Die Rolle von mentalen und konzeptuellen Modellen in der Mensch-Computer-Interaktion: Konsequenzen für die Software-Ergonomie?

Norbert A. Streitz

Institut für Psychologie der RWTH Aachen

Zusammenfassung: Dieser Beitrag dient der Erstellung einer einheitlichen Terminologie für die bei der Untersuchung der Mensch-Computer-Interaktion verwendeten Modellkonzepte. Damit verbunden ist eine Klärung der in verschiedenen theoretischen Bezugssystemen auftretenden Begriffsbildungen. Auf der Grundlage der vorgeschlagenen Erweiterung der Überlegungen von Norman [7] und einer Klassifizierung von Modellen ist es möglich, die beim Designprozeß relevanten Modellkomponenten zu identifizieren. Abschließend werden einige Implikationen für die Verwendung von Metaphern beim Design von "benutzerorientierten" Systemen diskutiert.

1 Einleitung

Bei der Diskussion von Konzepten für die Software-Ergonomie tauchen in letzter Zeit vermehrt Begriffsbildungen auf wie "mentales Modell", "konzeptuelles Modell", "Benutzermodell", "Partnerbild", "Metapher" [1], [7]. Allerdings ist die Begriffsbildung nicht einheitlich und man kann schon bald von einem "mental model zoo" sprechen [10]. Diese Konzepte haben nicht nur theoretische Relevanz, sondern spielen auch eine Rolle bei der Suche nach Orientierungsmöglichkeiten für die Entwicklung benutzerorientierter Systeme. Eine Klärung dieser Begriffsvielfalt ist notwendig, da sie einerseits Auswirkungen auf die Vergleichbarkeit empirischer Untersuchungen hat, die mit diesen Begriffen operieren. Andererseits spiegelt die Begriffsvielfalt auch einen Teil der Verständigungsschwierigkeiten wieder, die bei der Zusammenarbeit von Psychologen, Arbeitswissenschaftlern und Systemdesignern leider oft noch zu beobachten sind.

2 Globale Modelle der Mensch-Computer-Interaktion (MCI)

Es erscheint sinnvoll, zunächst einmal zwischen "globalen" Modellen der MCI und "spezifischen" Modellen, die bei einzelnen Benutzern, Designern und Psychologen existieren, zu unterscheiden. Die globalen Modelle beziehen sich auf die Rolle, die dem Computer in der Mensch-Computer-Interaktion zugewiesen wird. Dies wird oft durch eine Metapher für die Verwendungs-

weise des Computers gekennzeichnet. So gibt es die Vorstellung, daß der Computer ein **Werkzeug** ist, der bei der Ausführung einer Arbeitstätigkeit als Arbeitsmittel eingesetzt wird [2].

Eine andere Vorstellung ist die, daß der Computer ein **Partner** ist, mit dem der Benutzer kommuniziert. Man spricht dann von Mensch-Computer-Kommunikation. Der Rechner wird dabei zum virtuellen Dialogpartner. Bei dieser Modellvorstellung werden die aus der Mensch-Mensch-Kommunikation geläufigen Begriffsbildungen und Befunde auf die neue Situation übertragen. Obwohl diese Situation als formale Kommunikation von der Mensch-Mensch-Kommunikation unterschieden wird, wird sie trotzdem oft symmetrisch modelliert [5], [6]. Dies erscheint uns für einige Konzepte, wie z.B. Intentionen oder auch vollständige Selbstbilder auf der Seite des Computers als - jedenfalls zur Zeit noch - nicht gegeben. Andererseits spielt für unsere Betrachtungen das Konzept des Partnerbilds oder auch Partnermodells eine wichtige Rolle. Wir kommen darauf zurück.

Eine dritte Vorstellung ist die Metapher, daß der Computer als Hardware- und Softwaregesamtheit ein **Medium** darstellt, das dem Austausch von Ideen und Anweisungen zwischen dem Benutzer und dem Programmierer dient.

Während sich diese Ansätze auf die Rolle des Computers beziehen, sollte darüber hinaus aber auch die gesamte MCI-Situation in Beziehung zur Arbeitstätigkeit des Benutzers gestellt werden. Wir fassen dazu die Tätigkeit (z.B. eines Sachbearbeiters) an einem rechnerunterstützten Arbeitsplatz als Problemlösen auf. Als Bezugsrahmen haben wir dafür ein Problemlösemodell der Mensch-Computer-Interaktion vorgeschlagen (Streitz [10]). Die Wahl eines Problemlösemodells der MCI eröffnet einen natürlichen Zugang sowohl zu den theoretischen Ansätzen als auch zu den empirischen Befunden experimenteller Forschung, die in der Kognitionspsychologie zur Verfügung stehen. Mit dem auch für die menschliche Informationsverarbeitung zentralen Konzept der Repräsentation von Wissen ist eine Ausgangsbasis für eine gemeinsame Sprache zur Beschreibung der an der Mensch-Computer-Interaktion beteiligten Komponenten gegeben.

Bei der Problemsituation der MCI ist grundsätzlich zwischen zwei Hauptproblemen zu unterscheiden:

- dem Sachproblem
- dem Interaktionsproblem.

Ohne diesen Punkt hier weiter auszuführen (siehe dazu [9], [10]), läßt sich diese Unterscheidung wie folgt motivieren. Einem Sachbearbeiter (z.B. einem Konstrukteur) wird ein Auftrag zur Entwicklung eines technischen Produktes

übertragen. Wir verstehen dies als die Vorgabe eines Sachproblems als der eigentlichen inhaltlichen Anforderung. Der Zielzustand besteht in den dem Auftragsvorhaben entsprechenden Ausführungsunterlagen für das herzustellende Produkt in Form von Zeichnungen, Stücklisten, Anweisungen. Diese Tätigkeit ist dem Konstrukteur prinzipiell zwar vertraut, bei neuen Fragestellungen aber als Problemlösen zu bezeichnen. Benutzt er dabei den Computer (z.B. ein CAD-System) als Hilfsmittel (als Werkzeug), muß er mit diesem in Interaktion treten. Das Bearbeiten des Sachproblems mit Hilfe des Computers erzeugt nun ein neues zusätzliches Problem: das Interaktionsproblem. Dabei handelt es sich um kein einheitliches Problem, sondern um die Summe mehrerer Teilprobleme. Der Zielzustand besteht hier in einer erfolgreichen und effizienten Interaktion mit dem Computersystem bei der Bearbeitung des Sachproblems. Das Verhältnis zwischen Sachproblem und Interaktionsproblem ist in Abb. 1 dargestellt.

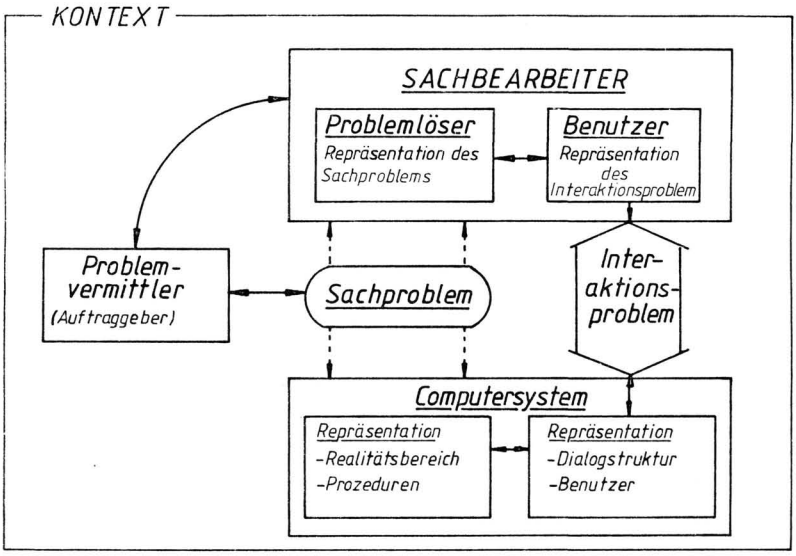


Abb. 1 Ein Problemlösemodell der Mensch-Computer-Interaktion

Eine zentrale Aussage dieses Modells ist, daß das Sachproblem nicht auf direktem Wege (--->) mit dem im Computer repräsentierten Wissen (über den Realitätsbereich und die vorhandenen Prozeduren) gelöst werden kann, sondern erst nach Überwindung des Interaktionsproblems. Dazu muß der Sachbearbeiter neben der Rolle des (Sach)Problemlösers auch die Rolle eines Benutzers übernehmen. Dies erfordert eine adäquate Repräsentation des das Interaktionsproblem betreffenden Komponenten des Systems. Den Repräsen-

tationen auf der Sachbearbeiterseite stehen korrespondierende Repräsentationen auf der Computerseite gegenüber.

Das Postulat einer benutzerfreundlichen oder auch benutzerorientierten Gestaltung der Mensch-Computer-Schnittstelle läßt sich hier als Minimierung des Interaktionsproblems (für den Benutzer) darstellen. Die konzeptuelle Trennung zwischen Sach- und Interaktionsproblem erscheint uns deshalb notwendig, weil bei der Beurteilung von interaktiven Computersystemen in vielen Fällen diese beiden Aspekte vermischt werden. Somit kann die Suche nach Ursachen von Unzufriedenheiten bei Benutzern und die dann erfolgende Modifikation des Systems sich auf die wesentlichen Aspekte konzentrieren. Es soll allerdings auch darauf hingewiesen werden, daß diese Trennung nicht absolut ist, d.h. es gibt in bestimmten Fällen auch Wechselwirkungen, die auf Abhängigkeiten der jeweiligen Repräsentationen basieren. So hat z.B. die Realisierung eines bestimmten Datenbankkonzeptes gewisse Rückwirkungen für mögliche Zugriffswege und damit auf das Interaktionsproblem.

3 Spezifische Modelle in der Mensch-Computer-Interaktion

Innerhalb des Problemlösemodells - aber auch in anderen globalen Modellvorstellungen - spielen die spezifischen Modelle eine entscheidende Rolle. Die spezifischen Modelle können als bei verschiedenen Personen vorhandene Wissensrepräsentationen aufgefaßt werden. Eine der bekanntesten Unterscheidungen wurde von Norman [7] vorgenommen. Er geht von einem "target system **t**" aus, unterscheidet dann zwischen einem konzeptuellen Modell **C(t)** dieses Systems und einem mentalen Modell **M(t)**, über das der Benutzer des Systems t verfügt. Weiterhin wird von einem **C(M(t))** als der Konzeptualisierung des mentalen Modells M(t) gesprochen. Schließlich führt er den Begriff des "system image" ein, als dem "Bild" (der Systemoberfläche), mit dem sich das System dem Benutzer darstellt. Dieses "system image" ist das Ergebnis des Designprozesses, der - nach Norman idealerweise - auf dem konzeptuellen Modell C(t) basieren soll, damit es konsistent und einsichtig ist. Bedienungsanleitungen sollen dem Benutzer C(t) vermitteln. Norman hofft dann, daß das mentale Modell M(t), das der Benutzer aufbaut, mit C(t) übereinstimmt. Dies kann allerdings nur unter der Voraussetzung gelten, daß das "system image" mit C(t) konsistent ist, was auch nicht immer der Fall ist.

Soweit die Darstellung bei Norman, an der wir einige Korrekturen und Erweiterungen anbringen wollen. Es wird aus den Ausführungen von Norman z.B. nicht klar, wessen C(t), bzw. C(M(t)) gemeint ist. Wir werden dazu im folgenden spezifizierende und klärende Aussagen machen. Auch sind Zweifel angebracht, ob C(t) die alleinige Grundlage des Designs und der Gestaltung der

Bedienungsanleitungen bilden soll. Young [13] verwendet für das mentale Modell M(t) den Begriff UCM (user's conceptual model). Wir werden aber in gewissem Sinne die Unterscheidung von Norman beibehalten und M(t) für das mentale Modell des Benutzers verwenden, während C(t) für konzeptuelle Modelle bei "Fachleuten" stehen sollen. Dabei gilt selbstverständlich, daß konzeptuelle Modelle wiederum die Grundlage der mentalen Modelle von Wissenschaftlern sind, wenn diese selbst die Rolle eines Benutzers einnehmen. Diese können dann auch unvollständig, instabil und inkonsistent sein, wie man es sonst nur von den mentalen Modellen naiver Benutzer annimmt. Andererseits kann man - wie Young - auch sagen, daß jedes mentale Modell eines Benutzers ein konzeptuelles Abbild des Systems darstellt.

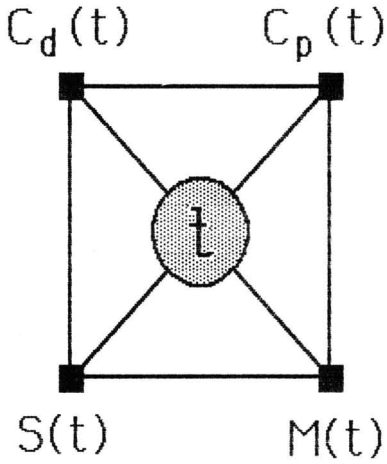
Die von uns vorgenommene Erweiterung führt zwar zu dem bereits beschworenen "mental model zoo", aber wir halten die Spezifizierung unterschiedlicher, für den Designprozeß und dessen Produkt wichtiger Perspektiven für sinnvoll und notwendig.

4 Das Perspektivenviereck und Modelle höherer Ordnung

Bei den (konzeptuellen) Modellen ist zu spezifizieren, wer diese Modelle und über wen oder was aufstellt. Hier ist unserer Meinung nach grundsätzlich zwischen zwei konzeptuellen Modellen zu unterscheiden: dem konzeptuellen Modell $C_d(M(t))$, das sich ein Designer über das mentale Modell des Benutzers macht, und dem Modell $C_p(M(t))$, das ein Psychologe über den Benutzer aufstellt. Bei adaptiven und ICAI-Systemen ist außerdem die Perspektive S(M(t)) zu berücksichtigen. Darunter ist ein Modell zu verstehen, das sich das System über den Benutzer aufbaut, bzw. genauer gesagt, ein Modell über das mentale Modell des Benutzers vom "target system".

In diesem Zusammenhang bedarf auch der Begriff des "target system" t von Norman einer weiteren Spezifikation. Dieses wird insbesondere deutlich, wenn man den formalen Ausdruck S(M(t)) betrachtet, in dem für das "System" zwei unterschiedliche Bezeichnungen (S und t) enthalten sind. Mit Bezug auf Anregungen aus Diskussionen mit Hoppe [4] ist zwischen dem allgemeinen Funktionsprinzip t und der Systemimplementation S(t) zu unterscheiden. Das Editieren von Texten oder das Arbeiten mit einem Kalkulationsprogramm basiert auf allgemeinen Funktionsprinzipien t und kann in abstrakter Weise (z.B. mit Algorithmen) beschrieben werden, während eine spezielle Softwarerealisierung auf einer konkreten Hardwarekonfiguration zur Systemimplementation S(t) führt. Diese Unterscheidung ist insbesondere für Lernprozesse und adaptive Systeme von Bedeutung.

Aus diesen Überlegungen leitet sich das in Abb. 2 dargestellte Perspektivenviereck über einer solchen Basisfunktion t ab. Damit ist die Behauptung verbunden, daß bei dem Benutzer mit M(t), dem Designer mit $C_d(t)$, dem Psychologen (oder Ergonomen) mit $C_p(t)$ voneinander verschiedene Modelle (Repräsentationen) der "target"-Funktion vorliegen. Hinzu kommt als vierte "Sichtweise" die durch das System realisierte Implementation S(t), die zwar auf $C_d(t)$ basiert, aber wegen der Berücksichtigung von Randbedingungen (z.B. technischer Art) nicht vollkommen mit $C_d(t)$ übereinstimmen wird. S(t) ist in gewissem Sinne mit dem "system image" zu identifizieren, wobei man hier eigentlich auch noch einmal zwischen der tatsächlichen (internen) Implementation und dem "äußeren" Erscheinungsbild dieser Realisierung unterscheiden müßte.



$C_d(t)$ = Sichtweise des Designers, $C_p(t)$ = Sichtweise des Psychologen
S(t) = Realisierung durch das System, M(t) = Mentales Modell (Benutzer)

Abb. 2 Perspektivenviereck mentaler und konzeptueller Modelle

Dabei ist anzumerken, daß die Unterschiede zwischen Psychologe und Designer auf dieser ersten Ebene, wenn auch vorhanden, noch nicht so bedeutungsvoll sind. Sie kommen aber auf den nächst höheren Modellebenen (siehe Abb.3) zum Tragen. Beziehen wir Abb. 2 auf die in der Kognitionspsychologie gemachte Annahme, daß ein Individuum beim Bearbeiten und Lösen eines Problems auf eine bereits vorhandene interne (mentale) Repräsentation des in Frage kommenden Realitätsbereichs zurückgreift, bzw. sich eine neue Ausgangsrepräsentation über t konstruieren muß, dann wird

die Bedeutung des Verhältnisses von S(t) und M(t) deutlich. Übertragen wir Ergebnisse von Untersuchungen zur Rolle von Ausgangsrepräsentationen für den Verlauf und den Erfolg des Problemlösens aus anderen Bereichen [8] [11], dann können wir die folgende These zur Benutzerorientiertheit als Forderung nach kognitiver Kompatibilität formulieren:

> Ein interaktives System ist umso benutzerorientierter, je geringer die Diskrepanzen zwischen den Repräsentationen S(t) und M(t) sind.

Hiermit stellt sich gleichzeitig die Frage, wie sich die Unterschiede zwischen den im System realisierten und den auf Seiten des Benutzers vorhandenen relevanten Wissensrepräsentationen erfassen lassen. Weiterhin muß das Ausmaß an "Diskrepanz" operationalisiert werden können.

An dieser Stelle ist außerdem darauf hinzuweisen, daß das Funktionsprinzip, i.e. das "target" t , selbst wieder zwei Komponenten aufweist, die sich aus dem Problemlösemodell der Mensch-Computer-Interaktion in Abb. 1 ergeben: dem Sach- oder Anwendungsaspekt (a) und dem Interaktionsaspekt (i). Wir kennzeichnen dies durch t = (a,i). Diese Differenzierung sei am Beispiel von M(t) kurz erläutert. Ein Sachbearbeiter hat eine bestimmte Vorstellung M(a) über das Funktionsprinzip (Vorgehensweise) und die Strukturierung, die der Bearbeitung seines Sachproblems, z.B. in Form eines Entscheidungsalgorithmus, zugrundeliegen. Andererseits hat er Vorstellungen über die notwendigen Schritte und Verfahren, wie er die für die Bearbeitung erforderlichen Daten sich beschaffen, miteinander verknüpfen und auf welche Berechnungsprozeduren er dabei zurückgreifen kann. Das weist darauf hin, daß es auch in den zuvor (ohne EDV) existierenden Arbeitssituationen Aspekte gibt, die man als Interaktionsproblem mit externen Quellen für die zur Bearbeitung notwendigen Informationen bezeichnen kann. In dem Fall, in dem vor der Einführung eines neuen Systems auch vorher schon mit einem (anderen) EDV-System gearbeitet wurde, ist das existierende M(i) dann durch die Erfahrung mit diesem System geprägt.

In Abb. 3 machen wir einen Vorschlag zur Klassifizierung möglicher Modelle. Durch die Anwendung von vier Operatoren auf unterschiedliche Operanden läßt sich die mögliche Vielfalt von Modellen in systematischer Weise erzeugen. Die dabei verwendeten Operatoren entsprechen den vier in Abb. 2 benannten Perspektiven. Auf einer ersten Ebene wird nur das Funktionsprinzip t als Operand in Betracht gezogen. Die Anwendung der Operatoren S, M, C_d, C_p führt dann zu den Modellen 1. Ordnung. Sie bilden die zu Beginn vorhandenen konzeptuellen (und mentalen) Repräsentationen des Funktionsprinzips t ab. Dabei nimmt S(t) eine Sonderstellung ein, da es eine implementierte Realisierung des Prinzips durch ein Computersystem darstellt.

Operator / Operand	S	M	C_p	C_d	Modelle
t	++	++	++	++	1. Ordnung
S(t)	*	++	+	+	2.Ordnung
M(t)	++	*	++	++	
$C_p(t)$	–	o	*	+	
$C_d(t)$	–	+	+	*	

++ = sehr relevant, + = relevant, o = nicht relevant,
* = "Selbstreflexion", – = existiert (noch) nicht

Abb. 3 Klassifizierung mentaler und konzeptueller Modelle in der MCI

Interessant ist nun der Übergang zu höheren Modellen, zunächst zu Modellen 2. Ordnung. Hier werden die vier Operatoren auf die Modelle 1.Ordnung als Operanden angewendet. Nicht alle daraus resultierenden 16 Kombinationen sind für unsere Fragestellung von gleicher Relevanz. So erhält man in der Hauptdiagonalen der entsprechenden 4 x 4 - Matrix die Modelle, die man als "Selbstreflexion" des jeweils eigenen Modells bezeichnen kann. Diese Modellkombinationen spielen auch eine nicht zu unterschätzende Rolle, sollen aber zunächst einmal ausgeklammert werden. Für die Software-Ergonomie erscheinen uns die folgenden Kombinationen als wichtig.

Die Forderung nach benutzerorientierten Computersystemen resultiert u. a. in der Notwendigkeit, die auf Seiten der Benutzer vorhandenen Modelle (subjektiven Repräsentationen), d.h. ihr Vorwissen, zu berücksichtigen. Dazu müssen diese Modelle erhoben und analysiert werden. Bei der Analyse mentaler Modelle M(t) werden von Designern und Psychologen unterschiedliche Methoden verwendet, die dann zu unterschiedlichen Vorstellungen über Benutzermodelle führen.

$C_p(M(t))$ = Modell, das der Psychologe über das mentale Modell des Benutzers erstellt (Gegenstand psychologischer Forschung)

$C_d(M(t))$ = Modell, das der Designer über das mentale Modell des Benutzers erstellt (der Designer versetzt sich in den "Benutzer")

So liefert $C_p(M(t))$ auf der Basis experimenteller Studien zwar eine detaillierte, unter kontrollierten Bedingungen erhobene und theoretisch einzuordnende Beschreibung von M(t). Diese ist aber leider nicht von vornherein auf die Bedürfnisse der Systemgestaltung abgestimmt und auch nicht unmittelbar durch den Systemdesigner umsetzbar. Andererseits basieren die von Systemdesignern erhobenen Vorstellungen der Benutzer über Funktionsprinzipien nicht selten alleine auf den eigenen Vorstellungen ("sich in die Rolle eines Benutzers versetzend") oder auf der Befragung von Kollegen. Diese auch als "Eindrucksforschung" [9] kritisierte Vorgehensweise ist aber für eine systematische Erhebung subjektiver Wissensrepräsentationen nicht geeignet, insbesondere nicht bei komplexen Sachverhalten. Unser Vorschlag geht dahin, beide Vorgehensweisen zu integrieren und so ein vollständiges und detailliertes, aber auch praktisch umsetzbares M(t) zu erhalten. Dies erfordert eine Abstimmung experimenteller Mindestanforderungen mit praktischen Bedürfnissen und daher interdisziplinär zusammengesetzte Arbeitsgruppen. Hier bietet sich dann auch Gelegenheit, die jeweiligen Modelle über die Vorstellungen des anderen, i.e. $C_p(C_d(t))$ und $C_d(C_p(t))$, zu modifizieren und anzugleichen.

Die Situation ist damit aber noch nicht vollständig beschrieben. Es muß nicht nur M(t) erhoben werden, sondern auch das sich im Verlaufe der Interaktion mit einem bestimmten System beim Benutzer aufbauende Modell M(S(t)). Dabei handelt es sich um dynamisch veränderliche subjektive Modelle, da im Verlaufe der Interaktion Lernprozesse zu berücksichtigen sind. In M(S(t)) fließen außerdem noch die Annahmen $M(C_d(t))$ ein, die der Benutzer über das konzeptuelle Modell $C_d(t)$, das dem Design wohl zugrundeliegt, aufstellt.

Will man auch M(S(t)) beobachten, so befinden wir uns auf der nächst höheren Ebene, den <u>Modellen 3.Ordnung</u>. Auf dieser Stufe gibt es 64 mögliche Kombinationen, die wir aber nicht alle behandeln. Auch hier ist für unsere Zwecke wieder nur eine Teilmenge relevant.

$C_p(M(S(t)))$ bzw. $C_d(M(S(t)))$ = Modell, das der Psychologe/ Designer über das Modell des Benutzers erhebt, das dieser über die Realisierung des Funktionsprinzips t durch das System S aufbaut.

Die beim Benutzer erhobenen mentalen Modelle M(t) und M(S(t)) werden dann vor dem Hintergrund der sowohl beim Designer als auch beim Psychologen vorhandenen Modelle analysiert. Beide verfügen über ihre konzeptuellen Modelle des Funktionsprinzips t hinaus natürlich auch über Repräsentationen der im System vorhandenen Realisierung S(t): $C_d(S(t))$ bzw. $C_p(S(t))$. Man sollte zwar meinen, daß $C_d(S(t))$ mit $C_d(t)$ übereinstimmen wird, aber das trifft nur in wenigen Fällen vollständig zu.

5 Berücksichtigung mentaler Modelle beim Design von Systemen

Nachdem man erhoben hat, wie Benutzer die relevanten Gegenstandsbereiche und damit verbundene Funktionsprinzipien repräsentieren - auf Techniken zur Erfassung subjektiver, mentaler Wissensrepräsentationen können wir in diesem Beitrag nicht eingehen - sollten diese bei der Gestaltung der Systemoberfläche Verwendung finden (--> These von der kognitiven Kompatibilität). Die Berücksichtigung von M(t), bzw. M(S(t)) beim Design steht im Gegensatz zu dem Ansatz, daß ein allein vom Designer entwickeltes $C_d(t)$ Grundlage des Designs S(t) sein sollte.

Mentale Modelle fungieren im Sinne der Handlungstheorie [3] in Form von "operativen Abbildsystemen" (OAS) als Basis zum Planen und Durchführen von Handlungen. Dazu gehört auch die Möglichkeit des internen Probehandelns und der Evaluation von Teilzielen. Bei der Berücksichtigung von mentalen Modellen in der Systemgestaltung sind im Prinzip zwei Wege zu beschreiten:

1) Im System ist ein bestimmtes S(t) realisiert, das auf einem konzeptuellen $C_d(t)$ basiert. Der Benutzer muß sich dieses $C_d(t)$ über das S(t) aneignen, bzw. er bekommt es über Handbücher und Bedienungsanleitungen vermittelt. Bei der Vermittlung von $C_d(t)$ wird auf M(t) nur Bezug genommen, indem man Ähnlichkeiten aufzeigt. Dabei können auch Metaphern verwendet werden (".. funktioniert wie eine Schreibmaschine", etc.)

2) Die Erhebung des M(t) erfolgt <u>vor</u> der Entwicklung von S(t) und dient dann neben dem vorher existierenden $C_d(t)$ mit als Leitlinie bei der Entwicklung. Damit diese Vorgehensweise aber auch dem Benutzer bewußt wird, vermittelt man das auf der Basis von M(t) entwickelte neue $C^*_d(t)$ über eine Systemoberfläche S(t), die an Konzepte von M(t) angelehnt ist. Dies geschieht dann über Metaphern wie z.B. dem Schreibtisch mit Ordnern, Dokumenten und Papierkorb (wobei letzterer eigentlich eher unter oder neben den Schreibtisch gehört).

Allerdings kommt man auch in diesen Fällen nicht ohne Bedienungsanleitungen aus, da die Verwendung von Metaphern auch Schwächen aufweist, wie wir noch zeigen werden. Zunächst sei auf folgende Punkte hingewiesen:

- Es sind Art und Ausmaß der Diskrepanzen zwischen M(t) und einem geplanten oder existierenden S(t) festzustellen, da diese sich dann in dem Modell M(S(t)) niederschlagen werden.
- Um solche Diskrepanzen zu vermeiden, sind die erhobenen Modelle $C_p(M(t))$ und $C_p(M(S(t)))$ entweder bei der Konzeption von S(t) bzw. eines entsprechend modifizierten S*(t) zu berücksichtigen.

- Schließlich sind adaptive Systeme einzubeziehen. Dazu muß ein S(M(t)) bzw. ein S(M(S(t))) aus der Interaktion des Benutzers mit dem System sukzessiv und für verschiedene Zeitpunkte erschlossen werden (Benutzermodelldiagnose). Hier gibt es eine Reihe von prinzipiellen und technischen Problemen der Realisierung. Das S(M(S(t))) sollte den jeweiligen Stand des Benutzers analysieren, sich ihm anpassen (Adaptivität) bzw. ihm Rückmeldung geben (aktives Help-System). Dazu ist es erforderlich, daß das S(M(S(t))) dynamisch konzipiert ist und - wenigstens in festzulegenden Zeitabschnitten - modifiziert wird. Wir unterscheiden dazu weiterhin zwischen Adaptivität und Adaptierbarkeit. Adaptive Systeme reagieren auf die Diagnose eines $M_{i+1}(S_i(t))$ als Veränderung auf ein zuvor bestehendes $M_i(S_i(t))$ selbstständig in der Form, daß sie ein $S_{i+1}(t)$ anbieten, das dem $M_{i+1}(S_i(t))$ besser entspricht. Adaptierbare Systeme dagegen können von dem Benutzer selbst ("the user as a designer") von einem $S_i(t)$ in ein $S_{i+1}(t)$ überführt werden. Adaptierbare Systeme sind weniger aufwendig, da sie keine Modelldiagnosekomponente enthalten müssen.

Die in der Matrix möglichen Modelle $S(C_d(t))$ und $S(C_p(t))$ sind bei z.Z. existierenden Systemen nicht anzutreffen und für uns, vor allem das $S(C_p(t))$, auch nicht interessant. Dies gilt nur, wenn man S hierbei als selbstständiges und aktives System auffaßt. Der Kombination $S(C_d(t))$ ließe sich aber eine neue Bedeutung zuordnen, wenn man darunter eine im System realisierte Erklärungskomponente für den Benutzer versteht, die das dem Design zugrundeliegende $C_d(t)$ näher erläutert. Dies könnte helfen, das entsprechende $M(C_d(t))$, das sich der Benutzer über den Designer macht, zu korrigieren.

Als eine Selbsterklärungskomponente ließe sich auch das S(S(t)) begreifen. Dazu muß aber die Notation in die Form S'(S(t)) überführt werden. Diese Modifikation ist im Prinzip bei allen Elementen der Hauptdiagonalen anzuwenden. Auf höheren Ebenen wäre dann z.B. unter M'(M(S(t))) die Selbstreflexion des Benutzers über sein mentales Modell der Systemrealisierung S(t) zu verstehen. An der Selbstreflexion sollte jedes Tutorial und Hilfemeldung anzusetzen, da nur der über die Selbstreflexion zugängliche Teil des M(S(t)) erreicht und verändert werden kann (z.B. auch durch Schulungen).

Probleme bei der Verwendung von Metaphern

Die Grundidee, Metaphern sowohl bei der Konzeption von Systemen als auch bei der Vermittlung von Wissen über existierende Systeme zu verwenden stellt sich aus psychologischer Sicht wie folgt dar. Es wird angenommen, daß der Benutzer über Vorwissen aus einem für das System relevanten Realitäts-

bereich verfügt. Ein neues System zu benutzen lernen bedeutet, Neuwissen mit dem vorhandenen Altwissen in Beziehung zu bringen. Die Wahl von Metaphern soll dazu dienen, daß das Neuwissen im wesentlichen auf die im Altwissen vorhandenen Strukuren Bezug nehmen kann. Entscheidendes Kriterium ist, daß das Neuwissen nicht oder nicht zu sehr mit dem Altwissen in Widerspruch steht. Dies ist aber nur selten zu erreichen. Woran liegt das?

Die dazu durchgeführte Analyse (siehe Streitz [12]) kann hier nicht im Detail wiedergegeben werden. Festzuhalten ist, daß zwei Arten von Analogien zu berücksichtigen sind: Analogien 1.Art, bei denen für dieselbe Grundstruktur zwei unterschiedliche "cover stories" (zwei Realitätsbereiche) konstruiert werden, und Analogien 2.Art, bei denen die Bezugnahme auf einen anderen Realitätsbereich auch eine Strukturänderung zur Folge hat. Obwohl man bei Metaphern gerne Analogien 1.Art vermitteln möchte, wirken sie auf der Basis des beim Benutzer vorhandenen Altwissens als Analogien 2.Art. So kommt es dazu, daß Relationen aus dem Metaphernbereich auf S(t) übertragen werden, die dort aber nicht zutreffen. Die Metapher bricht zusammen: sie "kollabiert." Dies ist und wird immer dann der Fall sein, wenn das Computersystem Möglichkeiten bereitstellt, die im gewählten Metaphernbereich nicht gelten.

5 Schlußfolgerungen

Wir haben zeigen können, daß eine überaus verzweigte Wechselwirkung zwischen den verschiedenen Modellen besteht, die bei den - am Designprozeß von benutzerorientierten Systemen - zu beteiligenden Partner vorhanden sind. Mit Hilfe einer geeigneten Klassifizierung war es möglich, alle mentalen und konzeptuellen Modelle, die bei der Mensch-Computer-Interaktion auftreten können, prinzipiell zu erfassen. Die in der Überschrift gestellte Frage muß also bejaht werden. Die Verwendung mentaler Modelle in der Software-Ergonomie hat eine Vielzahl von Konsequenzen und nicht immer erfreuliche.

Unsere Ausführungen zum Verhältnis von mentalen/konzeptuellen Modellen und Metaphern zeigen, daß sich Metaphern, die Analogien 2.Art sind, als zur Induktion mentaler Modelle M(S(t)) nicht geeignet erweisen. Dies gilt insbesondere dann, wenn es notwendig wird, neben der Metapher zusätzliche Modifikationsregeln zu vermitteln, die diese Diskrepanzen auffangen. Es bieten sich zwei Möglichkeiten an: Entweder ein genereller Verzicht auf Metaphern oder Metaphern mit einer zusätzlichen Vermittlung von "Abweichungsregeln". Entscheidet man sich nicht für die erste, die radikale Lösung, dann sollte eine Kosten-Nutzen-Analyse angestellt werden, inwieweit die Vorteile der Verwendung von Metaphern (z.B. in der Anfangssphase) die zusätzlichen Aufwendungen für das Erlernen und Befolgen der Abweichungsregeln überwiegen und somit den Einsatz von Metaphern rechtfertigen. In jedem Fall sollte aber angestrebt werden, daß die bei den Benutzern vorhandenen mentalen Modelle erhoben und berücksichtigt werden.

6 Literaturverzeichnis

[1] Caroll, J. : Mental models and software human factors. IBM Research Report RC 10616. Yorktown Heights, N.Y., 1984

[2] Dzida, W.: Dialogfähige Werkzeuge und und arbeitsgerechte Dialogformen. In H. Schauer & M. Tauber (Hrsg.), Informatik und Psychologie. (S. 54 - 86). Wien: Oldenburg, 1982.

[3] Hacker, W. : Allgemeine Arbeits- und Ingenieurpsychologie, Berlin: VEB der Wissenschaften, 1973.

[4] Hoppe, H.U. : Diskussionsbeitrag in der AG1 auf der 4.Tagung zur Mensch-Maschine-Kommunikation. Berlin : November 1984.

[5] Kupka, I., Maaß, S. & Oberquelle, H. : Kommunikation - ein Grundbegriff für die Informatik. Mitteilung Nr. 91 des Fachbereichs Informatik, Universität Hamburg: 1981.

[6] Maaß, S.: Mensch-Rechner-Kommunikation - Herkunft und Chancen eines neuen Paradigmas. (Dissertation) Bericht Nr. 104 des Fachbereichs Informatik, Universität Hamburg: 1984.

[7] Norman, D.A.: Some observations on mental models. In D. Gentner & A.L. Stevens (Eds.), Mental Models. (pp. 7 - 14). Hillsdale: Erlbaum, 1983.

[8] Streitz, N.A.: The importance of knowledge representation in problem solving: An example of text comprehension and problem solving. In G.Lüer (Hrsg.), Bericht über den 33. Kongreß der Deutschen Gesellschaft für Psychologie in Mainz. (S. 403 - 407). Göttingen: Hogrefe, 1983.

[9] Streitz, N.A. : Die Mensch-Maschine-Schnittstelle unter dem Blickwinkel der Kommunikation. Positionspapier zur 3. Arbeitstagung "Mensch-Maschine-Kommunikation", organisiert von der Gesellschaft für Mathematik und Datenverarbeitung (GMD). Bad Honnef:1983.

[10] Streitz, N.A.: Cognitive Ergonomics: An approach for the design of user-oriented interactive systems. In F. Klix (Ed.), MACINTER I (Man-Computer Interaction). Amsterdam: North-Holland, 1985. (in press)

[11] Streitz, N.A. Subjektive Wissensrepräsentationen als Determinanten kognitiver Prozesse. Unveröffentlichtes Manuskript (wird z.Zt. zur Veröffentlichung vorbereitet) Institut für Psychologie der RWTH Aachen: 1985.

[12] Streitz, N.A. Metaphors, analogies, and mental models: Implications for the design of adaptive user-system interfaces. Paper presented at the Symposium " Learning Issues for Intelligent Tutoring Systems ", organized by H. Mandl & A. Lesgold in Tübingen, June 1985.

[13] Young, R. : Surrogates and mappings: Two kinds of conceptual models for interactive devices. In In D. Gentner & A.L. Stevens (Eds.), Mental Models. (pp. 35 - 52). Hillsdale: Erlbaum, 1983.

Dr. Norbert A. Streitz, Institut für Psychologie der RWTH Aachen, Jägerstr. 17 - 19, D- 5100 Aachen.