

Aufgabenorientierte konzeptuelle Modellierung der Mensch-Computer Interaktion

Christian Stary, Wien

Zusammenfassung

Die konzeptuelle Modellierung von statischen Merkmalen sowie dynamischen Vorgängen an der Schnittstelle zwischen Mensch und Software ermöglicht die Berücksichtigung aufgabenorientierter Konzepte. Die vorgeschlagene Modellierungsart der Mensch-Computer Interaktion ist das Ergebnis interdisziplinärer Untersuchungen (Informatik, Kommunikationstheorie, kognitive Psychologie, Arbeitswissenschaften). Neben der Abbildung von Aufgaben auf Systemfunktionen ermöglicht das Repräsentationsschema die getrennte Darstellung von Interaktions- und Datenspezifika. Darüberhinaus finden didaktische Konzepte (Hilfe-, Lernstrategien, etc.) Eingang. Die Implementierung des Modells stellt ein Hilfsmittel für den Aufgabenanalytiker wie Systemdesigner dar.

1. Einleitung

Das Design aufgabenorientierter Benutzerschnittstellen stellt an Systementwickler höchste Anforderungen. Bedingt durch den Fortschritt der Hard- (Prozessorkapazitäten, Speichermedien, Interaktionshilfsmittel) und Software (Wissensverarbeitung, Modalitäten, Designhilfsmittel) gilt es, komplexe funktionale Schnittstellen mit möglichst frei gestaltbaren Benutzerumgebungen in Einklang zu bringen. Bestehende Interaktionsmodelle sind zeitaufwendig und umständlich handhabbar, da sie versuchen, Benutzerschnittstellen applikationsunabhängig zu vereinheitlichen (Hayes, 1985). Unterschiedliche Handlungsunterstützung für spezifische Applikationen bietet sich als Ausweg an, sobald vergleichbare Anstrengungen für das Design des Systemzuganges wie für Systemfunktionen aufgewandt werden (Stary, 1988b). Dies bedeutet, daß der Übergang von funktionalen Benutzerschnittstellen zu handlungsorientiertem Design umfassender Aufgaben- und Problemanalysen (anstelle einfacher Verhaltensanalysen) bedarf. Das Ziel unserer Forschungsarbeit war es daher, ein konzeptuelles Modell der Mensch-Computer Interaktion zu entwickeln, welches den Anforderungen komplexer Applikationen gerecht wird (Mehrfachdarstellung von Wissen, unterschiedliche Interaktionshilfsmittel, etc.).

Im Rahmen der konzeptuellen Modellierung (Brodie et al., 1984) wird eine Verständigungsebene auf hohem Abstraktionsniveau gewählt, sodaß die Entwicklung neuer Ansätze unter Berücksichtigung bereits vorgeschlagener Modelle möglich wird. Konzeptuelles Vorgehen beinhaltet das Hinterfragen von Sachverhalten (Analyse), die Darstellung (Beschreibung, Spezifikation) von Wissen (Modellierung) in einer Meta-Ebene unter Berücksichtigung unterschiedlicher Abstraktionsniveaus.

Trotzdem von einigen Designern die Notwendigkeit erkannt wurde, interdisziplinäres Wissen in konzeptuellen Modelle der Mensch-Computer Interaktion mitberücksichtigen zu müssen (z.B. Moran, 1981), ist eine Öffnung der Informatik, welche der Definition einer adäquaten Verständigungsebene bedarf, nicht gelungen. Die Anforderungen an aufgabenorientierte konzeptuelle Modelle der Mensch-Computer Interaktion erfordern folgende Vorgangsweise: Zur adäquaten Modellierung des Kontextes von Be-

nutzerschnittstellen ist es zunächst notwendig, neben Techniken der Informatik Wissen aus den Disziplinen Kommunikationswissenschaft, Arbeitsorganisation und kognitive Psychologie mitzuberücksichtigen. Der resultierende interdisziplinäre Ansatz muß formal darstellbar sein, um konsistentes, korrektes und implementierungsunabhängiges Schnittstellendesign zu ermöglichen.

2. Bisherige Modellierungsversuche

Die Zielsetzungen bisheriger Spezifikationen von Benutzerschnittstellen reichen von der Berücksichtigung von Interaktionsspezifika zu einem möglichst frühen Zeitpunkt der Systementwicklung über die Definition einer implementierungsunabhängigen Betrachtungs- und Darstellungsebene bis zur direkten Implementierbarkeit von Spezifikationen.

Abstraktionsmechanismen. Die Grundlage der meisten Modellierungsversuche bildet die Zerlegung der Interaktion in drei einander aufbauende Schichten (lexikalische, syntaktische, semantische Ebene) nach Morris, 1938, welche von Foley et al., 1982 in die Informatik eingebracht wurde. Der derzeitige Stand der Forschung läßt vornehmlich Modelle der syntaktischen Ebene erkennen. Darüberhinaus existieren holistische Herangehensweisen, welche das Schichtenmodell nach oben mit einer Aufgabenebene erweitern (Moran, 1981). Allerdings gelingt hier noch keine ausreichend formale Darstellung. Auch die Berücksichtigung kognitiver Prozesse (Reisner, 1984) bringt trotz formaler Darstellung keine bessere Anwendbarkeit. Schließlich gibt es Ansätze, welche sich mit der Abbildung von Elementen verschiedener Ebenen auseinandersetzen (z.B. Payne, 1984).

An verwendeten Paradigmen zur Modellierung der Mensch-Computer Interaktion innerhalb der Informatik können unterschieden werden: Algebren, Grammatiken, sowie Techniken des SW-Engineering.

Algebraische Spezifikationstechniken beschreiben Systemfunktionen in Form von Methoden und Axiomen (Chi, 1985). Aufgabenorientierte Konzepte müssen daher methodisch erfaßbar und auf Systemfunktionen abbildbar sein. Zahlreichen Vorteilen, wie konsistente Beschreibbarkeit oder direkte Implementierbarkeit, stehen Nachteile wie Reduktion auf statische Aspekte, umständliche Handhabung sowie ungenügende Prägnanz bei großer Funktionalität entgegen.

Grammatiken erfassen vor allem statische Aspekte, d.h. die Menge aller möglichen Dialoge eines Benutzers mit einer Applikation. Erweiterungen der herkömmlichen Darstellungsform (BNF) führen zur expliziten Berücksichtigung von Kommunikationspartnern (Shneiderman, 1982) sowie kognitiven Prozessen (Reisner, 1984). Aufgabenorientierte Spezifikation erlauben sowohl CLG (Moran, 1981) als auch die weniger restriktive TAG (Payne, 1984) sowie ETAG (Tauber, 1987). Zur Modellierung von Lern- oder Hilfestrategien wurden Grammatiken jedoch kaum eingesetzt.

Um die Dynamik von Interaktionsprozessen geeignet zu erfassen, wurden Zustand-Übergangsdiagramme eingeführt (Parnas, 1969). Während Kieras et al., 1985 Zustand-Übergangsdiagramme zur komplexen Aufgabenanalyse benutzen, bringt Wasserman, 1985 programmiertechnische Konzepte ein, um mittels Beschreibungen Prototypen von Benutzerschnittstellen zu implementieren. Diese Vorgehensweise wird allerdings durch zwei Nachteile getrübt: erschwerte Modellierung modusfreier Schnitt-

stellen sowie die Möglichkeit, unsinnige Kommandosprachen zu definieren. Weitere Ansätze aus dem Bereich des **SW-Engineering** betonen semantische (Studer, 1984) oder arbeitsorganisatorische Aspekte (Oberquelle, 1987). Durch die teilweise Anlehnung an programmiertechnische Konzepte werden Designkorrekturen sowie die Modellierung von Fehlersituationen erleichtert. Die ereignisorientierte Modellierung (Hill, 1987) basiert auf speziellen Automaten, mit deren Hilfe gleichzeitig eintretende Ereignisse modelliert werden können. Eine entsprechende Spezifikationssprache erlaubt die Definition exekutierbarer Schnittstellen. Ereignisorientierte Systeme beweisen zur syntaktischen Spezifikation größere deskriptive Mächtigkeit als formale Grammatiken und Zustand-Übergangsdiagramme (Greene, 1986).

3. Relevantes Wissen aus anderen Disziplinen

Die Theorie zwischenmenschlicher Beziehungen (**Kommunikationstheorie**) weist darauf hin, daß erfolgreicher Kommunikation gegenseitiges Verständnis zugrundeliegt (Satir, 1972). Bedingt durch die Erkenntnis inhaltsbezogener und beziehungsrelevanter Aspekte von Kommunikation wird eine pragmatische Betrachtungsebene notwendig (Watzlawick et al., 1967). Sie ermöglicht kontextspezifischen Bezug von Handlung und Kommunikation. Systemtheoretische Untersuchungen von statischen Elementen sowie dynamischen Prozessen der Kommunikation zeigen jedoch, daß verhaltensorientierte Modellbildungen zwischenmenschlicher Kommunikation nicht auf die Mensch-Computer Interaktion übertragbar, sondern günstigstenfalls Prinzipien übernehmbar sind (Stary, 1988b).

Die wissenschaftliche Auseinandersetzung mit **Arbeit** führte aufgrund des zur Zeit stattfindenden Umbruchs (Taylorismus -> integrierter Arbeitsplatz, System- -> Werkzeugperspektive, Maschinen- -> Menschzentrierung) noch nicht zu konzeptuellen Entwicklungen, welche eine differenzierte Betrachtungsweise von Hilfsmittel, wie der Computer eines darstellt, zulassen. Aus der Sicht der Arbeitswissenschaft wird das Arbeitshilfsmittel bzw. Medium Computer nicht speziell modelliert, sondern unter Unterstützung durch Technik eingereiht (z.B. Bierter, 1988).

Zweifelsohne bietet aufgabenorientierte **kognitive Psychologie** die für die Informatik attraktivsten Ansätze (Guedj et al., 1980). Sie entsprechen naturwissenschaftlichen Denkweisen (z.B. hierarchische Gliederung) und bestimmen informatische Modellbildungen. Zusammenfassend konnten wir einige Überlegungen und bereits angewandte Prinzipien aus den analysierten Disziplinen für eine holistische Modellierung der Mensch-Computer Interaktion übernehmen:

(1) Die Trennung zwischen "Knowing" und "Doing" (kognitive Psychologie) erscheint (trotz holistischem Ansatz) sinnvoll, als mit Sicherheit Planungsprozesse im Menschen ablaufen, welche evaluiert werden, ehe sie zu Aktivitäten führen (Hayes, 1980). Die Auffassung innerhalb der Arbeitswissenschaften, Arbeit als zweiseitige Interaktion (Mensch <-> Arbeitsmittel, Mensch <-> Arbeitsgegenstand) zu betrachten (Friedrich et al., 1987), weist daraufhin, daß der Arbeitsgegenstand getrennt von den Arbeitsmitteln zu modellieren ist. Bezogen auf die Mensch-Computer Interaktion bedeutet dies, daß jener Bereich der Aufgabenerfüllung, welcher sich auf die Daten des Problembereichs bezieht, von jenem Bereich, welcher sich auf den Umgang und die Art der Manipulation der Daten bezieht, getrennt zu betrachten ist.

(2) Handlungen, welche im Zuge der Aufgabenerfüllung gesetzt werden, sind im Menschen durch innere Modelle repräsentiert (Semmer et al., 1982). Spezifische Handlungsunterstützung kann daher nicht ausschließlich darauf gerichtet sein, dem Benutzer ein Modell der Applikation, welche nur einen Teil des gesamten Arbeitssystems darstellt, zu vermitteln. Vielmehr muß das Modell, welches der Mensch von dem technischen Hilfsmittel aufbaut, mit dem inneren Modell der Aufgabenerfüllung abgeglichen werden.

(3) Mensch-Computer Interaktion kann nicht ausschließlich über Funktionen abgewickelt werden. Der pragmatische Aspekt, d.h. die Auswirkungen von problemspezifischer Kommunikation auf Handlungsweisen müssen berücksichtigt werden. Dies führt zu applikationsspezifischer Modellierung.

4. Aufgabenorientiertes konzeptuelles Modellieren

Basierend auf der Idee semantischer Netze wurde ein konzeptuelles Modell der Mensch-Computer Interaktion entwickelt (Stary, 1988b), welches sich in drei Ebenen gliedert und zwischen Aktivitäten, welche den Problembereich einer Applikation betreffen, und Aktivitäten, welche dem Umgang mit der Applikation zuordenbar sind, unterscheidet. Damit wird es möglich, aufgabenorientierte Unterstützung, Lernstrategien und interaktive Hilfen gemäß ihrem Kontext und Grad an Komplexität **integriert** darzustellen.

Ähnlich der Schemata, welche im Bereich der Datenmodellierung zur Anwendung gelangen, hat sich die Definition von unterschiedlichen Ebenen im Rahmen der Modellierung der Mensch-Computer Interaktion aus folgenden Gründen als zielführend erwiesen:

(1) Aus der unmittelbaren Situation des Benutzers (Wissensstand, Sitzungsverlauf) gewinnt eine Aktion ihre Bedeutung, unabhängig davon, ob diese Aktion von der Applikation oder seitens des Benutzers initiiert wird. Beispielsweise kann eine Erklärungskomponente sowohl im Rahmen eines beabsichtigten Lernvorgangs oder als Folge eines mißlungenen Dialogschrittes aktiviert werden (-> getrennte Darstellung von Benutzeraktivitäten und Systemfunktionen).

(2) Nachdem seitens der Applikation die Bedeutung der Aktivität abgeklärt wurde, müssen sowohl Operationen auf Daten des Problembereichs als auch dialogspezifische Aktionen herausgefunden werden. Problemspezifische Aktionen sind Manipulationen von Datensätzen (Einfügen, Löschen sowie Ändern). Hingegen ist die Benutzung eines bestimmten Interaktionsmodus dialogspezifisch (-> Analyse dargestellter Benutzeraktivitäten).

(3) Um effizient Operationen durchführen zu können, ist es notwendig, eine weitere, den Benutzeraktivitäten untergeordnete Ebene einzuführen, welche 'maschinennahe' Funktionen (d.s. elementare Aktionen, welche seitens der Applikation unmittelbar ausgeführt werden können) enthält.

Das entwickelte konzeptuelle Modell ("Interaction Management Network") geht folglich von globalen aufgabenorientierten Aktivitäten des Benutzers aus, welche sowohl problem- als auch interaktionsspezifisch analysiert werden. Aktionen auf Daten des Problembereichs werden zu Systemfunktionen transformiert, während interaktionsspezifische Aktionen auf Interaktionsprozeduren abgebildet werden (Abbildung 1). Interaktionsbezogene Aktivitäten werden von gängiger Software kaum unterstützt, obwohl sie von unterschiedlichen Rollen des Benutzers bestimmt werden. Da mit ihrer Hilfe Lern-, Trai-

nings-, und Hilfekonzepte sowie die unmittelbare Benutzung einer Applikation modelliert werden können, sind pädagogische sowie arbeitsorganisatorische Steuerungsmechanismen vonnöten.

Wissen wird in Form von konzeptuellen Einheiten, deren Eigenschaften und Beziehungen zueinander repräsentiert. Nach der Abbildung auf Knoten und Kanten eines Netzes gelingt es mittels eines Klassenkonzeptes, unterschiedliche Sichtweisen zu definieren (z.B. Datenbeschreibungen, Lernstrategien, Hilfekonzepte, etc.). Klassen identifizieren Mengen von konzeptuellen Einheiten. Sie unterstützen den Prozess der Dekomposition sowie die Bildung von Abstraktionsebenen. Werden darüberhinaus Transformationsregeln entlang von Kanten erlaubt, sind dynamische Prozesse modellierbar.

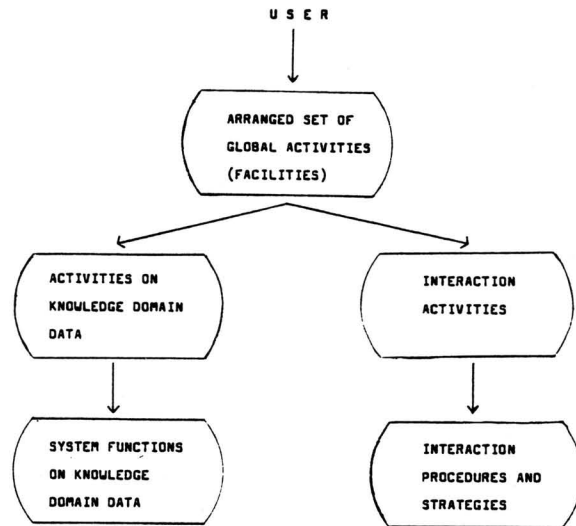


Abbildung 1

Die Ebenen des Repräsentationsschemas

Graphentheoretisch stellt das "Interaction Management Network" einen gerichteten Graph dar. Alle Knoten gehören einer von drei Ebenen an (Abbildung 1), wobei ihre Bedeutung durch die hierarchische Position sowie durch die, sie verbindenden Kanten festgelegt wird (Abbildung 2). Die oberste Ebene ist dem Benutzer einer Applikation zugewandt (sie stellt den Aktionsradius eines Benutzers dar), während die unterste Ebene funktionalen Schnittstellen (z.B. Prozeduraufrufen) entspricht. Somit können mittels des Netzes auch bestehende Applikationen um aufgabenorientierte Konzepte erweitert werden.

Das in Abbildung 3 gezeigte Beispiel zeigt den Ausschnitt eines konzeptuellen Modells zur Bewältigung des Berichtwesens innerhalb einer Organisation unter Zugrundelegung integrierter Software (Tabellenkalkulation, Textverarbeitung, Datenbank, Graphik). Das Erlernen computerunterstützter Erstellung von Arbeitsberichten ist benutzerspezifisch und wird durch die globale Aktivität "Learn Report Generation" dargestellt. Diese basiert auf der bereits festgelegten Aktivität 'Bearbeitung integrierter Dokumente' ("Handle Compound Documents") sowie auf der spezifizierten Lernstrategie ("Tutoring Integrated

Abbildung 2

IMN = (G_1, G_2, G_3),
 G_i gerichtete Graphen, $i \in \{1, 2, 3\}$

$G_1 = (V(G_1), E(G_1), \sigma_{G_1})$, $i \in \{1, 2, 3\}$
 $V(G_1)$ Menge von Knoten
 $E(G_1)$ Menge von gerichteten Kanten
 σ_{G_1} Inzidenzfunktion, welche jeder gerichteten Kante ein geordnetes Paar von Knoten zuweist
 $\sigma_{G_1}(e) = (x, y)$, $e \in E(G_1)$,
 $x, y \in V(G_1)$, $(x, y) \in V(G_1) \times V(G_1)$
 $NF(G_1)$ Menge aller direkten Nachfolger in G_1
 $VG(G_1)$ Menge aller direkten Vorgänger aus G_1
 $d^+(x)$ Menge der wegführenden Kanten, $x \in V(G_1)$
 $d^-(x)$ Menge der hinführenden Kanten, $x \in V(G_1)$
 $d(x) = d^+(x) + d^-(x)$ Grad, $x \in V(G_1)$

$G_1 = (V(G_1), E(G_1), \sigma_{G_1})$

$$| V(G_1) | \geq 1$$

$$\forall x \in V(G_1) : ((| NF(G_1) | = 0 \rightarrow | NF(G_2) | \geq 1) \vee (| NF(G_1) | \geq 1 \rightarrow | NF(G_2) | \geq 0)) \wedge (| NF(G_3) | = 0 \wedge | VG(G_1) | \geq 0 \wedge | VG(G_2) | \geq 0 \wedge | VG(G_3) | = 0)$$

$$d^+(x) \geq 1, d^-(x) \geq 0 \rightarrow d(x) \geq 1$$

$$| E(G_1) | \geq 1$$

$$\sigma_{G_1}(e) = (x, y), e \in E(G_1), \\ x, y \in V(G_1), ((x, y) \in V(G_1) \times V(G_1)) \vee ((x, y) \in V(G_1) \times V(G_2)) \vee ((x, y) \in V(G_2) \times V(G_1))$$

$$\text{Pfad} \left\{ \begin{array}{c} \underbrace{x_0 e_1 x_1 e_2 \dots e_j x_j \dots e_n x_n}_{\in G_1} \\ \underbrace{y_0 f_1 y_1 \dots f_j y_j \dots f_n y_n}_{\in G_2} \end{array} \right.$$

$G_2 = (V(G_2), E(G_2), \sigma_{G_2})$

$$| V(G_2) | \geq 1$$

$$\forall x \in V(G_2) : ((| NF(G_2) | = 0 \rightarrow | NF(G_3) | \geq 1) \vee (| NF(G_2) | \geq 1 \rightarrow | NF(G_3) | \geq 0)) \wedge (| NF(G_1) | \geq 0 \wedge | VG(G_3) | = 0 \wedge (| VG(G_1) | \geq 1 \rightarrow | VG(G_2) | \geq 0) \vee (| VG(G_1) | \geq 0 \rightarrow | VG(G_2) | \geq 1))$$

$$d^+(x) \geq 1, d^-(x) \geq 1 \rightarrow d(x) \geq 2$$

$$| E(G_2) | \geq 2$$

$$\sigma_{G_2}(e) = (x, y), e \in E(G_2), \\ x, y \in V(G_2), ((x, y) \in V(G_1) \times V(G_2)) \vee ((x, y) \in V(G_2) \times V(G_1)) \vee ((x, y) \in V(G_2) \times V(G_2)) \vee ((x, y) \in V(G_2) \times V(G_3))$$

$G_3 = (V(G_3), E(G_3), \sigma_{G_3})$

$$| V(G_3) | \geq 1$$

$$\forall x \in V(G_3) : | NF(G_1) | = 0 \wedge | NF(G_2) | = 0 \wedge | NF(G_3) | = 0 \wedge | VG(G_1) | = 0 \wedge | VG(G_2) | \geq 1 \wedge | VG(G_3) | = 0$$

$$d^+(x) = 0, d^-(x) \geq 1 \rightarrow d(x) \geq 1$$

$$| E(G_3) | \geq 1$$

$$\sigma_{G_3}(e) = (x, y), e \in E(G_3), \\ x, y \in V(G_3), ((x, y) \in V(G_2) \times V(G_3))$$

$$\text{Strategie: } \underbrace{x_0 e_1 x_1 e_2 \dots e_j x_j \dots e_n x_n}_{\in G_1} \underbrace{\dots}_{\in G_2} \underbrace{\dots}_{\in G_3}$$

$$x_j, y_j \in V(G_1), e_j, f_j \in E(G_1), i \in \{1, 2, 3\}, j = 1..n$$

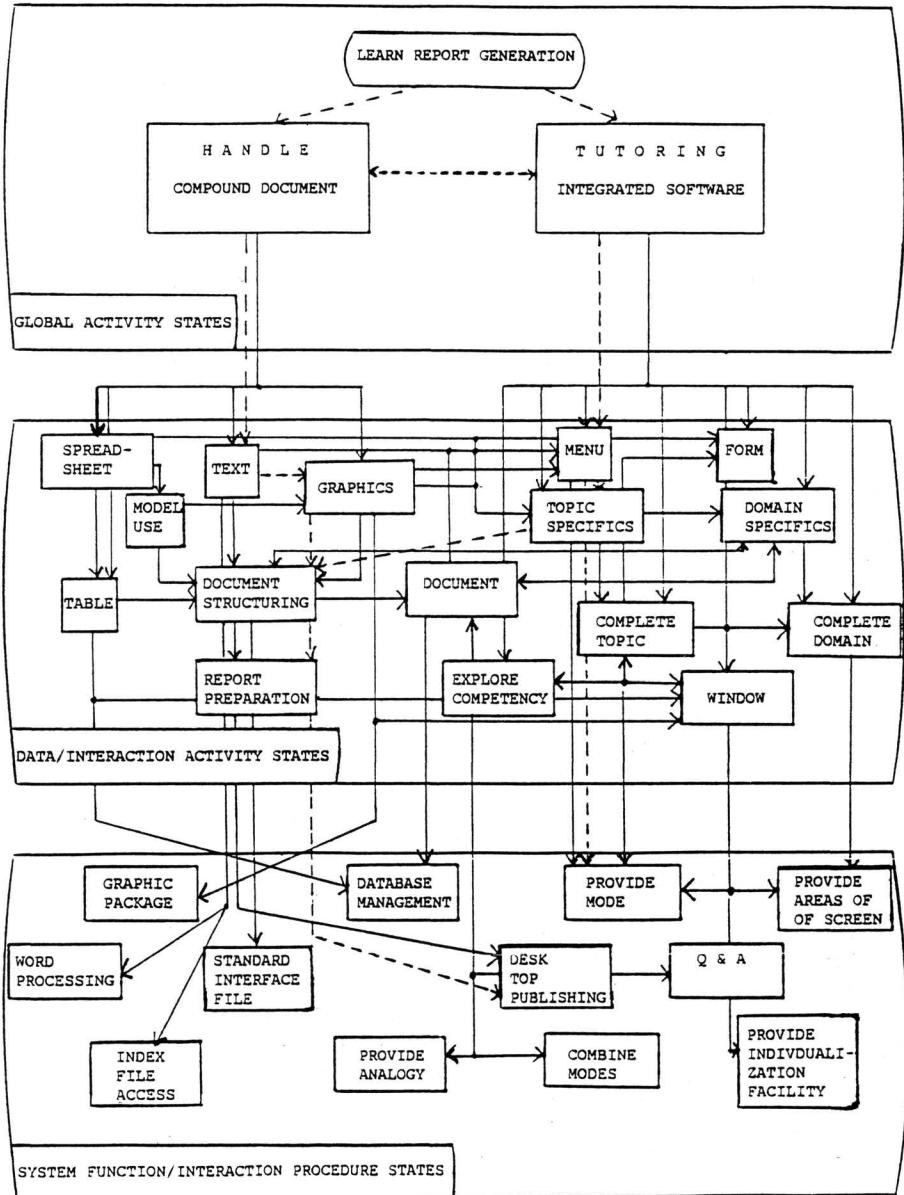


Abbildung 3

· Beispielhafter Ausschnitt einer konzeptuell modellierten Aufgabe

Software"). Der Pfad '-->' bezeichnet die benutzerspezifische Vorgangsweise sowie die Auswahl lern-strategischer Ziele.

Die benutzerspezifische Vorgangsweise besteht in diesem Fall im Erlernen der Textverarbeitung sowie des Graphikpakets, ehe das Strukturieren von Dokumenten erlernt wird. Danach wird auf das Desktop-Publishing System eingegangen, um Berichte zu verfassen. Die individuelle Lernstrategie besteht also im Begreifen ausgewählter Schwerpunkte und Modi, ehe die Gesamtaufgabe bewältigt wird.

An Klassen können in Abbildung 3 unterschieden werden (Ausprägungen sind in Klammern nach den Klassennamen angeführt):

Work Space ('spread sheet', 'text', 'graphics', 'table', 'document')
Activities: Generic ('learn report generation', 'handle', 'tutoring')
 Intermediate ('document structuring', 'report preparation')
Tutoring ('topic specific', 'domain specific', 'complete topic', 'complete domain')
Help ('explore competency')
Mode ('menu', 'form', 'window')
System Functions ('graphic package', 'word processing', 'index file access',
 'data base management', 'standard interface file access', 'desk top publishing')
Interaction Procedures ('provide mode', 'provide areas of screen', 'Q&A',
 'combine modes', 'provide individualization facilities')

Die Systemfunktionen werden in Abbildung 3 funktional zusammengefaßt. So birgt beispielsweise der Begriff Datenbankmanagementsystem eine Fülle von Operationen zur Eingabe und Manipulation von Daten in sich. Zur Darstellung der Modellierungsintention scheint allerdings eine derartige Subsum-mierung angebracht.

5. Ausblick

Das Ziel, für den Designprozeß von Benutzerschnittstellen, welcher bisher mit analytischen Techniken nicht in ausreichendem Maß unterstützt worden war, ein adäquates Hilfsmittel zu definieren, wurde mittels einer grundlegenden Änderung des Forschungsansatzes innerhalb der Informatik erreicht. Es führte zur Entwicklung einer interdisziplinären Verständigungsebene der Disziplinen Informatik, Kommunikationstheorie, Arbeitswissenschaft und kognitive Psychologie. Es war klar, daß höhere Anforderungen wie die konzeptuelle Modellierung der Zeit, individuelle Bewertung und Wissensbildung nur bedingt in einem ersten Schritt interdisziplinärer Modellierung erfüllt werden können.

Das vorgeschlagene Modell eröffnet die Möglichkeit, komplexe Aspekte der Mensch-Computer Interaktion zu betrachten, ohne Bezug auf die tatsächliche Implementierung nehmen zu müssen. Die in jeder Ebene zugelassene Netzstruktur erlaubt die Dekomposition komplexer Zusammenhänge, wobei unterschiedliche Gesichtspunkte, wie Aufgaben, Lernmodelle (z.B. Woolf et al., 1984) und umfassende Hilfekonzepte (Stary, 1988a) berücksichtigt werden können. Darüberhinaus garantiert die algebraische Spezifikation des "Interaction Management Network" konsistentes und korrektes Schnittstellendesign (Stary, 1988b). Die zusätzlich entwickelte graphische Notation (Abbildung 3) erhöhen die Anwendbarkeit des konzeptuellen Modells für Designer und potentielle Benutzer. Folgende Punkte bedürfen allerdings weiterer Untersuchung:

(1) Das IMN stellt einen OR-Graph dar, welcher mittels Pfaden zu einem AND-Graph modifiziert werden kann. Es ist zu klären, ob dieses Konzept zur Modellierung komplexer Zusammenhänge bzw. dynamischer Änderungen ausreicht oder ob (ähnlich semantischen Netzen) spezifische Arten von Kanten oder Transformationen (ähnlich Zustand-Übergangsdiagrammen) eingeführt werden müssen.

(2) Im Rahmen strukturierter SW-Entwicklung bedarf es der Einbettung konzeptueller Modelle in den Software-Entwicklungsprozeß sowie der Entwicklung von Evaluierungshilfsmittel (z.B. Zaja, 1987).

Literatur

- Bierter, W.: Wie werden wir morgen arbeiten und produzieren?, Proceedings Symposium "Angepaßte Technologie", TU Wien, grat, pp. 69-71, Jänner 1988.
- Brodie, M.L.; Mylopoulos, J.; Schmidt, J.W.: On Conceptual Modelling, Springer, New York, 1984.
- Chi, Uli, H.: Formal Specification of User Interfaces: A Comparison and Evaluation of Four Axiomatic Approaches, in: IEEE Transactions on Software Engineering, Vol. SE-11, No. 8, pp. 671-685, August 1985.
- Foley, J.D.; vanDam, A.: Fundamentals of Interactive Computer Graphics, Addison Wesley, Reading, MA, 1982.
- Friedrich, J.; Jansen, K.D.; Manz, T.: Organisationsmodelle für das Büro von morgen, in: Office Management 3/87, pp. 16-23, März 1987.
- Greene, M.: A Survey of Three Dialogue Models, in: ACM Transactions on Graphics, Vol. 5, No. 3, pp. 244-275, July 1986.
- Guedj, R.A.; tenHagen, P.J.W.; Hopgood, F.R.A.; Tucker, H.A., Duce, D.A. (eds.): Methodology of Interaction, Proceedings des gleichnamigen IFIP-Workshop, May 1979, Seillac, North Holland, Amsterdam, 1980.
- Hayes, J.: Cognitive Psychology and Interaction, in: Methodology of Interaction, eds: Guedj, R.A.; tenHagen, P.J.W.; Hopgood, F.R.A.; Tucker, H.A.; Duce, D.A., pp. 49-54, North Holland, Amsterdam, 1980.
- Hayes, P.J.: Executable Interface Definition Using Form-Based Interface Abstractions, in: Advances in Human-Computer Interaction, Vol. 1, ed.: Hartson, H.R., pp. 161-190, Ablex, Norwood, New Jersey, 1985.
- Hill, R.D.: Event-Response Systems - A Technique for Specifying Multi-Threaded Dialogues, in: Proceedings of CHI+GI 87, eds.: Carroll, J.M.; Tanner, P.P., ACM, April 5-9, Toronto, pp. 214-248, 1987.
- Kieras, D.; Polson, P.G.: An Approach to the Formal Analysis of User Complexity, in: International Journal of Man-Machine Studies 22, pp. 365-394, 1985.
- Moran, T.P.: The Command Language Grammar: a Representation for the User Interface of Interactive Computer Systems, in: International Journal on Man-Machine Studies, Vol. 15, pp. 3-50, 1981.

- Morris, Ch.W.: Foundations of the Theory of Signs, in: International Encyclopedia of Unified Science, Vol. 1, No. 2, eds.: Neurath, O.; Carnap, R.; Morris, Ch.W., pp. 77-137, University of Chicago Press, Chicago, 1938.
- Oberquelle, H.: Human-Machine Interaction and Role/Function/Action-Nets, in: Petri Nets: Central Models and Their Properties, Part I, eds.: Brauer, W.; Reisig, W.; Rozenberg, G., pp. 171-190, Springer, Berlin, 1987.
- Parnas, D.L.: On the Use of Transition Diagrams in the Design of a User Interface for an Interactive Computer System, in: Proceedings of the 24th National ACM Conference, pp. 379-385, 1969.
- Payne, S.J.: Task-Action Grammars, in: Human-Computer Interaction - INTERACT '84, ed.: Shackel, B., pp. 527-532, IFIP, Elsevier Science Publishers, 1985.
- Reisner, P.: Formal Grammar as a Tool for Analyzing Ease of Use: Some Fundamental Concepts, in: Human Factors in Computer Systems, eds.: Thomas, J.C.; Schneider, M.L., pp. 53-78, Ablex, Norwood, New Jersey, 1984.
- Satir, Virginia: Peoplemaking, Science and Behavior Books, Palo Alto, 1972.
- Semmer, N.; Schardt, L.P.: Qualifikation und Arbeit, in: Humane Arbeit, Band 4: Organisation der Arbeit, ed.: Zimmermann, L., pp. 72-150, Dezember 1982.
- Shneiderman, B.: Multiparty Grammars and Related Features for Defining Interactive Systems, in: IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 2, pp. 148-154, April 1982.
- Stary, Ch. (1988a): Help Concepts and Multimedia Applications, in: Information Technology for Organizational Systems, eds.: Bullinger, H.J.; Protonotarios, E.N.; Bouwhuis, D., Reim, F., pp. 652-659, Elsevier Science Publishers (North Holland), EEC, 1988.
- Stary, Ch. (1988b): Konzeptuelles Modellieren der Mensch-Computer Interaktion, Dissertation, TU Wien, Technisch-Naturwissenschaftliche Fakultät, September 1988.
- Studer, R.: Abstract Models of Dialogue Concepts, in: Proceedings of the 7th International Conference on Software-Engineering, March 26-29, Orlando, IEEE, pp. 420-429, 1984.
- Tauber, M.J.: On Mental Models and the User Interface, Research Report, University of Pittsburgh, 1987.
- Wasserman, A.: Extending State Transition Diagrams for the Specification of Human-Computer Interaction, in: IEEE Transactions on Software Engineering, Vol. SE-11, No. 8, pp. 699-713, August 1985.
- Watzlawick, P.; Beavin, J.H.; Jackson, D.: Pragmatics of Human Communication. A Study of Interactional Patterns, Pathologies and Paradoxes, W.W. Norton & Company, New York, 1967.
- Wolf, B.; McDonald, D.D.: Building a Computer Tutor: Design Issues, in: Computer, Vol. 17, No. 9, pp. 61-73, September 1984.
- Zaja, M.: Reliability Analysis of Interaction in Man-Machine Systems, in: Proceedings "HCI '87", International Commission on Human Aspects in Computing, August 10-14, Honolulu, Hawaii, p. 20, 1987.

Dr. Christian Stary
 Institut für Angewandte Informatik und Systemanalyse
 Abteilung für Datenbanken und Expertensysteme
 Technische Universität Wien
 Paniglgasse 16 - 181/2
 A-1040 Wien