

# Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen<sup>1</sup>

Alfred Kobsa<sup>2</sup>  
AG Wissensbasierte Informationssysteme  
Informationswissenschaft, Universität Konstanz  
Postfach 5560-D73, D-78434 Konstanz  
kobsa@inf-wiss.uni-konstanz.de

## 1 Einleitung

Interaktive Softwaresysteme werden von Benutzern<sup>3</sup> mit unterschiedlichsten Zielen, Interessen, Fähigkeiten, Erfahrungsgraden und Präferenzen verwendet. Um einem möglichst breiten Personenkreis zugänglich zu sein, bieten viele derzeit erhältliche Programme bereits die Möglichkeit, daß Benutzer (oder Systemadministratoren) in bestimmtem Ausmaß eine Anpassung des Programms an die jeweiligen individuellen Präferenzen vornehmen können. (Dies erfolgt meistens durch das Ausfüllen von Präferenzmenüs oder das Editieren von Profildateien.)

Diese *benutzerinitiierte und -selektierte* Adaption<sup>4</sup> ist meistens nicht ausreichend, um Computersysteme benutzerangepaßter zu machen, und wird auch wenig verwendet (siehe etwa Oppermann 1991). Der Grund dafür ist, daß in vielen Fällen die erwünschten Adaptionleistungen viel zu umfangreich sind, als daß der Benutzer jede einzelne von sich aus anfordern könnte oder wollte, und daß auch oft der Benutzer einfach nicht das notwendige Wissen über die Domäne, die gemachten Fehler oder die Anpassungsfähigkeiten des Computersystems hat, um die Notwendigkeit einer Anpassung erkennen und die optimale Anpassung bestimmen zu können. Die folgenden Anpassungsleistungen sind beispielsweise erwünscht (siehe etwa Kobsa & Wahlster 1989, Kok 1991, Kobsa 1991a, Kobsa & Pohl 1993, Schneider-Hufschmidt et al. 1993):

- Hypertext-Systeme, Erklärungskomponenten von Expertensystemen und natürlichsprachliche Auskunftssysteme sollten ihre Ausgabertexte an den (terminologischen) Wissensstand des Benutzers anpassen können.
- Information-Retrieval-Systeme, Datenbanken und Hypertext-Systeme sollten Navigationshilfen anbieten, die die Interessen, Ziele und Pläne des Benutzers berücksichtigen.
- Information-Retrieval- und Informationsfilter-Systeme sollten die Informationsbedürfnisse der jeweiligen Benutzer berücksichtigen, die sich aus deren Zielen und Interessen ergeben.

---

<sup>1</sup> Die vorliegende Arbeit entstand im Rahmen des Projekts BGP-MS, das von der Deutschen Forschungsgemeinschaft (Fördernummer Ko-1044/4-2) und von der Universität Konstanz (Fördernummer AFF 17/92) unterstützt wird. Für Kommentare und Anmerkungen danke ich Jörg-Cyril Höhle, Detlef Küpper und Wolfgang Pohl.

<sup>2</sup> Derzeit Gastforscher am Computer Science Department der Columbia University, New York.

<sup>3</sup> Statt 'Benutzer' lese man wahlweise auch 'Benutzerin'.

<sup>4</sup> Vgl. die Klassifikation von Adaptionarten in Dieterich et al. (1993).

- Natürlichsprachliche Systeme sollten bei der Generierung referentieller Ausdrücke das gemeinsame Wissen von System und Benutzer berücksichtigen.
- Die Lehrstrategien von tutoriellen Systemen sollten das jeweilige Verständnis und die Fertigkeiten des Lernenden miteinbeziehen.
- Online-Hilfesysteme sollten ihre Erklärungen an die Ziele, Pläne, Mißverständnisse und den Wissensstand des Benutzers anpassen.
- Das Layout, die Interaktionsoptionen und -formen von Benutzerschnittstellen sollten an die unterschiedlichen Aufgaben, Fähigkeiten und Präferenzen von Benutzern angepaßt sein.

Auch für andere komplexe technische Geräte mit inhomogenem Benutzerkreis, wie etwa Autos, Büromaschinen, Unterhaltungselektronik und Haushaltsgeräte, werden bereits Anpassungsmöglichkeiten an den jeweiligen Benutzer in Erwägung gezogen.

Da der Umfang bzw. die Art der notwendigen Anpassungsleistungen in vielen Fällen eine (ausschließlich) benutzerinitiierte und -selektierte Adaption ausschließt, werden schon seit längerer Zeit Möglichkeiten untersucht, interaktive Computersysteme zu befähigen, die Notwendigkeit einer Anpassung an den jeweiligen Benutzer *selbst zu erkennen*. Dafür müssen Computersysteme *Annahmen* über den Benutzer bilden und diese bei komplexerer Adaption in einem *Benutzermodell* speichern und verwalten. Ein Benutzermodell ist dabei eine Wissensquelle des Systems, "die Annahmen über alle Benutzeraspekte enthält, die für das Dialogverhalten des Systems relevant sind" (Wahlster & Kobsa 1989). Aufgabe einer *Benutzermodellierungskomponente* in einem interaktiven System ist es, ein Benutzermodell während der Interaktion mit dem Benutzer aufzubauen und laufend zu aktualisieren, aus Erstannahmen weitergehende Annahmen zu inferieren, die Konsistenz der Annahmen sicherzustellen, und andere Systemkomponenten auf Anfrage mit den derzeitigen Annahmen über den Benutzer zu versorgen.

Im Bereich der Benutzermodellierung (Kobsa & Wahlster 1989; Kok 1991, Kobsa 1991a, 93; McTear 1993, Kobsa & Pohl 1993) werden seit gut 10 Jahren Methoden entwickelt, mit deren Hilfe Benutzermodellierungskomponenten diese Aufgaben erfüllen können. Dabei werden zum einen verschiedene KI-Techniken verwendet und an die speziellen Bedürfnisse der Benutzermodellierung angepaßt. Zum anderen wurden in der Benutzermodellierung aber auch eine Reihe von eigenen Methoden entwickelt. In den folgenden Abschnitten sollen einige dieser Methoden beschrieben werden, getrennt nach den Problembereichen 'Aufbau des Benutzermodells während der Interaktion mit dem Benutzer', 'Repräsentation von Annahmen über den Benutzer, Schlußfolgerungen daraus und Konsistenzerhaltung' sowie 'Verwendung von Benutzermodellen in interaktiven Softwaresystemen'.

## **2 Automatischer Aufbau eines Benutzermodells**

Im Laufe der letzten zehn Jahre wurde eine große Anzahl von Methoden entwickelt, mit deren Hilfe ein interaktives Softwaresystem selbständig Annahmen über den gegenwärtigen Benutzer aufbauen kann. Welche davon in einem konkreten Anwendungssystem einsetzbar sind, hängt sehr stark von der Art des Systems ab, insbesondere von dessen Anwendungsdomäne, von der Art der Aufgaben, die Benutzer unter Verwendung des Systems erfüllen sollen, und teilweise auch von seiner Benutzerschnittstelle. Die meisten bisher entwickelten adaptiven Softwaresysteme setzen mehrere verschiedene Methoden ein.

Die Methoden zum Benutzermodelleraufbau lassen sich nach mehreren Dimensionen klassifizieren, von denen in Hinblick auf die praktische Einsetzbarkeit die Unterscheidung nach folgenden Kriterien am interessantesten erscheint:

- Art der getroffenen Annahmen über den Benutzer,
- Integriertheit des Benutzermodellenerwerbs in den normalen Mensch-Maschine-Dialog,
- verwendete Techniken,
- Beobachtungsnähe der getroffenen Annahmen, und
- Sicherheit der getroffenen Annahmen.

Im folgenden werden Benutzermodell-Erwerbsmethoden anhand dieser Klassifikation beschrieben. Weitere oder andere Dimensionen oder Bezeichnungen dafür finden sich etwa in Kass (1991) oder Chin (1993).

## 2.1 Art der getroffenen Annahmen über den Benutzer

Von bisher entwickelten Systemen werden üblicherweise eine oder mehrere der folgende Arten von Annahmen über den Benutzer gemacht (nach abnehmender Häufigkeit geordnet):

- *Vorhandenes oder fehlendes Wissen* des Benutzers, etwa um an das Benutzerwissen angepaßte Erklärungen, Beschreibungen oder Bildschirmlayouts zu generieren;
- *Ziele und Pläne* des Benutzers, etwa um Navigationshilfen oder relevante Zusatzinformationen zu geben;
- *Benutzerpräferenzen*, etwa zur Informationsselektion für den Benutzer;
- *Benutzerfähigkeiten*, etwa zur Anpassung von Benutzerschnittstellen;
- *Benutzermißverständnisse*, etwa zur Generierung von angepaßten Erklärungstexten oder zur Generierung von Übungsaufgaben.

## 2.2 Integriertheit des Benutzermodellenerwerbs

Der Erwerb eines Benutzermodells kann während der normalen Interaktion “im Hintergrund” stattfinden (integrierter Erwerb), oder *getrennt* von der normalen Interaktion des Anwendungssystems mit dem Benutzer erfolgen (separierter Erwerb). Der integrierte Erwerb eines Benutzermodells bleibt für den Benutzer weitgehend “unsichtbar”, da er ausschließlich auf Basis des regulären Dialogs mit dem Anwendungssystem stattfindet. Es werden dabei praktisch alle der in Abschnitt 2.3 beschriebenen Erwerbstechniken eingesetzt.

Ein separierter Erwerb des Benutzermodells basiert üblicherweise auf einem eigenen Dialog mit dem Benutzer. Dieser Erwerbsdialog kann vor der allerersten Interaktion mit dem Benutzer stattfinden; er kann aber auch (zusätzlich) mehrfach in die normale Interaktion eingeschoben werden. Je nachdem, ob die Initiative zum Benutzermodellenerwerb beim Benutzer oder beim System liegt, kann man zwischen den folgenden beiden Erwerbsarten unterscheiden:

- *Systemgesteuerter Erwerb*

Hierzu gehören vor allem Eingangsgespräche (Rich 1979ab, Koller 1993, Sanderson & Treu 1993) oder Eingangstests (Nwana 1991), die der Benutzer durchlaufen muß, bevor er ein System verwenden kann. Manche Systeme (etwa Sleeman 1985) verlangen auch, daß der Benutzer während der Normalinteraktion Zwischenfragen des Systems beantwortet, damit das System das Benutzermodell ergänzen und benutzerangepaßt reagieren kann. Beide Formen des systemgesteuerten separierten Benutzermodellerrwerbs können von Benutzern als störend empfunden werden, insbesondere unter Zeitdruck.

- *Benutzergesteuerter Erwerb*

Hierzu gehört vor allem das Ausfüllen von Profil- oder Präferenzdateien durch den Benutzer. Ihre Inhalte können bereits das Benutzermodell darstellen (wie das vielfach schon bei neuen Softwaresystemen der Fall ist) oder indirekt für den Benutzermodelleraufbau verwendet werden (wie etwa bei Boyle & Encarnacion, im Erscheinen). Solche Dateien enthalten meist Default-Werte und der Benutzer kann selbst entscheiden, wann und ob er die normale Interaktion unterbrechen und diese Standardangaben verändern möchte. Da die Initiative dafür ausschließlich beim Benutzer liegt, ist die Störung für ihn wohl geringer als bei einer systemgesteuerten Erwerbsphase (vgl. Oppermann 1992, Grunst et al. 1993).

Ein von der normalen Interaktion separierter Erwerb kann aber auch darin bestehen, daß ein Anwendungssystem auf ein *bereits existierendes* Benutzermodell zurückgreift. Ein solches kann vom System selbst in früheren Interaktionen mit dem Benutzer aufgebaut worden sein, oder auch aus der Interaktion des Benutzers mit anderen Softwaresystemen stammen. So kann etwa ein Datenbanksystem Annahmen über die Interessen des Benutzers in bezug auf die Inhalte der Datenbank bilden, die dann zur Filterung von Bulletin-Board-Artikeln und Electronic-Mail-Nachrichten über diese Datenbank verwendet werden (vgl. Kass & Stadnyk 1992). Orwant (1991) stellt die Idee in den Raum, daß alle Programme eines lokalen Netzwerks Annahmen über ihre Benutzer machen und an einen zentralen Benutzermodell-Server schicken, dessen gespeicherte Benutzermodelle wiederum von allen Programmen abgefragt werden können. Neal (1989) untersucht die Verwendung von Testsoftware (insbesondere von Computerspielen) zur Bestimmung bestimmter Benutzerfähigkeiten und zum Aufbau eines entsprechenden Benutzermodells für die eigentlichen Anwendungssysteme.

## **2.3 Verwendete Techniken**

Im Bereich der Benutzermodellierung wurden bislang eine große Anzahl von Mitteln verwendet, um aus der Interaktion mit Benutzern Annahmen über deren Wissen, Ziele, Pläne, Mißverständnisse, Präferenzen, Fähigkeiten etc. zu gewinnen. Dabei läßt sich eine Reihe von häufig verwendeten Techniken unterscheiden, die im folgenden beschrieben werden.

### **2.3.1 Primäre Erwerbsheuristiken**

Unter primären Erwerbsheuristiken sollen hier Regeln zum Aufbau von Benutzermodellen auf Basis hauptsächlich der *unmittelbaren* Interaktion mit dem Benutzer verstanden werden. Solche Heuristiken sind üblicherweise stark domänenabhängig. In einer Zugauskunftsdomäne (Allen 1979) läßt sich etwa aus der Benutzerfrage des Typs "Wann fährt der nächste Zug nach x"

schließen, daß der Benutzer mit dem nächsten Zug nach x fahren will. In einer Bahnspeditionsdomäne (wie etwa in Allen & Schubert 1993) ist diese Annahme nicht mehr legitim; der Benutzer will hier vielleicht einen Container nach x verladen oder Waggon dorthin verschicken.

Es gibt aber auch eine Reihe von domänenunabhängigen Erwerbsheuristiken, wie etwa die folgenden:

- *Korrekte Verwendung*: “Objekte (etwa Betriebssystemkommandos, mathematische Operationen, Begriffe), die der Benutzer korrekt verwendet, kennt er auch” (Chin 1989, Nwana 1991, Sukaviriya & Foley 1993).
- *Inkorrekte Verwendung*: “Objekte, die der Benutzer inkorrekt verwendet, kennt er nicht” (Quilici 1989, Hirschmann 1990).
- *Erklärungsanforderung*: “Objekte, über die der Benutzer Erklärungen anfordert, kennt er nicht” (Chin 1989; Boyle & Encarnacion, im Erscheinen).
- *Detaillierungsanforderung*: “Objekte, über die der Benutzer mehr Details wissen möchte, kennt er” (Boyle & Encarnacion, im Erscheinen).
- *Feedback*: “Wurde eine Systemausgabe unter Verwendung bestimmter Annahmen aus dem Benutzermodell erzeugt und kommt vom Benutzer positives/negatives Feedback, so verstärkte/verringere die Plausibilität dieser Annahmen” (Rich 1979ab).

Einige dieser Heuristiken lassen sich unter sprechakttheoretischen Gesichtspunkten verallgemeinern, wobei die Annahmen über den Benutzer als ihm zuschreibbare Handlungsvoraussetzungen interpretiert werden (siehe Pohl et al. 1993).

### 2.3.2 Stereotype

Der Stereotypenansatz wurde von Rich (1979ab) in die Benutzermodellierung eingeführt und später etwa von Chin (1989) erweitert. Er stellt eine beliebte Technik zum Erwerb von Benutzermodellen in solchen Anwendungsbereichen dar, in denen bald nach Dialogbeginn Annahmen über den Benutzer getroffen werden müssen, dabei aber ein bestimmtes Ausmaß von Fehlern in Kauf genommen werden kann. Das Aufstellen von Stereotypen für die Benutzer eines Anwendungssystems erfolgt in drei Schritten:

- *Identifikation von Benutzeruntergruppen*: Es müssen Untergruppen in der Benutzermenge identifiziert werden, deren Mitglieder üblicherweise bestimmte homogene anwendungsrelevante Merkmale besitzen.
- *Identifikation von Schlüsselmerkmalen*: Es müssen charakteristische Merkmale für die Zuordenbarkeit von Benutzern zu diesen Untergruppen identifiziert werden. Das Vorhandensein dieser Merkmale beim gegenwärtigen Benutzer muß vom Anwendungssystem beobachtet werden können.
- *Repräsentation in (hierarchisch geordneten) Stereotypen*: Die anwendungsrelevanten Charakteristika der identifizierten Benutzeruntergruppen müssen in einem geeigneten Repräsentationsschema (siehe Abschnitt 3) formal dargestellt werden. Die Gesamtheit aller repräsentierten Eigenschaften einer Benutzeruntergruppe wird als Stereotyp für diese Untergruppe bezeichnet. Wenn die Inhalte eines Stereotyps eine Untermenge der Inhalte eines anderen Stereotyps bilden, können Stereotypenhierarchien gebildet werden.

### **2.3.3 Ziel- und Planerkennung mit Planbibliotheken**

Aufgabe von Planerkennungstechniken ist es, aus der momentanen Benutzereingabe sowohl das vermutliche Ziel des Benutzers zu erkennen als auch die zu erwartenden nachfolgenden Planschritte, mit denen der Benutzer dieses Ziel zu erreichen sucht. In Anwendungsdomänen, in denen es nur eine begrenzte Anzahl von möglichen Benutzerzielen und -plänen gibt, können diese in sogenannten Planbibliotheken repräsentiert werden. Planerkennungsmechanismen vergleichen dann die Benutzereingaben kontinuierlich mit den gespeicherten Plänen und schränken die Menge der möglichen Interpretationen für diese Eingabe laufend ein (siehe etwa Schwab 1989).

### **2.3.4 Ziel- und Planerkennung durch Plankomposition**

Bei diesem Ansatz besitzt das System eine Bibliothek aller möglichen Benutzeraktionen, zusammen mit den Vorbedingungen und Effekten ihrer Ausführung (siehe etwa Carberry 1989). Die Benutzereingabe wird laufend verfolgt und ergänzt um alle möglichen nachfolgenden Benutzeraktionen, deren Vorbedingungen durch die Effekte der vorhergehenden Aktionen erfüllt sind und die auch alle beobachteten Benutzeraktionen als (indirekte) Vorbedingung haben. Diese Menge stellt die möglichen Planinterpretationen für die beobachteten Benutzeraktionen dar; sie wird im Verlauf der Interaktion kontinuierlich eingeschränkt.

### **2.3.5 Fehlerbibliotheken**

In Anwendungsbereichen, in denen ein Großteil der auftretenden Benutzerfehler auf eine begrenzte Anzahl von Fehlerarten zurückgeführt werden kann (siehe etwa Genesereth 1982, Hirschmann 1990) können die Charakteristika dieser Fehlerarten in sogenannten Fehlerbibliotheken repräsentiert werden. Fehlererkennungsmechanismen vergleichen dann kontinuierlich die Benutzereingabe mit den gespeicherten Fehlertypen.

### **2.3.6 Schlußfolgerungen**

Schlußfolgerungen sind ebenfalls zumeist Heuristiken zur Bildung von Annahmen über den Benutzer. Im Gegensatz zu den primären Erwerbsheuristiken basieren Schlußfolgerungen aber nicht auf der unmittelbaren Interaktion mit dem Benutzer, sondern hauptsächlich auf bereits gebildeten Annahmen. Schlußfolgerungen sind gleichfalls sehr domänenabhängig und müssen als Inferenzregeln in einem geeigneten Repräsentationsschema formal dargestellt werden.

## **2.4 Sicherheit der getroffenen Annahmen**

Die von Erwerbsheuristiken gebildeten Annahmen über den Benutzer sind normalerweise mit Unsicherheit behaftet. Der Grad der Unsicherheit ist stark domänenabhängig. Üblicherweise wird aber auch eine Abhängigkeit von der Beobachtungsnähe der verwendeten Erwerbstechnik und der "Präzision" der verwendeten Erwerbsregel angenommen. Annahmen, die auf unmittelbaren Beobachtungen in der Benutzerinteraktion beruhen (wie etwa Annahmen, die mit primären Erwerbsheuristiken gebildet werden) wird meistens eine höhere Sicherheit zugebilligt als

Annahmen, die durch Inferenzketten gebildet wurden. Annahmen, die aufgrund einer Erwerbsregel mit wenigen Prämissen und vielen Konsequenzen gebildet wurden (wie das etwa meistens bei Stereotypen der Fall ist) wird üblicherweise weniger Sicherheit zugesprochen als Annahmen, die auf einer Erwerbsregel mit hoher Prämissen/Konsequenz-Ratio beruhen.

### 3 Repräsentation, Schlußfolgerungen und Konsistenzerhaltung

Sobald Annahmen über den Benutzer gebildet wurden, können sie in einigen Anwendungsbereichen unmittelbar zu Adaptionzwecken verwendet und gleich wieder “vergessen” werden. Wenn es jedoch in einem Anwendungsbereich erwünscht ist, Annahmen für mögliche spätere Adaptionzwecke aufzuheben oder aus bereits gebildeten Annahmen weitere Annahmen zu inferieren, oder wenn Widersprüche zwischen den Annahmen über den Benutzer auftreten können, dann müssen sie in einem geeigneten Repräsentationssystem formal dargestellt werden.

In den meisten bisher entwickelten Systemen wurden nur relativ einfache Formalismen verwendet, wie etwa Attribut-Wert-Paare (Rich 1979ab), Frames (Schwab 1989), oder Untermengen von KL-ONE-ähnlichen Sprachen (Kobsa 1985, Paris 1989, Kass 1991). Der Grund hierfür ist, daß die repräsentierten Annahmen nur eine relativ einfache Struktur hatten (etwa die Form  $\langle \text{Benutzermerkmal} \rangle \langle \text{Merkmalsausprägung} \rangle$ ). In den letzten Jahren gibt es jedoch zunehmenden Bedarf danach, auch komplexere Annahmen über den Benutzer bilden und repräsentieren zu können. Dazu gehören etwa

- Annahmen über Regeln, die der Benutzer verwendet,
- Annahmen über den Benutzer, die universelle oder existentielle Quantifikation bzw. Disjunktion oder Negation beinhalten,
- mehr als eine Art von Annahmen über den Benutzer (etwa sein Wissen und seine Ziele),
- geschachtelte Annahmen (etwa Annahmen des Systems über Annahmen des Benutzers über Ziele des Systems) oder gemeinsame Überzeugungen (Mutual Beliefs), sowie
- mehrere Stereotype und insbesondere Stereotypenhierarchien.

Aus diesem Grund wurden in den letzten Jahren zunehmend komplexere Repräsentationsformen für Benutzermodelle verwendet. Dazu gehören etwa

- *PROLOG* (Finin 1989; Eydner & Vergara 1993), das neben den Repräsentations- und Inferenzmöglichkeiten auch noch einen gleitenden Übergang in eine Programmiersprache bietet;
- *Prädikatenlogik* (Appelt & Pollack 1992, Kobsa 1992, Fink & Herrmann, 1993), die mehr Ausdruckskraft als Prolog aufweist, was in vielen Domänen auch benötigt wird;
- Sprachen mit *Prädikaten höherer Ordnung* (van Arragon 1991), *Mögliche-Welten-Übersetzungen von Modallogik* (Bunt 1990, Pohl 1993) und *Partitionen*, die die Repräsentation von (geschachtelten) Annahmen über eventuell mehrere Akteure ermöglichen;
- *Konnektionistische Netzwerke* (Jennings & Higuchi 1993, Lindner & Bodendorf 1993), die etwa für Klassifikationsaufgaben eingesetzt werden, sowie *hybride Repräsentationssysteme* (Schwinn 1993).

Für Schlußfolgerungen auf Basis von Benutzermodellen werden übliche KI-Techniken verwendet, und zwar meistens Produktionsregeln (Kay 1990; Zukerman & McConachy 1993) oder Vererbungsmechanismen (Kass 1991), manchmal aber auch logikbasierte Methoden wie Klassifikation (Kobsa 1990a), Deduktion (van Arragon 1991) oder Abduktion (Appelt & Pollack 1992).

Bei den meisten Systemen sind die Schlußfolgerungen nichtmonotoner Natur, da bisherige Annahmen über den Benutzer beim Vorliegen konfliktärer neuer Annahmen eventuell zurückgezogen werden. In diesem Fall müssen auch alle Schlußfolgerungen, die aus den alten Annahmen gezogen wurden, revidiert werden, falls sie nicht von anderen Annahmen unterstützt werden. In der KI wurden für diese Aufgabe Truth-Maintenance-Systeme entwickelt, die die inferentiellen Abhängigkeiten zwischen Annahmen verwalten und beim Vorliegen von Konflikten zwischen Annahmen auf Basis von Prioritätsregeln festlegen, welche davon zurückgezogen werden sollen. Solche Systeme werden auch in der Benutzermodellierung in den letzten Jahren verstärkt eingesetzt (Huang et al. 1991, Brajnik & Tasso 1992, Eydner & Vergara 1993). Zu den Besonderheiten von Truth-Maintenance in diesem speziellen Anwendungsbereich gehört, daß die Prioritätsregeln für die Auswahl von zurückzusetzenden Annahmen stark die Herkunft dieser Annahmen berücksichtigen (Annahmen aus direkter Beobachtung werden üblicherweise Annahmen aus Inferenzen und diese wiederum Annahmen aus aktivierten Stereotypen vorgezogen). Eine weitere Besonderheit ist das (bisher noch ungelöste) Problem der Integration von Stereotypendeaktivierung in einen Truth-Maintenance-Algorithmus.

## **4 Shell-Systeme für Benutzermodellierungskomponenten**

Da die Entwicklung von Benutzermodellierungskomponenten in Anwendungssystemen sehr aufwendig ist, gibt es seit einigen Jahren Bestrebungen, Benutzermodellierungs-Shell-Systeme zu entwickeln (ähnlich wie etwa im Bereich Expertensysteme). Diese Shell-Systeme stellen eine Anzahl integrierter Mechanismen zur Verfügung, die in Benutzermodellierungskomponenten häufig benötigt werden. Ein Benutzermodellentwickler soll die Möglichkeit haben, ein für seine Anwendungsdomäne passendes Shell-System und daraus wiederum die für die Anwendung notwendigen Komponenten auszuwählen und mit dem Benutzermodellierungswissen des Anwendungsbereichs zu füllen. Das resultierende Laufzeitsystem würde dann alle oder zumindest zentrale Benutzermodellierungsaufgaben im Anwendungssystem erfüllen. Die Dienste der bisher entwickelten Werkzeugsysteme konzentrieren sich auf Repräsentation, Inferenz, Stereotypenverwaltung und Truth-Maintenance. Primärannahmen über den Benutzer müssen üblicherweise vom Anwendungssystem geliefert werden, da sie stark domänenabhängig sind.

Im folgenden soll die Funktionalität der bisher entwickelten umfangreicheren KI-orientierten Benutzermodellierungs-Shell-Systeme kurz beschrieben werden.

### **4.1 GUMS**

Das System GUMS (Finin 1989) akzeptiert Annahmen über den Benutzer als Eingabe des Anwendungssystems, speichert sie, prüft ihre Konsistenz mit bereits existierenden Annahmen (und zwar derart, daß es versucht, die Negation der neuen Annahme aus den bisherigen Annahmen herzuleiten), informiert die Anwendung über erkannte Widersprüche und beantwor-



tet Fragen in bezug auf die derzeitigen Annahmen über den Benutzer. Das System gestattet die Definition einer Stereotypenhierarchie, die streng baumartig sein muß. Auch kann jeweils nur ein einziges Stereotyp auf den Benutzer angewandt werden. Das Anfangsstereotyp muß vom Anwendungsprogramm bestimmt werden. Jedes Stereotyp enthält eine Anzahl von sicheren Fakten über den Benutzer. Gerät einer davon in Konflikt mit einer von der Anwendung gelieferten Annahme über den Benutzer, so wird das Stereotyp aufgegeben zugunsten des nächsthöheren Stereotyps, das den konfliktären Fakt nicht mehr enthält (Stereotype in Parallelzweigen werden als Alternative nicht in Betracht gezogen).

GUMS unterstützt zwei Arten von Inferenzregeln, nämlich “gesicherte” Inferenzregeln und Default-Inferenzregeln. Beide werden in Prolog repräsentiert und in Rückwärtsverkettung verarbeitet, wenn GUMS auf Anfrage der Anwendung die Herleitbarkeit einer Annahme aus den derzeitigen Annahmen über den Benutzer prüft und die Konsistenz von neu eintreffenden Annahmen mit den bisherigen Annahmen testet. Wenn das befragte Prädikat als geschlossen deklariert wurde (d.h. als *definiert* durch alle Assertionen, auf deren linken Seite es vorkommt), dann wendet GUMS dabei auch “negation as failure” an. Die Verwendung von Default-Regeln zwingt GUMS dazu, nach dem Finden einer Default-Lösung weiter nach einer Nicht-Default-Lösung zu suchen. Das Ergebnis einer Anfrage an GUMS wird im System nicht gespeichert. Deswegen besteht keine Notwendigkeit für Abhängigkeitsverwaltung und Truth Maintenance.

## 4.2 UMT

UMT (Brajnik & Tasso 1992) erlaubt die Definition von Stereotypen, die die Charakteristika von Benutzeruntergruppen in Form von Attribut-Wert-Paaren enthalten. Stereotype können in beliebigen Hierarchien geordnet sein, wobei Vererbung unterstützt wird. Jedes Stereotyp besitzt eine Aktivierungsbedingung, die festlegt, wann ein Stereotyp auf den gegenwärtigen Benutzer angewandt werden kann. UMT stellt auch einen Regelinterpretierer zur Verfügung, der die Definition von Inferenzregeln für Benutzermodelle gestattet. Mögliche Kontradiktionen zwischen angenommenen Benutzereigenschaften müssen ebenfalls explizit mit Hilfe solcher Regeln spezifiziert werden.

UMT akzeptiert und speichert Assertionen über den Benutzer, die vom Anwendungssystem gebildet werden. Diese Assertionen können nach dem Grad der Zuverlässigkeit als invariante *Prämissen* oder als später wieder zurückziehbare *Annahmen* betrachtet werden. Stereotype, deren Aktivierungsbedingungen durch die bisherigen Annahmen erfüllt werden, fügen weitere Annahmen hinzu (nämlich die Attribut-Wert-Paare, die die entsprechenden Benutzeruntergruppen charakterisieren). Einige dieser Annahmen können dabei untereinander widersprüchlich sein. UMT wendet nach jeder Veränderung des Benutzermodells alle Inferenzregeln (inklusive die Widerspruchsentdeckungsregeln) auf die Menge der Prämissen und Annahmen an und zeichnet auch die inferentiellen Abhängigkeiten auf.

Eine Truth-Maintenance-Komponente bestimmt dann alle möglichen Benutzermodelle, d.h. alle maximalen konsistenten Mengen von Assertionen, bestehend aus den Prämissen, einer Auswahl von Annahmen sowie allen Ableitungen daraus. Das “gegenwärtige Benutzermodell” wird durch Anwendung von Präferenzkriterien aus den möglichen Benutzermodellen selektiert. (Es wird dabei etwa den von der Anwendung gelieferten Annahmen ein höheres Gewicht beigemessen als den aus Stereotypen stammenden Annahmen.) Wenn später Inkonsistenzen mit neuen

Informationen aus der Anwendung entdeckt werden, können die Annahmen, auf denen die konfliktären Assertionen basieren, leicht entdeckt werden, da die Abhängigkeiten aufgezeichnet wurden. Die Menge der möglichen Benutzermodelle kann revidiert und re-evaluiert werden, um das neue gegenwärtige Benutzermodell zu finden.

### 4.3 PROTUM

PROTUM (Eydner & Vergara 1993) stellt eine Art von Vereinigung der Stärken von GUMS und UMT dar. Es basiert wie GUMS auf Prolog und besitzt wie UMT eine Abhängigkeitsverwaltung und ein Truth-Maintenance-System. Die Stereotypenhierarchie von PROTUM ist jedoch im Unterschied zu GUMS nicht auf eine Baumstruktur beschränkt und die vom TMS verwalteten Annahmen über den Benutzer nicht auf Attribut-Wert-Paare wie bei UMT. PROTUM berechnet für jedes Stereotyp den Grad der Erfülltheit seiner Aktivierungsvoraussetzungen und verwendet dieses Maß sowohl für die Aktivierung und Zurücksetzung von Stereotypen als auch für die Konfliktauflösung zwischen widersprüchlichen Annahmen aus zwei verschiedenen aktiven Stereotypen. Bei der Konfliktauflösung im Truth-Maintenance-System wird die Art und die Herkunft der konfliktären Annahmen berücksichtigt.

### 4.4 BGP-MS

Das in Entwicklung befindliche System BGP-MS (Kobsa 1990a, 1992, Höhle et al. 1993) enthält ein "partitioniertes" Benutzermodell, das die gleichzeitige Repräsentation von mehreren Typen von Annahmen über den Benutzer gestattet, insbesondere über sein Wissen, seine Ziele und seine Fähigkeiten. Annahmen können auch beliebig geschachtelt sein, d.h. sie können etwa Annahmen des Benutzers über Systemziele o.ä. umfassen. Großer Wert gelegt wird auf die Unterscheidung zwischen "privaten" Systemannahmen über den Benutzer und solchen Annahmen, die gemeinsam bekannt sind (Mutual Beliefs).

BGP-MS stellt beliebig hierarchisch geordnete Stereotype zur Verfügung, für die Aktivierungs- und Deaktivierungsregeln definiert werden können (dabei kann auf vordefinierte Regeln zurückgegriffen werden). Ein Stereotypenverwaltungssystem bestimmt zur Laufzeit die gegenwärtig am besten zutreffenden Stereotype (Präzision, Recall und Häufigkeit der Re-Evaluierung können dabei voreingestellt werden). Das Anwendungssystem kann BGP-MS Annahmen über den Benutzer in einer Sprache erster Ordnung mit Modaloperatoren mitteilen, die in effizientere interne Repräsentationen wie KN-PART (Fink & Herrmann 1993), SB-ONE (Kobsa 1991b) und Prädikatenlogik übersetzt wird. Das Deduktionssystem OTTER (McCune 1990) steht für prädikatenlogische Inferenzen zur Verfügung; eine Abhängigkeitsverwaltungs- und Truth-Maintenance-Komponente ist in Entwicklung. Graphische Schnittstellen erleichtern die Arbeit des Benutzermodellentwicklers.

Wie alle anderen Shell-Systeme verlangt auch BGP-MS, daß Primärannahmen über den Benutzer von der Anwendung gebildet werden, da sie sehr domänenabhängig sind. Zusätzlich offeriert BGP-MS aber auch eine Bibliothek von domänenunabhängigen Annahmen, die über den Benutzer getroffen werden können, wenn bestimmte Kommunikationsakte in der Interaktion mit ihm aufgetreten sind (Pohl et al. 1993). Das Anwendungssystem braucht dann BGP-MS nur mehr darüber zu informieren, welche Kommunikationsakte stattgefunden haben. Die daraus resultierenden Annahmen über den Benutzer werden von BGP-MS getroffen und in das Benut-

zermodell eingetragen. Wenn etwa das Anwendungssystem BGP-MS darüber informiert, daß es dem Benutzer mitgeteilt hat, daß es sein Ziel  $p$  ablehnt, dann wird BGP-MS die Annahme bilden, daß von nun an eine gemeinsame Überzeugung darüber besteht, daß der Benutzer  $p$  möchte, nicht aber das System. Diese Kommunikationsakte sind unabhängig vom Interaktionsmedium zwischen dem Benutzer und der Anwendung.

## 5 Verwendung und Evaluierung von Benutzermodellen in interaktiven Softwaresystemen

Während der Interaktion mit dem Benutzer kann das Anwendungssystem die gegenwärtigen Annahmen über den Benutzer bei der Benutzermodellierungskomponente laufend anfordern und zur Anpassung an den Benutzer verwenden. Das Anwendungssystem kann dabei

- diese Anpassung unmittelbar selbst durchführen (*automatische Adaption*), oder
- den Benutzer auf die Notwendigkeit einer Anpassung aufmerksam machen, ihm die Gründe dafür verständlich machen, und ihm die Entscheidung überlassen, ob die Anpassung auch tatsächlich durchgeführt werden soll (*computer-gestützte Adaption*, vgl. Dieterich et al. 1993).

Die zweite Alternative wird hauptsächlich dann in Frage kommen, wenn Anpassungen selten vorkommen, aber sehr bedeutsam sind. Dies wird beispielsweise wohl oft bei adaptiven Änderungen der Benutzerschnittstelle eines Anwendungssystems der Fall sein, da hier Benutzer eventuell umlernen oder auf neue Möglichkeiten aufmerksam gemacht werden müssen. Bei der Mehrzahl der oben aufgelisteten Anwendungsbereiche sind die Anpassungen jedoch extrem zahlreich (bei natürlichsprachlichen Systemen etwa muß bei der Generierung jedes einzelnen Pronomens oder Artikels das Benutzermodell konsultiert werden) und auch verhältnismäßig unauffällig (so daß dem Benutzer die Anpassung meistens wohl gar nicht auffällt). Aus diesem Grund wurden in den letzten 10 Jahren größtenteils Systeme mit automatischer Anpassung an den Benutzer entwickelt. Die beiden Alternativen schließen sich natürlich nicht aus, wie Oppermann (1992) und Fischer (1993) illustrieren.

Adaptivität und Benutzermodellierung können nicht Selbstzweck sein, sondern sollen die Benutzbarkeit eines Computersystems in bezug auf Effizienz, Verringerung der Fehlerrate, Verbesserung des Verständnisses und nicht zuletzt Benutzerzufriedenheit verbessern helfen. Während in den 80-er-Jahre kaum Wert auf eine entsprechende empirische Verifikation der Nützlichkeit der entwickelten adaptiven Systeme gelegt wurde, ist die Einsicht in die Notwendigkeit solcher Evaluierungen in den letzten Jahren deutlich gestiegen. Die folgenden Ergebnisse liegen derzeit vor:

- Kontext-sensitive Hilfe in einem WYSIWYG-Texteditor (Hirschmann 1990; Maurer et al. 1993; Krause et al., im Erscheinen): Es wurde gezeigt, daß die Anpassung von Hilfemenüs an den jeweiligen Benutzerfehler in einem WYSIWYG-Texteditor signifikant die Performanz verbesserte und die Redundanz und Fehlerrate signifikant verringerte.
- Navigationshilfe in einem Hypertext-System (Kaplan et al., 1993): Es wurde gezeigt, daß Navigationshilfe basierend auf Systemwissen über frühere Benutzer und die Ziele des derzeitigen Benutzers dessen Informationssuche signifikant beschleunigen kann.

- Adaptiver Hypertext (Boyle & Encarnacion, im Erscheinen): Es wurde gezeigt, daß die automatische Anpassung eines Hypertexts an den Wissensstand des Benutzers signifikant das Textverständnis und die Suchgeschwindigkeit verbesserte.
- Personalisierter News-Filter (Jennings & Higuchi, 1993): Es wurde gezeigt, daß die Präzision und der Recall eines Filters für elektronische News-Artikel, der Annahmen über die jeweiligen Benutzerinteressen bildet, von vielen Benutzern als zufriedenstellend empfunden wurde.

Diese und eine ganze Reihe anderer weniger stark empirisch abgesicherter Ergebnisse zeigen, daß Benutzermodellierung und automatische Adaption wahrscheinlich zumindestens in einigen Anwendungsbereichen nützlich sind. Die Ergebnisse sollten jedoch nicht übergeneralisiert werden, da "lohnende Adaption systemspezifisch ist; sie hängt von den Benutzern eines Systems ab und von den Anforderungen, die dieses System erfüllen soll" (Browne 1993, p. 69). Auch muß der Gewinn für den Benutzer in Relation gesetzt werden zum Zusatzaufwand, der für die Entwicklung eines adaptiven Systems mit Benutzermodellierung entsteht. Es sollten daher in Zukunft weitere adaptive Systeme mit Benutzermodellierungskomponenten in verschiedenen Anwendungsbereichen entwickelt, ihre Brauchbarkeit empirisch überprüft und damit die Breite möglicher Anwendungen für adaptive Systeme ausgelotet werden. Industrielle Anwendungen werden bereits konkret ins Auge gefaßt.

Nicht unerwähnt soll zum Schluß noch bleiben, daß es neben der Berücksichtigung individueller Benutzerunterschiede natürlich auch andere Gründe gibt, warum interaktive Dialogsysteme adaptive Komponenten enthalten sollten. Benutzer können etwa im Laufe Ihrer Arbeit mit einem System an Erfahrung gewinnen, oder ihre mit diesem System zu erfüllenden Aufgaben können sich ändern. Auch ist ein Benutzermodell in vielen Anwendungsdomänen nicht die einzige notwendige Wissensquelle für Adaptivität. Dialogmodelle, die den Inhalt und die Struktur der bisherigen Interaktion aufzeichnen, und Domänen- oder Aufgabenmodelle spielen ebenfalls eine wichtige (und vielfach noch wenig erforschte) Rolle. Benutzermodellierung und Adaptivität sind auch keine Allheilmittel, um ein benutzerunfreundliches System schlagartig besser zu machen. Benutzermodellierung ist zuletzt auch mit einer ganzen Reihe von Problemen in bezug auf Benutzerrechte an Benutzermodellen und den möglichen Mißbrauch von Benutzermodellierungstechniken verbunden (siehe Kobsa 1990b). Auf alle diese Punkte kann hier jedoch nicht weiter eingegangen werden.

## Literatur

- Allen, J. F. (1979): A Plan-Based Approach to Speech Act Recognition. TR 131/79, Dept. of Computer Science, University of Toronto, Canada.
- Allen, J. F. and L. K. Schubert (1993): The TRAINS Project. TRAINS Technical Note 91-1, Dept. of Computer Science, Univ. of Rochester, Rochester, NY.
- Appelt, D. E. and M. E. Pollack (1992): Weighted Abduction for Plan Ascription. *User Modeling and User-Adapted Interaction* 2, 1-25.
- Boyle, C. and A. O. Encarnacion (im Erscheinen): An Adaptive Hypertext Reading System. Submitted to *User Modeling and User-Adapted Interaction*.
- Brajnik, G. and C. Tasso (1992): A Flexible Tool for Developing User Modeling Applications with Nonmonotonic Reasoning Capabilities. *Proceedings of the Third International Workshop on User Modeling, Dagstuhl, Germany*, 42-63.
- Browne, D. (1993): Experiences from the AID Project. In: M. Schneider-Hufschmidt et al. (1993).
- Bunt, H. (1990): Modular Incremental Modelling of Belief and Intention. In: *Proc. of the Second International Workshop on User Modeling, Honolulu, HI*, 1-30.
- Carberry, S. (1989): Plan Recognition and Its Use in Understanding Dialog. In: Kobsa & Wahlster (1989).
- Chin, D. N. (1989): KNOME: Modeling what the User Knows in UC. In: Kobsa & Wahlster (1989).
- Chin, D. N. (1993): Acquiring User Models. In: *Artificial Intelligence Review* 7, Special Issue on User Modeling, in press.
- Dieterich, H., U. Malinowski, T. Kühme, M. Schneider-Hufschmidt (1993): State of the Art in Adaptive User Interfaces. In: M. Schneider-Hufschmidt et al. (1993).
- Eydner, G. und H. Vergara (1993): Die Benutzermodellierungshell PROTUM basierend auf PROLOG und KN-PART. In: Kobsa & Pohl (1993).
- Finin, T. W. (1989): A General User Modeling Shell. In: Kobsa & Wahlster (1989).
- Fink, J. and M. Herrmann (1993): KN-PART - Ein Verwaltungssystem zur Benutzermodellierung mit prädikatenlogischer Wissensrepräsentation. WIS Memo Nr. 5, AG Wissensbasierte Informationssysteme, Informationswissenschaft, Universität Konstanz.
- Fischer, G. (1993): Shared Knowledge in Cooperative Problem-Solving Systems: Integrating Adaptive and Adaptable Components. In: M. Schneider-Hufschmidt et al. (1993).
- Genesereth, M. R. (1982): The Role of Plans in Intelligent Teaching Systems. In: D. Sleeman and H. S. Brown, *Intelligent Tutoring Systems*. New York: Academic Press.
- Grunst, G., R. Oppermann und C. G. Thomas (1993): Benutzungmodellierung bei kontext-sensitiver Hilfe und adaptiver Systemgestaltung. In: Kobsa & Pohl (1993).
- Hirschmann, A. (1990): Das Hilfesystem MATHILDE. Dissertation, Philosophische Fakultät, Universität Regensburg.
- Höhle, J.-C., A. Kobsa and W. Pohl (1993): BGP-MS: Ein Werkzeugsystem für Benutzermodellierung (Projektbericht 1992-1993). Memo 6, AG Wissensbasierte Informationssysteme, Informationswissenschaft, Universität Konstanz.
- Huang, X., G. I. McCalla, J. E. Greer and E. Neufeld (1991): Revising Deductive Knowledge and Stereotypical Knowledge in a Student Model. *User Modeling and User-Adapted Interaction* 1, 87-115.
- Jennings, A. and H. Higuchi (1993): A User Model Neural Network for a Personal News Service. To appear in *User Modeling and User-Adapted Interaction* 3(1).
- Kaplan, C., J. Fenwick and J. Chen (1993): Adaptive Hypertext Navigation Based on User Goals and Context. To appear in *User Modeling and User-Adapted Interaction* 3(2).

- Kass, B. (1991): Building a User Model Implicitly from a Cooperative Advisory Dialog. *User Modeling and User-Adapted Interaction* 1, 203-258.
- Kass, B. and I. Stadnyk (1992): Using User Models to Improve Organizational Information. In: *Proceedings of the 3rd International Workshop on User Modeling*, Dagstuhl, Germany, 135-147.
- Kay, J. (1990): um: A Toolkit for User Modelling. In *Proc. of the Second International Workshop on User Modeling*, Honolulu, HI, 1-11.
- Kobsa, A. (1985): *Benutzermodellierung in Dialogsystemen*. Springer, Berlin, Heidelberg.
- Kobsa, A. and W. Wahlster, eds. (1989): *User Models in Dialog Systems*. Berlin: Springer.
- Kobsa, A. (1990a): Modeling the User's Conceptual Knowledge in BGP-MS, a User Modeling Shell System. *Computational Intelligence* 6, 193-208.
- Kobsa, A. (1990b): User Modeling in Dialog Systems: Potentials and Hazards. *AI & Society: The Journal of Human and Machine Intelligence* 4, 214-231.
- Kobsa, A., ed. (1991a): *User Modeling and User-Adapted Interaction: An International Journal*. Dordrecht: Kluwer Academic Publishers (published since 1991).
- Kobsa, A. (1991b): Utilizing Knowledge: The Components of the SB-ONE Knowledge Representation Workbench. In: J. Sowa, ed.: *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. San Mateo, CA: Morgan Kaufmann.
- Kobsa, A. (1992): Towards Inferences in BGP-MS: Combining Modal Logic and Partition Hierarchies for User Modeling (Preliminary Report). In: *Proceedings of the 3rd International Workshop on User Modeling*, Dagstuhl, Germany, 35-41.
- Kobsa, A. (1993): User Modeling: Recent Work, Prospects and Hazards. In: M. Schneider-Hufschmidt et al. (1993).
- Kobsa, A. und W. Pohl (1993): *Arbeitspapiere des Workshops 'Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen'*, Berlin, 13.-15. 9. 1993. WIS-Memo 7, AG Wissensbasierte Informationssysteme, Informationswissenschaft, Universität Konstanz.
- Kok, A. (1991): A Review and Synthesis of User Modeling in Intelligent Systems. *The Knowledge Engineering Review* 6, 21-47.
- Koller, F. (1993): A Demonstrator Based Investigation of Adaptability. In: Schneider-Hufschmidt et. al. (1993).
- Krause, J., A. Hirschmann and E. Mittermaier (im Erscheinen): The Intelligent Help System COMFOHELP: Towards a Solution of the Practicability Problem for User Modeling and Adaptive Systems. Submitted to *User Modeling and User-Adapted Interaction*.
- Lindner, H.-G. und F. Bodendorf (1993): Ein neuronales Konzept für adaptive Anwendungen. In: Kobsa & Pohl (1993).
- Maurer, H., E. Mittermaier und M. Schommler (1993): Das intelligente Hilfesystem ComfoHelp. In: Kobsa & Pohl (1993).
- McCune, W. W. (1990): OTTER 2.0 Users Guide. Technical Report ANL-90/9, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL.
- McTear, M., ed. (1993): *Artificial Intelligence Review*, special issue on user modeling (in press).
- Neal, L. R. (1989): The Role of User Models in System Design. Ph.D. Dissertation, TR-18-89, Harvard University, October 1989.
- Nwana, H. S. (1991): User Modelling and User Adapted Interaction in an Intelligent Tutoring System. *User Modeling and User-Adapted Interaction* 1, 1-32.
- Oppermann, R. (1991): Experiences with Evaluation Methods for Human-Computer Interaction. *Arbeitspapiere der GMD*, Nr. 540, St. Augustin, Germany.
- Oppermann, R. (1992): Adaptively Supported Adaptability. In: *Proceedings of the 6th European Conference on Cognitive Ergonomics*, Balatonfüred, Hungary, 255-268.
- Orwant, L. (1991): The Doppelganger User Modelling System. In *Proc. of the IJCAI Workshop W4: Agent Modelling for Intelligent Interaction*, pp. 164-168, Sydney, Australia.
- Paris, C. (1989): The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise. In: Kobsa & Wahlster (1989).

- Pohl, W. (1993): Viewübergreifendes Schließen in BGP-MS. In: Kobsa & Pohl (1993).
- Pohl, W., A. Kobsa und O. Kutter (1993): Benutzermodelleraufbau durch Präsuppositionsanalyse in interaktiven Softwaresystemen. In: Kobsa & Pohl (1993).
- Quilici, A. (1989): AQUA: A System that Detects and Responds to User Misconceptions. In: Kobsa & Wahlster (1989).
- Rich, E. (1979a): Building and Exploiting User Models. PhD Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.
- Rich, E. (1979b): User Modeling via Stereotypes. *Cognitive Science* 3, 329-354.
- Sanderson, D. P. and S. Treu (1993): Adaptive User Interface Design and Its Dependence on Structure. In: Schneider-Hufschmidt et al. (1993).
- Schneider-Hufschmidt, M., T. Kühme and U. Malinowski, eds. (1993): *Adaptive User Interfaces: Principles and Practise*. Amsterdam: North Holland Elsevier.
- Schwab, T. (1989): Methoden zur Dialog- und Benutzermodellierung in adaptiven Computersystemen. Dissertation, Institut für Informatik, Universität Stuttgart.
- Schwinn, J. (1993): Architecture for Hybrid User Modelling. In: Kobsa & Pohl (1993).
- Sleeman, D. (1985): UMFE: A User Modelling Front-End Subsystem. *Intl. J. Man-Machine Studies* 23, 71-88.
- Sukaviriya, P. and J. D. Foley (1993): A Built-in Provision for Collecting Individual Task Usage Information in UIDE: the User Interface Design Environment. In: Schneider-Hufschmidt et. al. (1993).
- Wahlster, W. and A. Kobsa (1989): User Models in Dialog Systems. In: Kobsa & Wahlster (1989).
- van Arragon, P. (1991): Modeling Default Reasoning Using Defaults. *User Modeling and User-Adapted Interaction* 1, 259-288.
- Zukerman, I. and R. McConachy (1993): Consulting a User Model to Address a User's Inferences during Content Planning. To appear in *User Modeling and User-Adapted Interaction*.