

BENUTZERORIENTIERTE BESCHREIBUNG VON INTERAKTIVEN SYSTEMEN MIT RFA-NETZEN

Horst Oberquelle, Hamburg

Zusammenfassung: Für die benutzerorientierte Beschreibung interaktiver Systeme einschließlich ihrer organisatorischen Einbettung werden begriffliche Grundlagen und primär graphische Ausdrucksmittel vorgestellt. Sie basieren auf einer Charakterisierung von Arbeitsorganisationen auf drei Ebenen, Rollen-, Funktions- und Aktionsebene, und nutzen Ideen der Petrinetztheorie.

Einleitung

Benutzerorientierte Beschreibungen interaktiver Systeme können während der Systementwicklung die Kommunikation zwischen Entwicklern und zukünftigen Benutzern unterstützen und dem Benutzer eine Vorstellung von seiner neuen Arbeitsumgebung vermitteln. Während Einführung und Einsatz können sie zur notwendigen Entwicklung mentaler Modelle durch den Benutzer (vgl. Oberquelle, 1984a) beitragen. Sie können diese Aufgabe am besten erfüllen, wenn sie die Perspektive des Benutzers berücksichtigen und eine "Zugriffsarchitektur" (Zemanek, 1984) wiedergeben, die wir als Benutzungsmodell bezeichnen wollen. Ein Benutzungsmodell sollte die statische Struktur des Mensch-Maschine-Systems (automatisierte Komponenten und ihre organisatorische Einbettung) ebenso erfassen wie die Dynamik möglicher Abläufe.

Die verwendete Beschreibungssprache hat großen Einfluß auf die Lesbarkeit und Verständlichkeit solcher Benutzungsmodelle. Ihre Grundkonzepte, die darstellbaren Aspekte von Systemen und ihre Form (Text oder Graphik, Grad der Formalisierung) müssen dem genannten Ziel angepaßt sein. Die Vorteile von Graphik zur Visualisierung konzeptioneller Unterschiede und zur transparenten Beschreibung von komplexen Zusammenhängen sind bisher kaum ausgenutzt worden. Die heute verfügbaren PCs mit Graphiksystemen eröffnen dafür neue Möglichkeiten.

In einer umfangreichen Untersuchung (Oberquelle, 1986) wurden ein konzeptioneller Rahmen zur Erfassung von Arbeitssituationen mit interaktiver Rechnernutzung aus der Perspektive der beteiligten Personen

sowie vielfältige graphische Ausdrucksmittel erarbeitet. Für einen Text-Editor (SOS) wurde dabei eine vollständige Beschreibung mit verschiedenen, primär graphischen Hilfsmitteln (Kanal-/Instanz-Netze, Zustand-/Aktivitäts-Netze, erweiterte Syntaxdiagramme, Ikonen und Prinzipskizzen) erstellt (Oberquelle, 1984b). In einem Test mit Anfängern zeigte sich, daß diejenigen, die die graphische Beschreibung verwendet hatten, im Durchschnitt bessere Kenntnisse über das System hatten und schneller und fehlerfreier arbeiteten (Oberquelle, 1985). Damit wurde die Annahme, daß graphische Beschreibungssprachen nützlich sein können, weiter untermauert. Die eingesetzten sprachlichen Hilfsmittel waren jedoch nicht ausdrucksstark genug, um Arbeitsorganisationen mit interaktivem Rechneinsatz im allgemeinen zu erfassen. Sie wurden deshalb zu den Rollen-/Funktions-/Aktionsnetzen (RFA-Netzen) weiterentwickelt. RFA-Netze basieren auf der Idee, daß Arbeitsorganisationen sinnvoll auf drei Ebenen (Rollen-, Funktions- und Aktionsebene) zu charakterisieren sind und durch primär graphische, kohärente (Teil-) Beschreibungen von verschiedenen Aspekten und Ausschnitten mit variabler Detaillierung und Präzision (maximal semi-formal) hinreichend genau erfaßt werden können.

Wesentliche Konzepte und Darstellungsmittel von RFA-Netzen werden im 2. Abschnitt erläutert und im 3. Abschnitt an einem Anwendungsbeispiel demonstriert. Der 4. Abschnitt gibt einen Ausblick auf weitere Anwendungsmöglichkeiten und offene Fragestellungen.

Konzepte und Ausdrucksmittel von RFA-Netzen

Grundlage der RFA-Netze ist ein Begriffsgerüst, das eng mit den oben erwähnten drei Ebenen zusammenhängt. Für konzeptionell unterschiedliche Dinge und Beziehungen werden graphisch unterschiedliche Netzkonstrukte eingesetzt. Graphisch ähnliche Konstrukte sollen auf inhaltliche Verwandtschaften hinweisen und das Verständnis erleichtern. Wichtige Begriffszusammenhänge haben eine Entsprechung in netztheoretischen Zusammenhängen der graphischen Ausdrucksmittel.

Unter einem Netz verstehen wir ganz allgemein einen Graphen mit zwei Sorten von Knoten (S- und T-Elemente) und gerichteten Kanten (F-Elemente = Flußrelation) mit der Eigenschaft, daß nur Knoten unterschiedlicher Art durch F-Kanten verbunden sind. Für die folgende Darstellung sind nicht die formalen Eigenschaften von Netzen interessant sondern die speziellen Interpretationen von S-, T- und F-Elementen und deren Zusammenhang.

Wir betrachten zunächst die Hilfsmittel für die Beschreibung der statischen Struktur von Benutzungsmodellen und wenden uns dann der Dynamik zu.

(i) Rollen- und Funktionsnetze

Ausgangspunkt unserer statischen Beschreibung von Arbeitsorganisation ist die Auffassung, daß sich die Gesamtaufgabe jeder Stelle als eine Zusammenfassung von inhaltlich weitgehend abgeschlossenen, lose gekoppelten Teilaufgaben erfassen läßt. Jede solche Teilaufgabe nennen wir eine Rolle und die für sie zuständige Person den Rollenträger. Jede Person ist in der Regel Träger vielfältiger Rollen und kann weitgehend frei entscheiden, in welcher Rolle sie aktuell agiert. Dieses Rollenkonzept beinhaltet wichtige pragmatische Aspekte: den Sinngehalt der Aufgabe, die Verantwortlichkeit des Rollenträgers, die Kompetenz zur Nutzung von Entscheidungsspielräumen sowie Rechte und Pflichten gegenüber anderen Rollen und deren Trägern. Rollen können auf zweierlei Arten kooperieren: a) durch Austausch von Materialien oder Informationen über gemeinsame Schnittstellen und b) durch gemeinsame Handlungen. Der zweite Fall wird hier nicht weiter behandelt.

Beschreibungen auf der Rollenebene betrachten Rollen als atomare Einheiten und erfassen die Art der Kooperation. Die pragmatischen Aspekte werden nicht explizit dargestellt. Rollen werden durch ihre Bezeichnung und die Angabe des Rollenträgers charakterisiert.

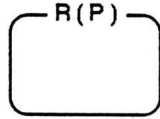
Über die pragmatischen Aspekte hinaus kann jede Rolle genauer durch ihre Realisierung in Form einzelner Tätigkeiten, die vom Rollenträger selbst oder von Maschinen ausgeführt werden, lokaler Ressourcen und durch die zwischen Tätigkeiten ausgetauschten Objekte und Daten charakterisiert werden. Eine solche funktionale Zergliederung ordnen wir als Phänomen der Funktionsebene zu. Unter einer Funktion verstehen wir eine sequentielle Tätigkeit eines bestimmten Handlungsträgers (Mensch oder Maschine) zusammen mit den nur für diese Tätigkeit verwendeten Ressourcen. Funktionen interagieren durch den Austausch von Objekten oder Daten über gemeinsame Schnittstellen.

Jede Rolle kann durch ein Geflecht von Funktionen und internen, sie verbindenden Schnittstellen funktional vollständig erfaßt werden. Jede Rolle enthält mindestens eine Funktion, jede Funktion umfaßt genau eine Tätigkeit. Die Kooperation (einschließlich Kommunikation) von Rollen wird durch Interaktion ihrer Funktionen realisiert. Schnittstellen

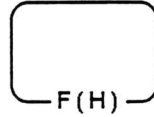
zwischen Funktionen (und Rollen) sind organisatorische Positionen, an denen sich gemeinsam genutzte reale Objekte oder Datenobjekte aufhalten können. Positionen können auch intern genutzt werden, um Objekte oder Daten für eigene Zwecke aufzubewahren.

Um diese Konzepte und Zusammenhänge zum Ausdruck zu bringen, werden die folgenden Netzkonstrukte verwendet:

T-Elemente



Rolle R
mit Träger P



Funktion F
mit Handlungsträger H



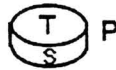
Tätigkeit T

Komplexe aus Rollen, Funktionen oder Tätigkeiten werden mit doppelter Umrandung gezeichnet. Automatisierte Funktionen oder Tätigkeiten können durch einen fetten unteren Balken hervorgehoben werden.

S-Elemente



elementare Position P
zur Aufnahme eines
Objektes vom Typ T



Speicherposition P der
Klasse S zur Aufnahme
von Objekten vom Typ T

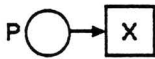


komplexe Position P,
bestehend aus mehreren
einfacheren Positionen

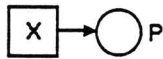
F-Elemente bedeuten Zugriffsrechte:

(Es werden hier zur Erläuterung 'neutrale' S- und T-Elemente verwendet.)

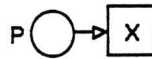
Objektfluß (schwarze Pfeilspitze) Datenfluß (weiße Pfeilspitze)



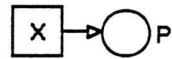
X kann Objekt
aus P entnehmen



X kann Objekt
in P ablegen



X kann Daten aus
einem Objekt in
P lesen, ohne
das Objekt aus
seiner Position
zu entfernen



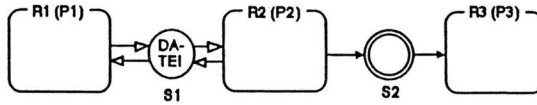
X kann Daten in
einem Objekt in
P ablegen, ohne
das Objekt aus
seiner Position
zu entfernen

Netze, die nur Rollensymbole als T-Elemente enthalten, heißen Rollennetze (R-Netze). Netze, die nur T-Elemente der Funktionsebene enthalten, heißen Funktionsnetze (F-Netze). Tätigkeiten und private Positionen können als interne Strukturen von Funktionen gezeichnet und durch das zugehörige

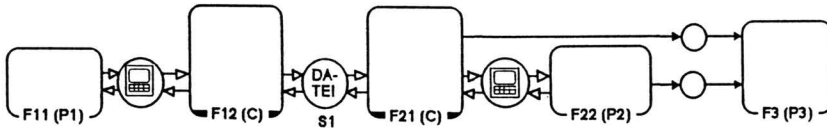
Funktionssymbol umrandet werden. Entsprechendes gilt für die Veranschaulichung von Komplexbildungen bei R- und F-Netzen.

Abb.1 Beispiele für Rollen- und Funktionsnetze

a)



b)



c)

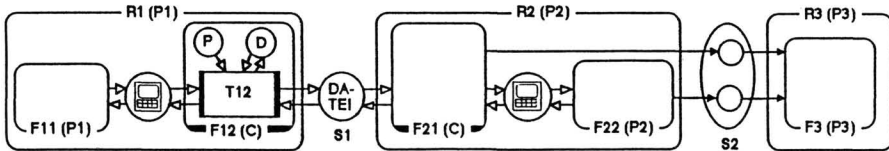


Abb.1 a) zeigt schematisch ein Rollennetz, in dem R1 und R2 durch Informationsaustausch und R2 und R3 durch Dokumentfluß über Schnittstellen kooperieren. Abb.1 b) stellt eine mögliche Realisierung auf der Funktionsebene dar, bei der für die Kommunikation rechnergestützter Datenaustausch genutzt wird. a) kann offensichtlich aus b) durch eine Netzvergrößerung gewonnen werden. c) gibt den Zusammenhang zwischen a) und b) in einem gemischten Rollen-/Funktions-(RF-)Netz an und zeigt darüber hinaus, daß die automatisierte Funktion F12 über private Positionen verfügt, auf die von ihrer Tätigkeit T12 zugegriffen werden kann.

(ii) Aktionsnetze

Auf der Aktionsebene konzentrieren wir uns auf die Dynamik von Arbeitsabläufen und erfassen sie in Aktionsnetzen (A-Netzen). Es wird angenommen, daß die Handlungsträger nach Plänen arbeiten, die in Beschreibungen explizit gemacht werden können. Die kleinsten Aktionseinheiten, die nicht weiter zerlegt werden können oder sollen, bezeichnen wir als elementare Handlungen. Komplexe Handlungen und Aktionen ergeben

sich als spezielle Abstraktionen, die sich anhand der Netzdarstellung später am besten erläutern lassen.

Elementare Aktionsnetze haben elementare Handlungen als T-Elemente und sind durch Kontrollfluß mit Zuständen der Handlungsträger zeitlich ('Kontrollaspekt') und durch Objekt- bzw. Datenfluß mit elementaren Positionen räumlich ('Objektaspekt') verbunden. Zustände und elementare Positionen bilden also die S-Elemente.

Die folgenden Netzkonstrukte werden in elementaren Aktionsnetzen benutzt:

<u>Kontrollaspekt:</u>			<u>Objektaspekt:</u>
T-Elemente	S-Elemente	F-Elemente	S- und F-Elemente
< Name > ; < Bedingung > < Wirkung >			elementare Positionen; Objektfluß und Datenflußpfeile wie bei R- und F-Netzen, interpretiert als <u>Zugriff</u> (statt als Zugriffsrecht)
elementare Handlung	Zustand (kleiner Kreis)	Kontrollfluß (offener Pfeil)	

Jede elementare Handlung wird als bedingte Handlung aufgefaßt, die nur dann ausgeführt wird, wenn die angegebene <Bedingung> durch die zugreifbaren Objekte erfüllt ist und der Handlungsträger sich für die Handlung entscheidet. Die Ausführung hat die angegebene <Wirkung> auf die Ausgangsobjekte. Aktuelle Situationen eines elementaren A-Netzes können durch Markierung von Zuständen mit individuellen Kontrollmarken und von Positionen mit individuellen Objektmarken beschrieben werden. Die Dynamik kann als Fluß dieser Marken gemäß einer Schaltregel im Sinne der Netztheorie exakt definiert werden, was wir hier nicht näher ausführen können.

In allgemeinen Aktionsnetzen (A-Netzen) werden zusätzlich die folgenden Netzelemente zugelassen, die sich durch Netzvergrößerungen ergeben (vgl. Abb.2) und eine vergrößerte Modellierung erlauben.

Name /
Erklärung

Komplexe Handlung, die mehrere einfachere Handlungen zusammenfaßt und Zwischenzustände enthalten kann

Name /
Erklärung

elementare Aktion, die eine elementare Handlung und für sie private Positionen umfaßt

Name /
Erklärung

komplexe Aktion, die mehrere einfachere Aktionen und nur diesen gemeinsame Positionen zusammenfaßt und Zwischenzustände enthalten kann

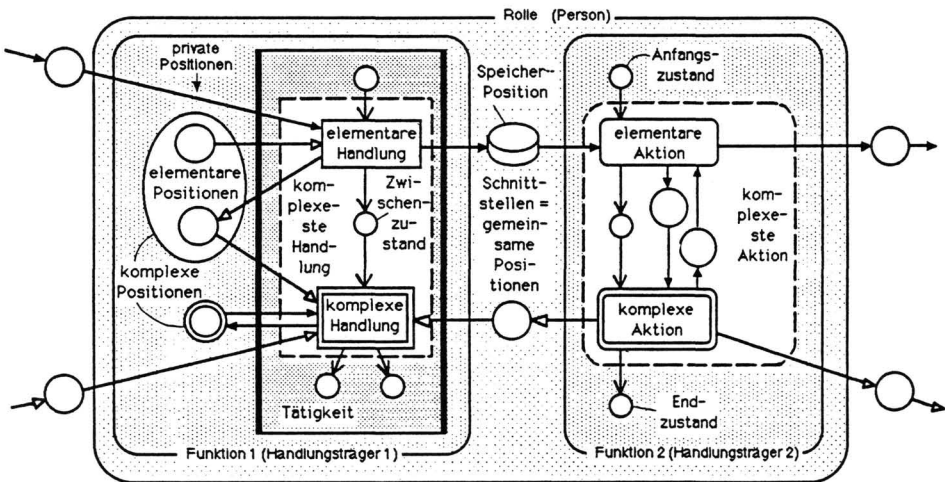
Außerdem können die größeren Positionsdarstellungen (Speicherposition, komplexe Position, vgl.(i)) verwendet werden.

Jedes A-Netz kann eindeutig in ein Kontrollnetz (nur Zustände als S-Elemente) und in ein Objektnetz (nur Positionen als S-Elemente) zerlegt werden. Häufig reicht einer der beiden Aspekte für das Verständnis auf der Aktionsebene aus. Im 3.Abschnitt werden Kontrollnetze für die Modellierung der Dynamik eines Editors eingesetzt.

(iii) Rollen-/Funktions-/Aktionsnetze

Einzelbeschreibungen auf den drei Ebenen mit variierender Verfeinerung und gemischte Beschreibungen, die Konzepte verschiedener Ebenen in einem Diagramm kombinieren und damit ihren Zusammenhang transparenter machen, fassen wir unter dem Sammelbegriff Rollen-/Funktions-/Aktionsnetze (RFA-Netze) zusammen. Entsprechend verwenden wir die Bezeichnungen RF-, RA- und FA-Netze für speziellere Kombinationen. Abb.2 gibt einen schematischen Überblick über die Darstellungsmittel der drei Ebenen und ihren Zusammenhang über Netzvergrößerungen bzw. -verfeinerungen.

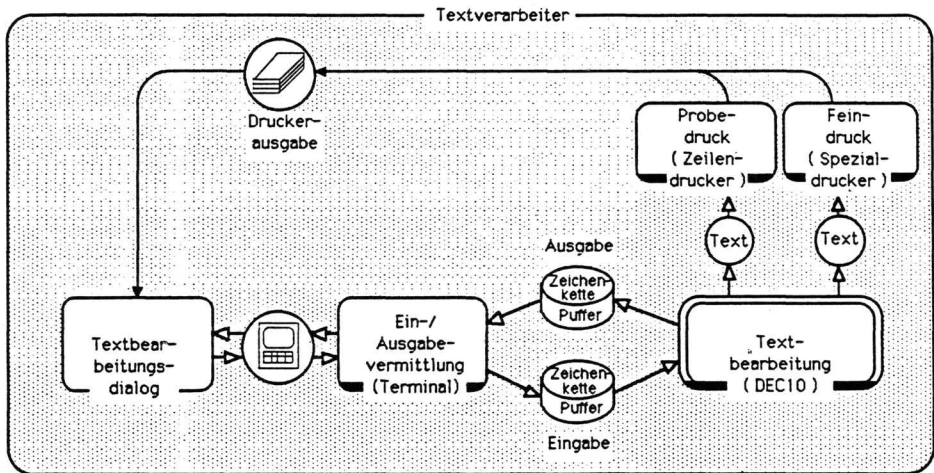
Abb.2 Beziehungen zwischen RFA-Konzepten und -konstrukten



Ein Anwendungsbeispiel

Um einige der Einsatzmöglichkeiten von RFA-Netzen zu demonstrieren, geben wir Ausschnitte aus einer überarbeiteten Beschreibung für den SOS-Editor (vgl. Oberquelle, 1984b). Die Arbeit mit dem Texteditor wird als eine Rolle 'Textverarbeiter' erfaßt, die automatisierte Funktionen des Time-sharing-Systems DEC10 benutzt, um gedruckte Texte für eigene Zwecke herzustellen. Die im Rechner gespeicherten Texte werden als stationäre Datenobjekte behandelt. Die Möglichkeit der Kooperation mit anderen Rollen durch rechnervermittelten Austausch von gespeicherten Texten wird nicht betrachtet. Sie würde eine andere Sicht der Rolle 'Textverarbeiter' ergeben. Die Struktur der Rolle wird durch das Rollen-/Funktions-Netz in Abb.3 grob erfaßt.

Abb.3 Die Rolle 'Textverarbeiter' und die Interaktionsmöglichkeiten mit automatisierten Funktionen (RF-Netz)



Für die gezielte Nutzung der Editiermöglichkeiten und das Verständnis der auf dem Bildschirm angezeigten Daten benötigt der Textverarbeiter ein hinreichend präzises, vollständiges Modell der statischen Aufbaustruktur des Funktionskomplexes 'Textbearbeitung' und seines dynamischen Verhaltens. Abb.4 macht deutlich, daß zwei wesentliche Funktionskomplexe, 'Monitor' und 'SOS-Editor', zu unterscheiden sind. Beide haben Zugriff auf die Benutzerdateien im Dateisystem, welches eine Schnittstelle zwischen beiden darstellt. Außerdem interagieren sie durch den Austausch von Aufruf- und Resultat-Daten, die jeweils nach Empfang vom Empfänger gelöscht werden. Die Druckfunktionen sind nur über den

'Monitor' erreichbar.

Die weitere Verfeinerung des Funktionskomplexes 'SOS-Editor' führt zu vier ähnlichen Funktionen, die alle durch Kombination und Spezialisierung aus dem in Abb.5 gezeigten Funktionsnetz hervorgehen.

Abb.4 Die Textbearbeitungsfunktionen (1.Verfeinerung, F-Netz)

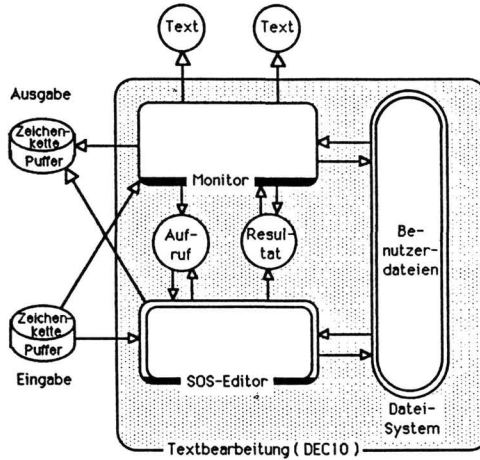
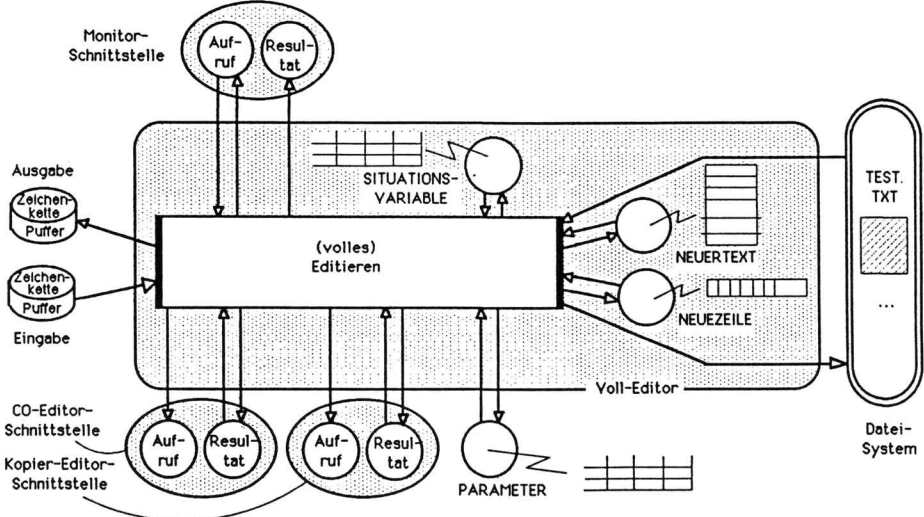


Abb.5 Statische Struktur eines Volleditors und seiner Umgebung (F-Netz)

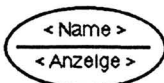


Dieses Funktionsnetz stellt alle für das Verständnis wesentlichen Komponenten dar. Die Tätigkeit '(volles) Editieren' hat lesenden

Zugriff auf Benutzerdateien, kann einen neuen Text und eine neue Zeile bearbeiten und geänderte Texte in Dateien des Datei-Systems zurückschreiben. Sie wird durch lokale Situationsvariable und durch Parameter, auf die auch andere Komponenten zugreifen können, gesteuert. Beide sind durch Editierhandlungen veränderbar. Die Objekttypen in den wichtigsten Positionen sind durch Ikonen angegeben. Sie können ebenfalls informal durch graphische Strukturbeschreibungen und erläuternden Text erklärt werden (vgl. Oberquelle, 1985).

Eingaben sind Zeichenketten, die gemäß der Kommandosprachensyntax des Monitors bzw. des SOS-Systems konstruiert sein müssen. Sie können durch erweiterte Syntaxdiagramme (vgl. Oberquelle, 1985) mit graphischen Mitteln vollständig und anschaulich erfaßt werden. Ausgaben sind ebenfalls Zeichenketten, die den Inhalt der bearbeiteten Datenobjekte (z.B. einen Bereich aus NEUERTEXT oder NEUEZEILE) oder Mitteilungen an den Träger der Rolle 'Textverarbeiter' als gleichzeitigen Handlungsträger des Textbearbeitungsdialogs darstellen.

Die Dynamik der Editiertätigkeit kann vollständig mit Aktionsnetzen beschrieben werden. Die nachfolgenden Abbildungen zeigen Teile einer solchen stufenweisen Beschreibung, die den Kontrollaspekt hervorhebt. Bei den Zuständen wird zur Erhöhung der Transparenz zwischen Dialogzuständen und internen Zuständen unterschieden:



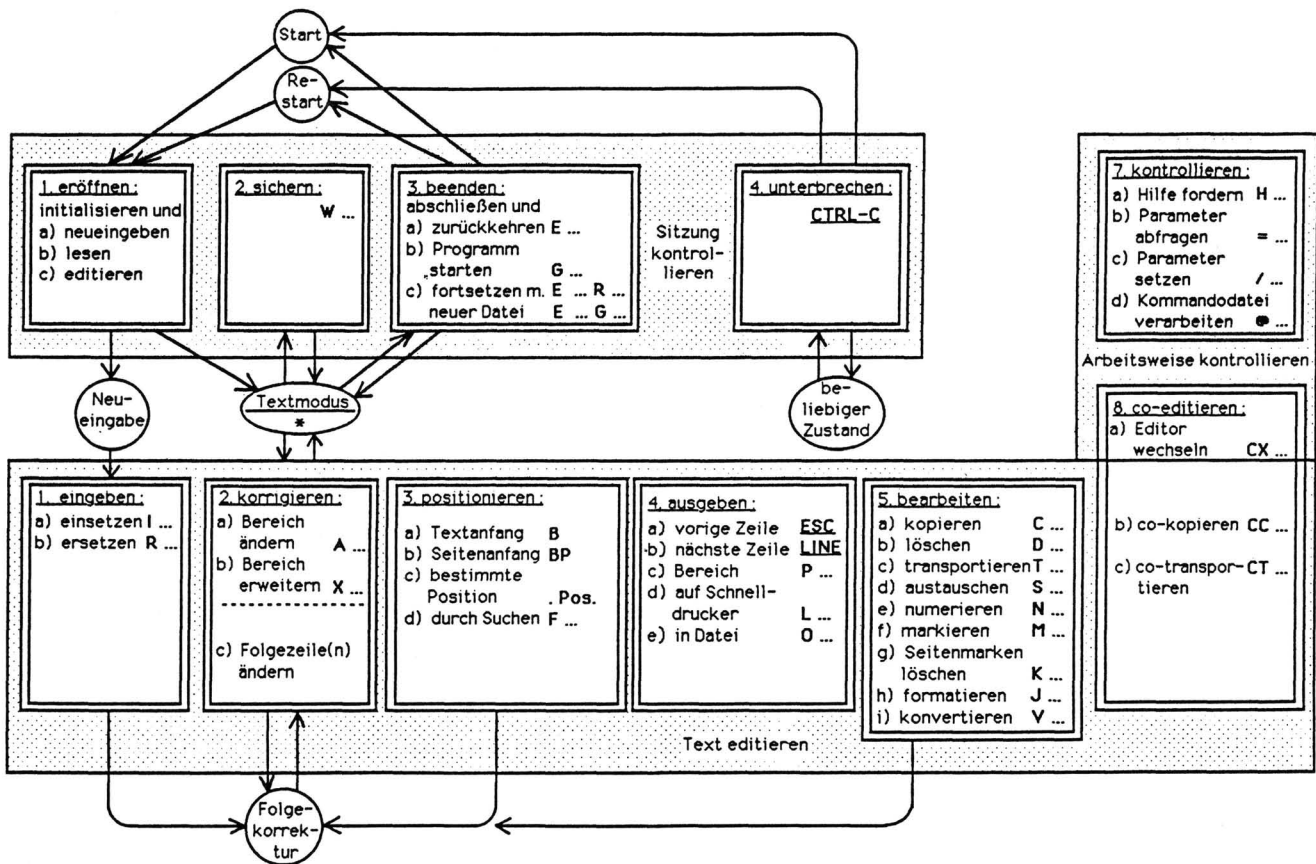
Dialogzustand (mit Anzeige): Die nächste Handlung wird durch die Eingabe des Benutzers ausgewählt. Die Angabe der Anzeige ist optional. Ist sie angegeben, so erscheint sie jedesmal, wenn dieser Zustand erreicht wird.



Interner Zustand: Die nächste Handlung wird vom Funktionsträger selbständig in Abhängigkeit vom Inhalt seiner Datenobjekte ausgewählt.

Abb.6 zeigt einen Überblick über alle Klassen von Editierhandlungen, die durch Aufruf des Editors aus den internen Zuständen 'Start' bzw. 'Restart' oder aus dem Dialogzustand 'Textmodus' erreichbar sind. 'Start' ist Anfangs- und Endzustand der Tätigkeit 'Editieren'. Bei jeder Klasse sind einzelne enthaltene Handlungen verbal und durch Kurzformen der für ihre Auswahl benötigten Kommandos angegeben. Eine Sonderstellung nimmt die Handlung 'unterbrechen' ein, die aus jedem Zustand des Editors durch Eingabe des Kommandos CTRL-C gefordert werden kann. Stellvertretend für alle diese Zustände ist ein zusätzlicher Zustand mit der Bezeichnung 'beliebiger Zustand' eingefügt worden.

Abb. 6 Handlungsklassen von SOS und zugehörige Kommandos (A-Netz, speziell Kontrollnetz mit Handlungskomplexen)



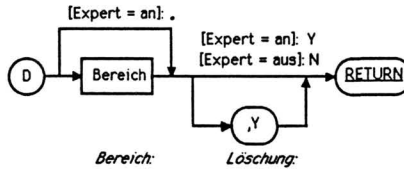
Durch weitere Verfeinerungen auf der Aktionsebene können entsprechende Diagramme für jeden einzelnen Handlungskomplex und die den einzelnen Kommandos zugeordneten Handlungen und sie verbindende Zustände angegeben werden. Aber selbst die Handlung, die einem Kommando entspricht, kann eine weitere Verfeinerung erfordern, z.B. wenn ihre Ausführung zu weiteren Dialogzuständen führt oder wenn abhängig von den Argumenten unterschiedliche Teilhandlungen gewählt werden. Abb.7 zeigt ein Beispiel für eine semi-formale, vorwiegend graphische Beschreibung eines Kommandos und der damit ausgewählten Handlung.

Abb.7 Das Kommando 'delete' und die Handlung 'löschen'

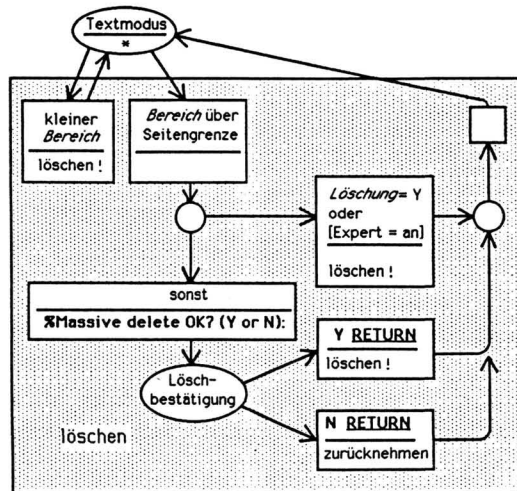
löschen (delete)

Aufgabe: Löschen eines Bereiches in NEUERTEXT, eventuell mit Bestätigung bei Überschreiten einer Seitenmarke (massive delete)

Syntax:



Semantik:



Die Handlung wird hier zunächst verbal erläutert. Die Struktur des Kommandos wird durch ein erweitertes Syntaxdiagramm angegeben,

welches deutlich macht, welche Argumente geliefert werden (im Beispiel: Bereich und Löschung) und wie fehlende Angaben ergänzt werden (durch Angabe der Ergänzungen ('defaults') an den durchgezogenen Kanten). Die Semantik des Kommandos wird durch ein Kontrollnetz erfaßt, das in Bedingungen und Wirkungen elementarer Handlungen Bezug nimmt auf die aktuelle Eingabe (fett gedruckt), Argumente (kursiv) und Systemvariable (= Situationsvariable und Parameter; in eckigen Klammern). Die internen und die externen Wirkungen (= Ausgaben; in Fettdruck) werden in dieser Reihenfolge notiert und sichtbar voneinander getrennt.

Aktionsnetze der hier verwendeten Art decken alle wesentlichen Möglichkeiten von Zustandsdiagrammen ab, wie sie von verschiedenen Autoren (Jacob, 1983; Kieras & Polson, 1984; Wasserman, 1985) für die Spezifikation von interaktiven Systemen vorgeschlagen wurden. Sie zeigen in Kombination mit den anderen Diagrammen wesentlich deutlicher den Zusammenhang zwischen Kontroll- und Objektaspekt und zwischen dynamischer und statischer Struktur des Gesamtsystems in seiner Umgebung.

Ausblick

Wir haben wesentliche Teile aus einem umfassenden Konzept zur primär graphischen Beschreibung von Arbeitsorganisationen mit und ohne Rechneinsatz vorgestellt. Sie erfassen zusammenhängend die Dynamik, die Statik und einen Teil der Pragmatik und können für die Formulierung von Benutzungsmodellen eingesetzt werden. Weitere Konzepte von RFA-Netzen behandeln die hier nicht näher erläuterte Kooperation bzw. Interaktion über gemeinsame Handlungen, zusätzliche Abstraktionsmöglichkeiten, die Präzisierung des Objektbegriffs und die Verarbeitung strukturierter Objekte (vgl. Oberquelle, 1986).

Die hier vorgestellten Hilfsmittel sind primär für Analytiker und Designer als Autoren von Beschreibungen und für Benutzer als Leser gedacht, vergleichbar der Situation zwischen Architekt und Bauherrn. RFA-Netze können in der Praxis an der Wandtafel oder auf Papier in Rohfassungen entstehen und mit graphischen Editoren in eine modifizierbare und qualitativ hochwertig druckbare Form gebracht werden. Alle Abbildungen dieses Beitrags wurden z.B. mit dem Graphiksystem MacDraft auf einem Apple Macintosh erstellt.

Die RFA-Konzepte sind bisher zur transparenteren Beschreibung einer Vielzahl von beispielhaften Situationen aus der Literatur herangezogen worden und haben dabei ihre Flexibilität bewiesen. Das oben

behandelte SOS-System ist eines dieser Beispiele. Ihre Nützlichkeit in praktischen Anwendungssituationen während der Systementwicklung und für die benutzerorientierte Dokumentation muß noch weiter erforscht werden.

Literaturverzeichnis

- Jacob, R.J.K. (1983). Using Formal Specifications in the Design of a Human-Computer Interface. Communications of the ACM 26,3, S. 259 - 264
- Kieras, D., Polson, P. (1984). A Generalized Transition Network Representation for Interactive Systems. in: Janda, A. (Ed.). Human Factors in Computing Systems. Amsterdam: North-Holland, S. 103 - 106
- Oberquelle, H. (1984a). On Models and Modelling in Human-Computer Co-Operation. in: van der Veer et al. (Eds.). Readings in Cognitive Ergonomics - Mind and Computers. Lecture Notes in Computer Science, Vol. 178, Berlin, Heidelberg, New York, Tokyo: Springer, S. 26 - 43
- ders. (1984b). Beschreibung von Dialogsystemen mit Diagrammen: Der Text-Editor SOS. Universität Hamburg, Fachbereich Informatik, Mitteilung Nr. 130
- ders. (1985). Semi-formal Graphic Modelling of Dialog Systems. in: Papers presented at the 6th European Workshop on Applications and Theory of Petri Nets. Helsinki University of Technology, Digital Systems Laboratory, Espoo, S. 1 - 16 und Universität Hamburg, Fachbereich Informatik, Bericht Nr. 113
- ders. (1986). Sprachkonzepte für die kooperative Rollenentwicklung. Habilitationsschrift, Universität Hamburg, Fachbereich Informatik
- Wasserman, A.I. (1985). Extending State Transition Diagrams for the Specification of Human-Computer Interaction. IEEE Transactions on Software Engineering, Vol. SE-11,8, S. 699 - 713
- Zemanek, H. (1984). Über die Grenzen der Einsicht im Computerwesen. in: Wettstein, H. (Hrsg.). Architektur und Betrieb von Rechensystemen. Informatik-Fachberichte, Band 78, Berlin, Heidelberg, New York, Tokyo: Springer, S. 1 - 25

Prof. Dr. Horst Oberquelle
 Universität Hamburg
 Fachbereich Informatik
 Rothenbaumchaussee 67/69
 D-2000 Hamburg 13