsysteme						

Tabelle 1: Klassifizierung nach Repräsentationsformen

Diese Klassifikation kann auch auf anderen Ebenen getroffen werden. So werden die Methoden der Künstlichen Intelligenz (KI) nach Strecker [29] vorrangig zwischen dem *symbolischen* und *subsymbolischen* Paradigma unterschieden. Während im symbolischen Paradigma die Objekte der realen Welt mit Hilfe von Symbolen und Abhängigkeiten zwischen den Objekten mittels formaler Regeln dargestellt werden, nimmt das subsymbolische Paradigma konnektionistische Modelle, um Probleme durch gewichtete Knotenverbindungen lösen zu lassen. Nach Blutner [4] ist das Wissen in konnektionistischen Systemen in Form von Verbindungsmatrizen oder äquivalenter Übergangsfunktionen kodiert. Ein symbolisches System bedient sich üblicherweise (Default-) logisch basierter Datenbasen zur Wissensrepräsentation.

In der Arbeit von Koller [15] wird ein Klassifikationsmodell intelligenter Problemlösemethoden vorgestellt, das ein Dreieck zwischen symbolischer, subsymbolischer und numerischer Verarbeitung aufspannt. Ebenso können die in diesem Artikel dargestellten Methoden aus dem Blickwinkel der individuellen und der stereotypischen Benutzermodellierung gesehen werden.

Die einzelnen adaptiven Methoden können weiterhin dem überwachten (supervised learning), dem unüberwachten (unsupervised learning) und dem bestärkenden (reinforcement learning) Lernverfahren zugeordnet werden. Während beim *unsupervised learning* selbstständig Regelmäßigkeiten gefunden und in Lernregeln transformiert werden, bedient sich das *supervised learning* exakt definierten Regelsystemen um Inferenzen zu bilden. Beim *reinforcement learning* [21] wird dem System mitgeteilt, ob die Inferenz korrekt oder inkorrekt war. Der Unterschied zum überwachten Lernen besteht darin, dass der exakte Differenz zwischen richtig und falsch nicht mitgeteilt wird. Es wird lediglich die Wahrscheinlichkeit erhöht, dass die gezogene Inferenz in die *richtige Richtung* weist.

Die einzelnen adaptiven Methoden¹ lassen sich überblicksmäßig besser in einem semantischen Klassifizierungsschema darstellen. Diese semantische Assoziation der Methoden zu Klassifikationsmerkmalen wird durch die dargestellte Matrix in Tabelle 2 verdeutlicht. In den senkrechten Spalten sind dabei die adaptiven Methoden und in den waagerechten Spalten das Klassifikationsmerkmal dargestellt. Die markierten Kreuzungspunkte verdeutlichen die semantische Zugehörigkeit. Die klassifizierten adaptiven Methoden werden in den nächsten Abschnitten erläutert.

Eine zweite Klassifikationsmöglichkeit besteht darin, die Merkmale der Algorithmen zu untersuchen und bezüglich der Nützlichkeit zu bewerten. Merkmale adaptiver Methoden sind:

- die *Lernfähigkeit* des Algorithmus, d.h. ob die Möglichkeit von Generalisierungen bestehen,
- die *Erklärbarkeit* der gezogenen Inferenzen,
- die *Verarbeitungsgeschwindigkeit* des Algorithmus, d.h. wie schnell liefern die Daten eine Inferenz

¹Die Abkürzungen sind in Tabelle 2 aufgelöst.

	EH	OV	ST	KM	PZ	BK	GA	EA	SK	FL	BN	DS	NN	SA
konnektionistisch											•	•	•	•
stochastisch			•				•						•	
regelbasiert								•						
heuristisch	•		•					•						
probabilistisch											•	•		
unsupervised							•		•	•	•	•	•	•
supervised	•	•	•											
reinforcement				•	•	•		•						
symbolisch	•		•					•			•	•		
subsymbolisch										•			•	
individuelle BM	•						•			•	•	•	•	
stereotypische BM			•					•						

Tabelle 2: Semantisches Klassifizierungsschema adaptiver Methoden

- die *Verarbeitungsmethode* des Algorithmus, d.h. ob sequentiell oder parallel Daten verarbeitet werden
- die *Inferenzmöglichkeit*, d.h. ob sie automatisch erfolgt, die wiederum alle oder nur spezielle Annahmen berücksichtigt werden oder ob manuell Inferenzregeln aufgestellt werden,
- die *Inferenzmöglichkeit unter Unsicherheit*, d.h. werden Wahrscheinlichkeiten berücksichtigt,
- die Form der *Wissensrepräsentation*, d.h. wie Objekte modelliert werden müssen – mit Wahrscheinlichkeiten oder als Fakten,
- der *Speicherplatz*, d.h. sind mehrere Zwischenwerte für die Inferenzbildung notwendig,
- die *Erweiterbarkeit*, d.h. können zusätzliche Wissensitems automatisch einfließen oder müssen die manuell berücksichtigt werden,
- der *Aufwand* der Implementierung des Algorithmus,
- der *Ansatz*, d.h. die verwendeten Parameter zur Ansteuerung der Algorithmen.

In Tabelle 3 sind diese Merkmale anhand implementationstechnisch relevanter Eigenschaften bewertet und den adaptiven Methoden zugeordnet. Dabei wurden zusätzlich die adaptiven Methoden den beschriebenen Paradigmen, den symbolischen und den konnektionistischen Modellen, zugeordnet.

Im Folgenden sind die am häufigsten derzeit verwendeten adaptiven Methoden auf dem Gebiet der Benutzermodellierung aus den Blickwinkeln der Klassifizierungsschemata erläutert.

1.1 Primäre Erwerbsheuristiken

Der Aufbau eines Benutzermodells mittels primärer Erwerbsheuristiken besteht aus der unmittelbaren Interaktion mit dem Benutzer und ist somit den ad hoc-Methoden zuzuordnen. In der Arbeit von Kobsa [14] werden aus Arbeiten anderer Autoren folgende domänenunabhängige Erwerbsheuristiken vorgestellt:

- Aus einer *korrekten Verwendung* von Objekten schließt man das Wissen darum. Demzufolge wird aus einer *inkorrekten Verwendung*, geschlossen, dass das Objekt unbekannt ist.
- Aus einer *Erklärungsanforderung* wird ebenfalls das Nichtwissen um ein Objekt geschlossen – aus einer *Detaillierungsanforderung* die Kenntnis des Wissensitems.
- Ein positives/negatives *Feedback* des Benutzers zu einer getroffenen Systemannahme verstärkt/verringert die Plausibilität dieser Annahme.

Aus den explizit gewonnenen Daten sind die Ableitungen sehr gut erklärbar und die Verarbeitungsgeschwindigkeit ist sehr hoch. Jedoch bietet diese Methode keine Berücksichtigung von Wahrscheinlichkeiten und damit keine Inferenzmöglichkeit unter Unsicherheit.

1.2 Overlay-Modelle

Hahn [7] definiert in Overlay-Modellen das Benutzerwissen als Teil des Expertenwissens. Jedes Wissenselement des Expertenwissens erhält eine Kennzeichnung, ob es vom Benutzer bereits beherrscht wird oder nicht. Diese sehr grobe Unterscheidung wird durch erweiterte Overlay-Modelle, die den Grad des Könnens speichern (Ausprägung zwischen den Extremwerten *beherrscht gar nicht* und *beherrscht absolut*), teilweise kompensiert. In erweiterten Overlay-Modellen wird also typisches Fehlerverhalten berücksichtigt.

Das Overlay-Modell hat den Nachteil, dass Fehler des Benutzers nicht interpretiert werden können. Es wird lediglich festgestellt, dass eine Wissensseinheit des Expertenwissens nicht vorhanden ist, nicht aber, wie ein falsches Resultat einer Übungsaufgabe zustande gekommen sein mag.

Zusätzliche Fehlerbibliotheken werden in Overlay-Modellen eingesetzt, um alle möglichen bzw. typischen Fehlerquellen eines Benutzers im Vorfeld zu definieren. Das Modell enthält jetzt neben den Fehlern des Benutzers auch die fehlerhaften Vorstellungen eines Wissenselements. Anhand dieser Informationen kann das System die Ursachen für die spezifischen Probleme eines Lernenden lokalisieren und entsprechend darauf reagieren.

Methode	Eigenschaften											
	Supervised/Unsupervised/Reinforcement Lernverfahren											
	Lernfähigkeit											
	Erklärbarkeit											
	Verarbeitungsgeschwindigkeit											
	Verarbeitungsmethode											
	Inferenzbildung											
	Inferenzen unter Unsicherheit											
	Wissensrepräsentation											
	Speicherplatz											
	Erweiterbarkeit											
	Aufwand											
	Ansatz											
Erwerbsheuristik (EH)	S	schwer	sehr gut	hoch	seq.	man.	keine	sehr leicht	niedrig	man.	sehr niedrig	symb.
Overlay-Modelle (OV)	S	schwer	gut	mittel	seq.	man.	keine	sehr leicht	hoch	man.	niedrig	
Stereotypen (ST)	S	mittel	gut	mittel	seq.	auto unvollst.	mittel	leicht	hoch	auto	mittel	
Kandidat Kritik-Modell (KM)	R	schwer	gut	mittel	seq.	auto unvollst.	mittel	leicht	mittel	man.	niedrig	
Präferenzen als Ziele (PZ)	R	schwer	sehr gut	hoch	seq.	man.	keine	sehr leicht	niedrig	man.	niedrig	
Bewertungskriterien (BK)	R	schwer	gut	mittel	seq.	man.	mittel	sehr leicht	niedrig	man.	mittel	
Genetische Algorithmen (GA)	U	leicht	schwer	hoch	par.	auto unvollst.	gut	leicht	hoch	auto	hoch	
Evolvierende Agenten (EA)	U	mittel	gut	mittel	seq.	auto unvollst.	mittel	mittel	mittel	auto	mittel	
Symbolische KI (SK)	U	mittel	mittel	langsam	seq.	auto vollst.	gut	leicht	hoch	man.	hoch	
Fuzzy-Logik (FL)	U	mittel	schwer	langsam	seq.	auto vollst.	gut	mittel	hoch	auto	hoch	
Bayes'sche Netze (BN)	U	leicht	mittel	langsam	par.	auto vollst.	gut	schwer	hoch	auto	sehr hoch	konn.
Dempster-Shafer-Theorie (DS)	U	leicht	sehr schwer	langsam	par.	auto vollst.	gut	schwer	hoch	auto	sehr hoch	
Neuronale Netze (NN)	U	leicht	keine	langsam	par.	auto vollst.	gut	schwer	hoch	auto	sehr hoch	
Spreading Activation (SA)	U	leicht	keine	langsam	par.	auto vollst.	gut	schwer	hoch	auto	sehr hoch	

Tabelle 3: Merkmale von Benutzermodellierungsmethoden

Bei einer inkorrekten Lösung des Benutzers bzw. Lernalers wird versucht, die Abweichung von der richtigen Lösung durch die Anwendung eines solchen typischen Fehlers zu erklären. Damit setzt sich das Benutzermodell aus den vom Lerner korrekt beherrschten Wissensinhalten des Domänenexpertenwissens und zusätzlich den Elementen der Fehlerbibliothek, die der Lerner bei der Lösung von Aufgaben benutzt hat, zusammen.

Unter Speicherplatzaspekten betrachtet, ist das Overlay-Verfahren als ungünstig einzuschätzen, da die Initialisierung des Modells sehr lange dauert, wenn für jedes einzelne Element der Kenntnisstand ermittelt werden muss. Der Umfang der Bibliothek wächst mit allen möglichen Fehlern, die im Voraus antizipiert werden müssen.

Abweichungen, die nicht mit den in der Bibliothek enthaltenen Fehlern erklärt werden können und damit keinen Aufschluss über die Entstehung der inkorrekten Lösung bieten, ist ein weiterer Nachteil bei der Realisierung eines Systems mittels Overlay-Modell. In Fehlerteil-Bibliotheken werden die möglichen Benutzerfehler deshalb dynamisch zur Laufzeit erzeugt, um einen Kompromiss zwischen Rechenzeit und Speicherplatz zu erreichen. Vorteile dieser Technik sind nach Harrer [8] einerseits der geringere Umfang von Bibliotheken von Fehlerteilen und andererseits, dass Fehler bei Bedarf synthetisiert werden können. Ein Problem hierbei ist allerdings, dass selbst bei einer geringen Anzahl von Fehlerteilen sich die Anzahl der möglichen Kombinationen von Fehlerteilen exponentiell entwickeln kann, bevor ein Fehler synthetisiert wird, der die Abweichung der Lernerlösung zu erklären vermag.

1.3 Stereotypen

Stereotypen sind die häufigste anzutreffende Form um tutorielle Systeme adaptiv zu gestalten. Sie wurde von Rich [26] in die Benutzermodellierung eingeführt und später von Chin [5], um eine doppelte Stereotypisierung erweitert. Sie werden verwendet, um korrelierende Eigenschaften von Benutzern zu einer Klasse zusammenzufassen, um die Eigenschaften einer Gruppe oder Klasse auf andere Benutzer mit gleichen oder ähnlichen Intentionen zu übertragen. Stereotypen werden in einer hierarchischen Struktur angeordnet. Dieses Verfahren erlaubt einerseits unbekannte Annahmen über den Benutzer durch die Klassenzugehörigkeit zu treffen und andererseits eine rasche Initialisierung eines Benutzermodells bei geringer Information über den individuellen Benutzer [3].

Stereotypen können nach Adelsberger [1] bereits zur Entwicklungszeit vom Programmentwickler nach anwendungsrelevanten Benutzermerkmalen bestimmt werden, denen dann die Benutzer zur Laufzeit des Systems anhand von Regeln zugeordnet werden. Eine andere Möglichkeit besteht darin, die Bildung neuer Stereotypen, bedingt durch das Auftreten neuer Eigenschaften, dem System zu überlassen. Dieses Verfahren wurde durch Lebowitz [17] in Form des Unimem-Algorithmus implementiert. Durch das automatische Berücksichtigen neuer Eigenschaften ist auch ein auf Unimem-basiertes System automatisch erweiterbar, ohne Änderungen am Algorithmus selbst vorzunehmen.

Da Benutzereigenschaften an untergeordnete Stereotypen vererbt werden, sind vererbte Annahmen mit einem Unsicherheitsfaktor (rating) versehen, die ausdrücken, wie sicher eine bestimmte Annahme einer Eigenschaft ist. Treten Widersprüche auf werden, beispielsweise bei numerisch gespeicherten Werten, Durchschnittswerte berechnet, Werte überschrieben oder bei einer festgelegten Anzahl von Widersprüchen neue Stereotypen gebildet. Diese Widerspruchsregelung weist den Nachteil auf, dass nicht alle Annahmen mit ähnlich hoher Berechtigung berücksichtigt werden können [27]. Ebenso kann nicht rückwirkend von speziellen auf allgemeine Eigenschaften geschlossen werden. Der Stereotypenansatz ist mit mittlerem Aufwand implementierbar und war bzw. ist für viele Domänen vollkommen ausreichend. Die Inferenzen dieses Ansatzes sind jedoch nicht so differenziert wie bei der Verwendung eines probabilistischen Ansatzes.

Ein anderer Ansatz mittels Stereotypen wird in der Arbeit von Morik [23] beschrieben, die Stereotypen als Inferenzregeln nutzen. In diesem System fließen nicht Attribute zur Selbsteinschätzung in die Stereotypenhierarchie. Vielmehr werden aus den Attributen Bewertungskriterien in Form von Inferenzregeln geschlossen und in ein Partnermodell übertragen. Aus diesen Inferenzregeln werden Benutzer bestimmten Stereotypen zugeordnet.

Ein weiterer Anwendungsfall des Stereotypenansatzes ist das kollaborative und inhaltsbasierte Filtern. Sowohl die Methode des Kollaborativen Filtern als auch die Methode des inhaltsbasierten Filtern bezeichnen eine Technik, die versucht die Gewichtung bzw. Verlässlichkeit von Hinweisen an den Benutzer zu erhöhen, indem es Benutzer in Gruppen einordnet und Lösungen empfiehlt, die anderen Mitgliedern der Gruppe weiterhelfen [22]. Insofern sind diese Methoden ebenfalls dem Stereotypenansatz zuzuordnen.

Beim inhaltsbasierten Filtern werden die Vorschläge um jene erweitert, die im semantischen Zusammenhang mit vorangegangenen Vorschlägen standen. Nachteilig an beiden Techniken ist, dass erst genügend Datensätze anderer Benutzer gespeichert sein müssen, um Vorschläge zu erzeugen.

Chin [5] führte erstmalig doppelte Stereotypen ein. Zum einen wird der bisherige Stereotypenansatz nach Benutzergruppen, die unterschiedliche Wissensgrade reflektieren, verwendet. Zum anderen werden die Wissensinhalte einer Domäne in Relation zu ihrem Schwierigkeitsgrad gesetzt. Ein Benutzerstereotyp beherrscht demnach unterschiedliche Grade an Konstrukten einer Schwierigkeitsebene. Jameson [11] berechnet aus der Reaktion des Benutzers auf Wissensinhalten mit bekanntem Schwierigkeitsgrad die Wahrscheinlichkeiten für diesen Benutzertyp und die zu erwartenden Reaktionen auf andere Wissensinhalte. Nach Jameson [10] ist die doppelte Stereotypisierung adäquat mit kognitiven Modellen, die ein Mensch nutzt, um sich neue Wissensbereiche anzueignen.

1.4 Kandidat-Kritik-Modell

Im System von Linden et al. [18] werden Domänenobjekte mithilfe einer gewichteten Zusammenfassung, aus bewerteten Constraints, eingeschätzt. Die Constraints stellen die Präferenzen der Benutzer dieses Systems dar, welche im Intervall von 0 bis 1, wobei 0 die schlechteste und 1 die beste Bewertung darstellt, abgebildet sind. Um auf die Bewertungskriterien des Benutzers zu schließen, werden zunächst Default-Annahmen getroffen, die dann durch jeweilige Benutzerkritiken weiter verfeinert werden. Durch diese Evidenz wird das Benutzermodell, das die Constraints speichert, aktuell gehalten. Diese Methode kann durch dieses Verfahren dem *reinforcement learning* zugeordnet werden.

Die multiattributive Objektbewertung erlaubt es, sowohl die Attributbewertungen als auch die Gesamtbewertung von Objekten vorherzusagen. In Domänen, in denen ein einzuschätzendes Objekt, viele Attribute aufweist, könnte dieser Ansatz erhebliche Akzeptanzprobleme aufweisen, da für jedes voneinander unabhängige Attribut Benutzeranfragen gestellt werden müssten, um auf die Gesamtbewertung zu schließen. Dieser Ansatz kann im Falle einer Erweiterung zwar automatisch neue Objekte für die Bewertung aufnehmen, die Gewichtung jedes einzelnen Objekts muss aber manuell festgelegt werden.

1.5 Präferenzen als Ziele

In der Arbeit von Beek [2] werden ausgehend von Default-Annahmen über Benutzerziele, die Eingaben des Benutzers verwendet, um weitere Ziele abzuleiten. Ein Ziel wird dann erreicht, wenn Unterziele als positiv bewertet wurden. In diesem reinforcement learning-Verfahren wird die Bewertung eines Attributes dadurch ausgedrückt, ob die Ausprägung das Ziel erfüllt, das dem Attribut entspricht. Damit kann eine Gesamtbewertung eines Objektes etwa durch die Anzahl der erreichten Ziele geschätzt werden. Nachteilig an diesem Ansatz ist, dass es keine Möglichkeit gibt, undefinierte Attribute zu bewerten. Somit ist der Implementationsaufwand sehr gering, die Inferenzbildung muss aber manuell festgelegt werden. Die Behandlung von Wahrscheinlichkeiten wird mit diesem Ansatz nicht unterstützt.

1.6 Bewertungskriterien als Ziele

Das System von Elzer et al. [6] greift die Methode der Präferenzen von Beek [2] auf, indem Bewertungskriterien als Präferenzen dargestellt werden. Elzer erweitert diese reinforcement learning-Methode um die Stärke einer Präferenz aus Sicht des Benutzers und um die Sicherheit einer Präferenz aus Sicht des Systems. Aufgrund der Ähnlichkeit der Attributsausprägungen mit den Zielausprägungen, können diese Präferenzen bewertet werden. Durch die zusätzliche Angabe der Stärke und Sicherheit einer Präferenz kann eine Gesamtbewertung eines Objektes berechnet werden. Dadurch sinkt jedoch die Erklärbarkeit einer Gesamteinschätzung durch die Einbeziehung der Wahrscheinlichkeiten ab. Eine Erweiterung ist durch die Definition der Zielausprägung von Attributen nur manuell durchzuführen.

1.7 Genetische Algorithmen

Genetische Algorithmen zählen zu den Hauptausprägungen evolutionärer Algorithmen [29]. Sie orientieren sich am Vorgang des natürlichen Evolutionsprozesses, welcher auf langfristige Verbesserungen angelegt ist. Der Algorithmus verwendet bei der Auswahl von Objekten, wie zum Beispiel die Ergebnisse von Suchanfragen, die evolutionären Operatoren Reproduktion, Varianten und Selektion. In einem iterativen Prozess entstehen dabei neue Generationen von Lösungen, deren Erfüllungsgrad zum Optimum, die Fitness, mit anderen Objekten verglichen wird. In diesem Prozess werden nur diejenigen Objekte verwendet, deren Fitness hoch genug ist. In jeder Generation bzw. jedem Iterationsschritt konvergieren die Objekte allmählich zum Optimum, so dass immer bessere Lösungen entstehen.

Vorteilhaft an diesem Algorithmus sind nach Strecker [29] hohe Verarbeitungsgeschwindigkeiten durch die Möglichkeit der parallelen Verarbeitung, die Einsatzfähigkeit für hochkomplexe Problemstellungen und die Kombinierbarkeit mit anderen Verfahren, beispielsweise zur Optimierung konnektionistischer Ansätze [20] oder als Lernverfahren in Neuronalen Netzen [9]. Demgegenüber ist der heuristische Charakter, dass in gegebener Zeit keine Optimallösungen erreicht werden, als nachteilig einzustufen. Da sich nur die fittesten Objekte sich vererben, erfolgt das Berücksichtigen neu hinzugekommener Attribute, im Falle einer Erweiterung, automatisch.

1.8 Evolvierende Agenten

Evolvierende Agenten werden erstmalig im System von Sheth und Maes [28] beschrieben. Jeder Agent verwaltet ein Bewertungsmodell des Benutzers hinsichtlich verschiedener Attributsausprägungen. Durch die Gewichtung von bestimmten Attributen wird das Benutzerinteresse hinreichend repräsentiert. Die Gesamtbewertung eines Objekts wird durch einfache Addition der gespeicherten Attributgewichte berechnet. Stimmen Gesamtbewertung und Benutzerinteresse annähernd überein, wird dem Benutzer ein Vorschlag unterbreitet. In einer sich anschließenden Bewertung des Vorschlags von Seitens des Benutzers erhalten die Attributsausprägungen positive bzw. negative *Fitnesspunkte*. Aufgrund der Rückmeldung des Benutzers wird dann die Einschätzung seiner Bewertungskriterien angepasst, indem die Agenten ihr lokales Modell auf den neuesten Stand bringen. Dieses Verfahren entspricht dem beschriebenen reinforcement learning.

Die Agenten können sich untereinander fortpflanzen. Die nächste Generation von Agenten besteht dann aus den fittesten der Eltern-Generation. Somit kommt es zu einer Verschmelzung von Benutzerinteressen, die widersprüchliche Ausprägungen enthalten können. Es wird allerdings nicht geklärt, wie diese Inkonsistenzen behandelt werden. Die Fortpflanzung erfolgt durch Kopie, Mutation und Kreuzung, gemäß dem Prinzip der genetischen Algorithmen. Dadurch können auch neu hinzukommende Attribute automatisch berücksichtigt werden.

Des Weiteren werden die Bewertungskriterien dezentral verwaltet. Für das beschriebene Dialogsystem ist die dezentrale Verwaltung als ungünstig einzuschätzen, da zur Berechnung der aktuellen Bewertungskriterien eines Benutzers sämtliche Agenten konsultiert werden müssen, d.h. die Verarbeitungsgeschwindigkeit sinkt dadurch ab. Auch hier tritt wieder das Problem der Inkonsistenz auf, welches nicht weiter behandelt wird. Besser wäre eine zentrale Verwaltung der Bewertungskriterien.

1.9 Symbolische Künstliche Intelligenz

Als Repräsentant des symbolischen Ansatzes zählen die Expertensysteme. In diesen Systemen wird das fachspezifische Wissen und die Schlussfolgerungsfähigkeit von Fachleuten durch kausal erklärbare Muster, in Form von Regeln oder Logiken, nachgebildet [15]. Die Inferenz basiert auf der Ableitung von Lösungen durch Manipulation der Symbole. Die funktionalen Zusammenhänge müssen explizit vorab festgelegt werden, so dass die Erweiterbarkeit manuell gelöst werden muss. Innerhalb von Expertensystemen, im Inferenzsystem, wird zwischen folgenden Formen der Wissensrepräsentation unterschieden: Aussagenlogik, Constraints, objektorientierte Darstellung, semantische Netze, Prädikatenlogik und Produktionsregelsysteme. Die Vorteile der symbolischen Wissensverarbeitung liegt bei Expertensystemen in der Trennung zwischen dem Wissen selbst und der Wissensverarbeitung in getrennten Komponenten. Die Verarbeitungsgeschwindigkeit ist demzufolge langsamer als bei anderen Methoden. Der Vorteil liegt aber in der vollständigen Berücksichtigung aller Attribute.

1.10 Fuzzy-Logik

Um Annahmen unter Unsicherheit ohne Probabilistik zu treffen, scheint die Fuzzy-Logik ein geeigneter Ansatz für die Benutzermodellierung zu sein. So verwendet beispielsweise Chin [5] Fuzzy-Regeln, um das Wissensniveau eines Benutzers einzuschätzen. Ein Ansatz mit Fuzzy-Logik sollte nach Jameson [12] gegenüber anderen Methoden zur Behandlung von Unsicherheit vorgezogen werden, falls das Ziehen von Schlussfolgerungen in Begriffen von linguistischen Variablen natürlich ist. Linguistische Variablen werden verwendet, um Zusammenhänge zwischen beobachtbarem Verhalten und den Benutzerinteressen zu beschreiben. Eine Dialogsteuerung auf dieser Basis in einem System, müsste nach Schäfer [27] den selben Ansatz der Fuzzy-Logik verwenden, so dass die Dialogsteuerung auf die modellierten Zusammenhänge zurückgreifen kann.

Wie Bewertungskriterien von Benutzern mit Hilfe von Fuzzy-Logik ausgedrückt werden können, wird in der Arbeit von Kölln [16] beschrieben. Ähnlich wie im System von Elzer et al. [6] gibt es im System von Kölln für jedes Attribut zwei Parameter. Die *Güte* gibt an, wie gut ein Attribut bewertet wird, und die *Relevanz* gibt an, wie wichtig das Attribut ist. Die Gesamtbewertung wird durch eine ungewichtete Zusammenfassung der beiden Attributparameter berechnet, wobei die Relevanz die Güte beeinflusst. Damit können die Attributbewertungen des Benutzers vorhergesagt werden. In der Arbeit von Popp und Lödel [25] werden ebenfalls mithilfe der Fuzzy-Logik Bewertungskriterien aufgestellt und daraus Zugehörigkeitsfunktionen ermittelt. Durch Zusammenfassung der Zugehörigkeitsfunktionen wird die Gesamtbewertung ermittelt. Der Ansatz mit Fuzzy-Logik gestattet durch die Zusammenfassung der Zugehörigkeitsfunktionen das automatische Berücksichtigen neuer hinzukommender Attribute, das eine Erweiterung des Systems mit neuen Eigenschaften gestattet.

1.11 Bayes'sche Netze

Die im Folgenden dargestellten Methoden zur Benutzermodellierung basieren auf den konnektionistischen Ansätzen, das heißt Aktionen werden als Knoten in einem Netzwerk repräsentiert.

Ein geeigneter Ansatz Schlussfolgerungen unter Unsicherheit zu ziehen, sind Bayes'sche Netze (BN). Dieser probabilistische Inferenzalgorithmus ermöglicht die systematische Berücksichtigung und differenzierte Darstellung von Unsicherheit. In BNs werden Benutzereigenschaften in Form von Wahrscheinlichkeitsverteilungen (Beliefs) dargestellt. Die Wahrscheinlichkeit gibt an, wie sicher eine Eigenschaft zutrifft. Das Netz verwaltet, gegenüber anderen Ansätzen, die Sicherheiten mit der aufeinanderfolgende Inferenzen gezogen werden. Solch ein Zusammenhang wird durch eine Tabelle bedingter Wahrscheinlichkeiten (Verbindungsmatrix) beschrieben.

Auf der Basis von Anfangswahrscheinlichkeiten über Benutzereigenschaften wird durch das Lösen des Netzes, einem Propagierungsalgorithmus, eine bestimmte Eigenschaft vorhergesagt. Je nach Revision oder Bestätigung werden die Wahrscheinlichkeiten weiter verfeinert. Der Bewertungsprozess wird in BNs dabei aus der multi-attributiven Objektbewertung gebildet, indem ein einzuschätzendes Objekt auf der Basis verschiedener Bewertungsdimensionen bewertet wird. Mit jeder Bewertungsdimension ist ein relatives Gewicht assoziiert. Die Gesamtbewertung ergibt sich als Summe der gewichteten Bewertungen des Objekts auf den einzelnen Bewertungsdimensionen. Um die genannten Gewichte einzuschätzen, verwendet Schäfer [27] die Interpretation verschiedener Dialogbeiträge des Benutzers.

Die Arbeit von Schäfer [27] erweitert die Bayes'schen Netze um eine Zeitmodellierung und zeigt, dass dynamische Bayes'sche Netze (DBN) ebenfalls als Inferenzmechanismus für Benutzermodellierungskomponenten adaptiver Dialogsysteme geeignet sind. So ist das darin beschriebene DBN in der Lage, Schlussfolgerungen unter Unsicherheit mit Berücksichtigung der Zeit, unter Verwendung von sogenannten Zeitscheiben, zu ziehen, da Benutzereigenschaften stark dynamisch sind. Die Abhängigkeiten zwischen den Zeitscheiben sind dann die genannten bedingten Wahrscheinlichkeiten.

Wie bei allen konnektionistischen Ansätzen ist der Implementationsaufwand gegenüber den symbolischen Ansätzen wesentlich höher, und die Verarbeitungsgeschwindigkeit nimmt exponentiell mit der Anzahl der einzuschätzenden Objekte zu. Jedoch können Inferenzen unter Unsicherheit gezogen werden. Auch das Hinzukommen neuer Objekte wird im Netz automatisch berücksichtigt und vollständig bearbeitet. Die Erklärbarkeit der inferierten Daten ist bei BNs durch die Beliefs mit mittlerem Aufwand nachvollziehbar.

1.12 Dempster-Shafer-Theorie

Ähnlich wie in Bayes'schen Netzen ist die Dempster-Shafer-Theorie (DST) eine weitere Methode zur Behandlung von Unsicherheit. Die Theorie stellt gegenüber BNs einen allgemeineren Ansatz dar, obgleich der Inferenzalgorithmus einen höheren Aufwand erfordert. Die DST wurde speziell für die Darstellung und Verarbeitung von *Unwissenheit* entwickelt. Das hat zur Folge, dass nicht wie bei BNs Anfangswahrscheinlichkeiten definiert werden müssen. Nachteilig ist allerdings die Erklärbarkeit von Inferenzen eines DST-basierten Algorithmus. Da mehrere Parameter (z.B. Beliefmasse und Plausibilität) verwendet werden, ist die Verständlichkeit und Erklärbarkeit nach Schäfer [27] schwieriger als bei der Verwendung von Bayes'schen Netzen.

1.13 Neuronale Netze

Auch Neuronale Netze wurden für die Einschätzung von Bewertungskriterien zur Modellierung des Benutzers verwendet. Das System von Karunanithi und Alspector [13] lernt mittels diesen Ansatzes die Bewertungen von Benutzern kennen und kann nach einer Trainingsphase die Gesamtbewertung eines Objekts unsupervised vorhersagen. Allerdings ist die Vorhersage der Beurteilung von einzelnen Attributen nicht vorgesehen. Nachteilig an diesem Ansatz ist die erforderliche überwachte Trainingsphase. Daher muss die Benutzermodellierungskomponente eines solchen Systems auch schon zu Beginn des Dialoges noch sehr unsichere Einschätzungen über den Benutzer liefern. Desweiteren nachteilig ist der intransparente Inferenzprozess. Die Erklärbarkeit von Systementscheidungen ist in Neuronalen Netzen nach Schäfer [27] recht gering. Die Arbeit von Lindner [19] stellt einen konnektionistischen Ansatz für adaptive Anwendungen vor. Dort besteht das System aus untereinander kommunizierenden Neuronen eines neuronalen Netzes, deren Gewichtung ihre Relevanz angeben. Die Neuronen beinhalten zur Signalverarbeitung einen Propagierungs-, eine Aktivierungs- und eine Ausgabefunktion. Die Propagierungsfunktion summiert die eintreffenden, gewichteten Signale. Aktivierungs- und Ausgabefunktion stellen eine Schwellenfunktion dar. Aus den Aktivierungen und Gewichten der Verbindungen zwischen den Neuronen lassen sich Rückschlüsse über den Benutzer ziehen. Niedrige Gewichte werden dabei einem Erfahrenen und hohe einem Anfänger

gerecht, wobei ständige Änderungen in einem neuronalen Netz vorgenommen werden [19]. Das Wissen über den Benutzer ist in neuronalen Netzen über alle Knoten des Netzes verteilt. Sollen Benutzermodelle, als explizite Informationsstruktur, Wissen aus Neuronalen Netzen akquirieren, müssen die subsymbolischen Darstellungen auf eine symbolische Struktur abgebildet werden.

1.14 Spreading-Activation-Ansatz

In der Arbeit von Thies [30] wird ein weiterer konnektionistischer Ansatz beschrieben. Im Gegensatz zu den Neuronalen Netzen, werden hier Aktivierungen symbolisch weitergeleitet und verarbeitet. Die Ausbreitung der Aktivierungen erfolgt nach Erfüllung spezifischer Constraints entlang definierter Verbindungen. Die Merkmale dieses Ansatzes sind identisch mit denen der Neuronalen Netze.

1.15 Hybridsysteme

Hybride Ansätze in der Benutzermodellierung kombinieren die eben beschriebenen einzelnen Ansätze. In der Arbeit von Blank [3] wird ein Kombinationsmodell aus neuronalen Netzen und dem Stereotypen-Ansatz beschrieben, indem stark schematisierte Stereotypen stärker individualisiert werden.

In Hybridsystemem können die Vorteile einzelner Ansätze bezüglich der in Tabelle 3 aufgeführten Merkmale verwendet werden, um die jeweils geeignete adaptive Methode vorteilhaft domänenabhängig anzupassen.

2 Weiterentwicklung in einem Shell-System

In Tabelle 3 sind die Merkmale der einzelnen adaptiven Methoden zusammenfassend dargestellt.

Dabei sind verschiedene realisierungsrelevante Kriterien mit Vor- und Nachteilen bewertet. In adaptiven Anwendungen soll jedoch der eine oder andere Vorteil von adaptiven Methoden, wie leichte Lernfähigkeit, hohe Verarbeitungsgeschwindigkeit, automatisch vollständige Inferenzbildung und eine automatische Erweiterbarkeit, unter Duldung mancher Nachteile, wie hoher Implementationsaufwand, schwere Erklärbarkeit und schwere Wissensrepräsentation, implementiert werden. Das Ziel einer neuen Benutzermodellierungskomponente muss es also sein, neben der Implementierung der Anforderungen an Benutzermodellierungs-Shell-Systeme [14], adäquate Algorithmen anzubieten, welche flexibel, d.h. ohne aufwändige Implementationsänderungen ein bestehendes adaptives System um zusätzliche Wissens Elemente erweitern und zudem zeiteffektiv arbeiten. Somit soll versucht werden, lediglich die Vorteile der bestehenden Methoden zu nutzen.

Eine anwendungsübergreifende, domänenunabhängige Komponente, welche neben Klassifikationsmethoden das adaptive Verhalten der Anwendung definiert und dabei für den jeweiligen Anwendungsfall bzw. Problemfall die vorteilhafteren adaptiven Methoden ansteuert, bilden die neuentwickelten Controltypes. Die Controltypes bilden ein Rahmensystem zur Modellierung von adaptiven Methoden zur Gestaltung adaptiver Anwendungen. In einer hybriden Modellierung können somit die Vorteile mehrerer adaptiver Methoden kombiniert werden. Hierfür stellt eine entwickelte Beschreibungssprache die notwendigen Befehlsstrukturen bereit. Mittels Controltyp-Funktionen, -Methoden und -Prozeduren kann mit einem Benutzermodell kommuniziert werden, um die benutzerspezifischen Informationen zur Modellierung einer eigenen adaptiven Methode nutzbar zu machen. Folgende Auflistung zeigt die wesentlichsten Funktionen der Controltypes. Berücksichtigt werden:

- die *Anzahl der Mausklicks* und die *Aufenthaltsdauer* auf einer Bildschirmseite zur **Navigationsunterstützung**
- die *Erfolgsrate*, *Häufigkeit*, *Vollständigkeit* und *Auswahl der Übungen* für verschiedene **adaptive Interventionen**, wie angepasste Hinweise, generative Übungen, Planerkennung etc. Dazu zählt ebenso das
- *Anfordern von spezifischen Programmfunktionen und Hintergrundinformationen* (Interaktionbeobachtung).
- Anhand der Daten aus dem Benutzermodell kann der Benutzer einer stereotypischen Benutzerklasse zugeordnet werden: **Benutzerklassifikation**

Das Beschreibungsschema dieser Sprache geht von einer hierarchischen Gliederung aus. So wird der zu erstellende Bildschirminhalt als *page* bezeichnet und stellt den Rahmen für in *Tags* eingeschlossene Befehlsstrukturen für die Dialogaufbereitung zur Verfügung. Gegenwärtige marktführende Entwicklungsumgebungen bzw. Autorensysteme für vorrangig multimediale Lehr- und Lernsysteme, wie Asymetrix Toolbook oder Macromedia Director bevorzugen die Metapher von Bildschirmseiten oder Frames. Aus diesem Grund wurde für die Controltypes dieser Strukturierungsbereich explizit mit *page* vorgesehen, um eine kompatible Form der Anforderungsdefinitionen zu schaffen.

Im Folgenden ist ein einfaches Beispiel aus der Regelbeschreibungssprache für eine einfache Dialogmodellierung aufgeführt (Source 1), welches besagt, dass ab dem 2-maligen Absolvieren der Übungssituation 1 der Dialog mit der ID 0 ausgegeben werden soll, ansonsten der Dialog mit der ID 1.

Source 1 (Beschreibungssprache)

```
<if>(#um.game[1].count==2)<dialog>id(0)</dialog><else><dialog>id(1)</dialog></if>
```

Dieser Auszug greift mittels der Funktion *#um* auf das Benutzermodell-Attribut *game[1].count* zurück, welches ausdrückt, wie oft die Übung 1 absolviert wurde. Aus dem gezogenen Vergleich wird entweder Dialog ID 0 oder Dialog ID 1 zur Anzeige gebracht. In einem gesonderten Abschnitt der Controltypes (*;dialogs*) wird die Zuordnung zwischen ID-Nummer und konkretem Ausgabebetext hergestellt.

Um Controltype-Regeln anzuwenden, müssen diese auf die gespeicherten Daten der Benutzermodelle zugreifen. Für einen einfachen Zugriff auf Benutzermodelleinträge ist die Controltype-Funktion *#um.name* implementiert, wobei *name* das Attribut des Benutzermodells bezeichnet. Neben dieser Funktion definiert eine Zuordnungstabelle zwischen den Befehlen und den Modelleinträgen die Schnittstelle zwischen Benutzermodell und Controltypes, die diesen Zugriff erst ermöglicht. In einem separaten Abschnitt der Controltype-Definition (*;interface*) wird die Zuordnung zwischen Controltype-Funktion und Benutzermodell-Eintrag explizit formuliert:

```
lastSuccess(i) → #game[i].success
```

Dieses Beispiel definiert die Relation der Funktion *lastSuccess* zum Benutzermodell-Attribut *#game[1..6].success*. Somit lassen sich eigene Funktionen definieren, die in der Regelbeschreibungssprache Anwendung finden können. Mittels der Befehlsstrukturen der Controltypes können nunmehr auch die einzelnen adaptiven Methoden formuliert werden.

Zur weiteren Beschreibung der Controltypes sei auf die Arbeit von Nebel et al. [24] verwiesen.

Literatur

- [1] Adelsberger H.H. et al. Computer Assisted Learning I: Benutzermodellierung, Universität Essen, 2001 auf http://beta1.wi-inf.uni-essen.de/teaching/lectures/ws0102/folien_cal/6-benutzer.pdf vom 13.12.2001
- [2] van Beek P. A model for generating better explanations, in: Proceedings of the 25th Conference of the Association for Computational Linguistics (ACL), S. 215220, Stanford, California, 1987
- [3] Blank KH. Benutzermodellierung für adaptive interaktive Systeme: Architektur, Methoden, Werkzeuge und Anwendungen, Sankt Augustin, 1996
- [4] Blutner R Nichtmonotones Schließen und neuronale Netze, in: Busemann S., Harbusch K., Wermter S. (Hrsg.), Hybride konnektionistische, statistische und regelbasierte Ansätze zur Verarbeitung natürlicher Sprache. DFKI-D-98-03, Saarbrücken 1998 (ISSN 0946-0098), pp. 65-75
- [5] Chin D.N. KNOE: Modeling what the user knows in UC, in: Kobsa A., Wahlster W. (Hrsg.), User models in dialog systems (S. 74107), Berlin: Springer, 1989
- [6] Elzer S., ChuCarroll J., Carberry S. Recognizing and utilizing user preferences in collaborative consultation dialogues, in Kobsa A., Litman D. (Hrsg.), Proceedings of the Fourth International Conference on User Modeling (S. 1924), Boston, MA: User Modeling, Inc., 1994
- [7] Hahn K. Hauptseminar: Intelligente Lehrsysteme, Technische Universität München, Fakultät für Informatik auf <http://wwwpaul.in.tum.de/seminare/lehrsystemeSS98/Vortrag01/>
- [8] Harrer A.G. Unterstützung von Lerngemeinschaften in verteilten intelligenten Lehrsystemen, Dissertation, Technische Universität München, Institut für Informatik, 2000
- [9] Heistermann J. Genetische Algorithmen und ihre Anwendung als Lernverfahren für neuronale Netze, Dissertation, Universität Frankfurt, 1993
- [10] Jameson A. Generalizing the Double-Stereotype Approach: a Psychological Perspective, in: Proceedings of the Third International Workshop on User Modeling, DFKI Document D-92-17, Saarbrücken, 1992
- [11] Jameson A. Knowing what others know, Promotionsschrift, Nijmegen, 1992
- [12] Jameson A. Numerical uncertainty management in user and student modeling: An overview of systems and issues, User Modeling and UserAdapted Interaction, Vol. 5, 1996
- [13] Karunanithi N., Alspecter J. Featurebased and cliquebased user models for movie selection, in: Carberry S., Zukerman I. (Hrsg.), proceedings of the Fifth International Conference on User Modeling, 2934, Boston, MA: User Modeling, Inc., 1996
- [14] Kobsa A. Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen. Tagungsband zur 17. Fachtagung für Künstliche Intelligenz (KI 93). Humboldt-Universität zu Berlin. 1993
- [15] Koller A. Hybride Wissensverarbeitung in der präventivmedizinischen Diagnostik, Dissertation, Universität-GH Paderborn, 1999
- [16] Kölln M.E. Employing user attitudes in text planning, in: Proceedings of the Fifth European Workshop on Natural Language Generation, 163179, Leiden, Niederlande, 1995
- [17] Lebowitz M. Concept learning in a Rich Input Domain: Generalization-Based-Memory, in: Boulay B. Advance in Artificial Intelligence-II, Elsevier Science Publishers B. V., 1986
- [18] Linden G., Hanks S., Lesh, N. Interactive assessment of user preference models: The Automated Travel Assistant, in: Jameson A., Paris C., Tasso C. (Hrsg.), User modeling: Proceedings of the Sixth International Conference, UM97, 6778, Vienna, New York: Springer, 1997
- [19] Lindner HG, Bodendorf F. Ein neuronales Konzept für adaptive Anwendungen, in: Kobsa A. Pohl W. (Hrsg.), Arbeitspapiere des Workshops Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Berlin, 1993
- [20] Mandischer M. Genetische Algorithmen zur Optimierung Konnektionistischer Modelle, Master Thesen, Universität Dortmund, 1992
- [21] Mazur D Reinforcement Learning, 2001 auf <http://www.igi.tugraz.at/lehre/seminarB/SS01/Mazur1.pdf> vom 09.09.2002
- [22] Mertens P. Wie lernt der Mensch den Computer kennen?, Bayerisches Forschungszentrum für wissensbasierte Systeme, Forwiss, Erlangen; München; Passau, 1999
- [23] Morik K, Rollinger C.R. Partnermodellierung im Evidenzraum, in. Neumann B. (Hrsg.), GWAI-83: 7th German Workshop of Artificial Intelligence, Berlin, Heidelberg, Springer, 1983
- [24] Nebel IT., Paschke R. Implementation und Evaluation eines adaptives Diabetes-Schulungsprogramms auf der Basis der Benutzermodellierung mittels Controltypes, KI-Zeitschrift, Gesellschaft für Informatik, Vol. 02, 40-43, 2002
- [25] Popp H., Lödel D. Fuzzy techniques and user modeling in sales assistants, User Modeling and UserAdapted Interaction, Vol. 5, 349370, 1996
- [26] Rich E. Building and Exploiting User Models, PhD Thesis, Cornegie-Mellon University, 1979
- [27] Schäfer R. Benutzermodellierung mit dynamischen Bayes'schen Netzen als Grundlage adaptiver Dialogsysteme, Dissertation, Universität Saarbrücken, 1998
- [28] Sheth B., Maes P. Evolving agents for personalized information filtering, in: Proceedings of the Ninth Conference on Artificial Intelligence for Applications, CAIA-93 (S. 345352). Orlando, FL., 1993
- [29] Strecker S, Schwickert AC. Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht, Arbeitspapiere Wirtschaftsinformatik Nr. 4/1997, Justus-Liebig-Universität Gießen, 1997
- [30] Thies M.A Planbasierte Hilfeverfahren für direkt-manipulierbare Systeme: Erkennung, Vervollständigung und Visualisierung von Interaktionsplänen, Dissertation, Universität Stuttgart, 1994