

ADAPTIERBARE BENUTZERSCHNITTSTELLEN

Christian Rathke, Boulder

Zusammenfassung

Die Benutzerschnittstelle zu Informationssystemen ist von entscheidender Bedeutung. Für den Benutzer ist sie das System selbst. Die Benutzerschnittstelle der meisten Informationssysteme bietet sich den Benutzern in immer derselben Weise dar. Benutzerschnittstellen sind kaum an Aufgabe und Benutzer anzupassen. Diese Tatsache reduziert den Nutzen eines solchen Systems beträchtlich. Informationssysteme werden daher oft als umständlich, ineffektiv und für das jeweilige Problem als nicht so ganz geeignet erfahren. Einen Ausweg aus dem Dilemma bieten adaptierbare Benutzerschnittstellen.

Adaptierbare Benutzerschnittstellen werden an Hand eines wissensbasierten Tabellenkalkulationssystems konkretisiert, das dem Benutzer Gestaltungsmöglichkeiten auf verschiedenen Ebenen bietet. Mit FINANZ kann eine *formularbasierte* Benutzerschnittstelle an Aufgabe und Benutzer angepaßt werden. Sowohl *Aussehen* wie *Struktur* der Formulare werden *interaktiv* gestaltet. Die so gestaltete Benutzerschnittstelle vermag den Benutzer bei seiner Aufgabe wirkungsvoll zu unterstützen.

1. Einleitung

Die Frage nach dem Nutzen von Informationssystemen für den Benutzer muß umformuliert werden in: "Welche Art von Informationssysteme nützen dem Benutzer?". Wir geben eine Antwort auf diese Frage, indem wir Anforderungen an Benutzerschnittstellen formulieren und ein konkretes, prototypisches System vorstellen, das versucht, den Anforderungen gerecht zu werden.

Indem wir uns auf die Benutzerschnittstelle konzentrieren, klammern wir bewußt Fragen nach dem *Nutzen von Informationssystemen überhaupt* aus. Wir beschäftigen uns nicht damit, ob ein System zum Auffinden von Literatur, Kaufhauskataloge, Fahrplanauskunftssysteme, Personalinformationssysteme, Polizeikarteien, usw. grundsätzlich von Nutzen oder Schaden sind. Wir wollen auch offen lassen, wer im einzelnen die Nutznießer dieser Informationen sind und zu wessen Vor- und Nachteil sie verwendet werden. Wir sind uns aber bewußt, daß diese Fragen im Vorfeld geklärt werden müssen, bevor von Schaden und Nutzen eines computerbasierten Informations-

systems für den einzelnen Benutzer gesprochen werden kann.

Die Benutzerschnittstelle zu Informationssystemen ist von entscheidender Bedeutung. Für die Benutzer ist sie das System selbst. Durch sie erfahren sie von seinen Eigenschaften. Mittels der Benutzerschnittstelle machen sie ihre Wünsche dem System bekannt. Der Nutzen für den Benutzer wird daher in erster Linie durch die Gestaltung der Benutzerschnittstelle bestimmt.

Die Benutzerschnittstelle der meisten Informationssysteme bietet sich den Benutzern in immer derselben Weise dar. Die Suche nach Information wird unabhängig von Aufgabe oder Benutzer formuliert. Antworten des System sind ebenfalls an einem "Durchschnittsbenutzer" ausgerichtet. Benutzerschnittstellen sind kaum an Aufgabe und Benutzer anzupassen. Diese Tatsache reduziert den Nutzen eines solchen Systems beträchtlich. Auf dem Benutzer lastet die Aufgabe, seine Anfragen so zu formulieren, daß sie vom Informationssystem in seinem Sinne verstanden werden. Dem Benutzer bleibt es überlassen, Systemäußerungen zu interpretieren und die für sein Problem relevanten Informationen zu filtern.

Informationssysteme werden daher oft als umständlich, ineffektiv und für das jeweilige Problem als nicht so ganz geeignet erfahren. Einen Ausweg aus dem Dilemma bieten adaptierbare Benutzerschnittstellen. Ihre Notwendigkeit wird mit den folgenden Thesen begründet:

1. Es gibt für ein gegebenes Informationssystem keine "beste" Benutzerschnittstelle. Diese hängt in starkem Maße von der konkreten Aufgabe und dem jeweiligen Benutzer ab.
2. Aufgaben und Benutzer können vom Systemdesigner unmöglich in allen Einzelheiten vorhergesehen und in das Systemdesign mit einbezogen werden. Es stellt sich weniger das Problem einer optimalen Benutzerschnittstelle, sondern ob eine vorhandene leicht an wechselnde Rahmenbedingungen angepaßt werden kann.
3. Benutzerschnittstellen müssen durch den Endbenutzer angepaßt werden können, d.h. die Anpassung des Systems muß sich in der terminologischen Welt des Benutzers, seiner Aufgabe, seines Kenntnisstandes vollziehen lassen.
4. In anpaßbaren Systemen wird die Benutzerschnittstelle selbst zum Gegenstand der Betrachtung. Daher müssen Eigenschaften der Benutzerschnittstelle explizit repräsentiert werden.
5. Die Herausforderung "intelligenter Systeme" besteht in der Nutzbarmachung der Intelligenz für die Kommunikation mit dem Benutzer und erst in zweiter Linie für das Lösen "schwieriger" Aufgaben.

In den folgenden Abschnitten werden diese Thesen an Hand eines wissensbasierten Tabellenkalkulationssystems konkretisiert, das dem Benutzer Gestaltungsmöglichkeiten auf verschiedenen Ebenen bietet. Unterstützt werden diese Möglichkeiten

durch Verwendung von ObjTalk (Rathke, 1986), einer objektorientierten Wissensrepräsentationssprache, die vom Autor über einen längeren Zeitraum entwickelt wurde und in einer Vielzahl von Systemen eingesetzt wird (Fischer, Gunzenhaeuser, 1986).

2. FINANZ

FINANZ ist ein System, mit dem *formularorientierte* Anwendungen erstellt und modifiziert werden können (Rathke, 1983). Sowohl Aussehen wie Struktur der Formulare werden *interaktiv* gestaltet. Das entstehende System ist in der Lage, den Benutzer bei seiner Aufgabe wirkungsvoll zu unterstützen. Abbildung 2-1 zeigt eine typische Bildschirmsituation beim Ausfüllen von Finanzierungsplänen für ein Projekt.

Abbildung 2-1: Finanzierungspläne für ein Projekt

The screenshot displays several overlapping windows from the FINANZ system:

- Jahresgehaelter**: A window showing salary data for different categories.

Bat I	60000
Bat Ia	55000
Bat IIa/Ib	40000
Bat Ub-IIa	
Bat K-Uc	
Lohnempf.	
Entgelte	
- Personalposten/Bat Ia**: A window showing details for Bat Ia.

Anzahl	2
Jahresgehalt	55000
Insgesamt	110000
- Personal**: A window showing a personnel list with columns for dates and amounts.

von	bis	Jahr	Insgesamt
1.1.1986	1.1.1988		
0	Bat I	0	
2	Bat Ia	110000	110000
9	Bat IIa/Ib	360000	
0	Bat Ub-IIa	0	
0	Bat K-Uc	0	
0	Lohnempf.	0	
0	Entgelte	0	
			470000
- INVESTITION**: A small icon of a document with the word 'INVESTITION' written on it.

Ein mit FINANZ erstelltes System zum Ausfüllen von Finanzierungsplänen für einen Projektantrag beim Bundesministerium für Forschung und Technologie.

Die Verwendung von Formularen als die grundlegende Form der Kommunikation mit FINANZ hat Vorteile für die Benutzer und das System.

Bildschirmformulare orientieren sich an Formularen der Alltagswelt. Formulare sind eine der wichtigsten, standardisierten Formen zum Austausch von Informationen. Sie klassifizieren und strukturieren Informationen. Aufgrund dieser Eigenschaften eignen sich Formulare auch für die Interaktion mit Computersystemen. Im Gegensatz zur Kommunikation in natürlicher Sprache ist durch das Formular eine

Strukturierung vorgegeben. Dies befreit Benutzer und System von Spezifizieren und Erkennen des Kontextes, in dem eine Äußerung gemacht wird. Einige Nachteile herkömmlicher Formulare können durch Verwendung des elektronischen Mediums ausgeglichen oder in Vorteile umgekehrt werden. Papierformulare sind oft verwirrend und schwer verständlich. Zum richtigen Ausfüllen *fehlen* häufig wichtige Informationen. Demgegenüber enthalten Formulare oft Felder und Texte, die für den Ausfüllenden *nicht relevant* sind. Ein Computersystem kann in mancher Beziehung flexibler als Papier und Bleistift sein.

Bildschirmformulare können der jeweiligen Aufgabe angepaßt werden. Die Flexibilität eines Computersystems macht es unnötig, Informationen für verschiedene Zwecke *gleichzeitig* in einem Formular darzustellen. In den verschiedenen Bearbeitungsstufen - vom Ausfüllen bis zur Verwendung des Formulars - werden unterschiedliche Anforderungen an Aufbau und Inhalt gestellt.

Berechnungen werden vom System ausgeführt. Feldinhalte, die von anderen Formularfeldern abhängen, können automatisch berechnet werden. In Formularen gibt es üblicherweise Teilsummen, Produkte, Überträge und prozentuale Abhängigkeiten zwischen Feldern. In wissensbasierten Systemen kann "Berechnung" komplexe Entscheidungsprozesse beinhalten. Abhängigkeiten sind dann nicht auf arithmetische Beziehungen beschränkt. In einem Forschungsprojekt stehen z.B. die Anzahl und Qualifikation von Mitarbeitern, die Aufgabenstruktur, der effektive Einsatz von Mitteln und die Gesamtlaufzeit des Projekts in einer komplexen Beziehung zueinander.

Lokale Überprüfungen der Feldinhalte werden unmittelbar nach oder schon während der Eingabe vorgenommen. Mehrdeutige oder ungenaue Eingaben können auf Grund des Wissens über die Funktion des Systems verstanden werden (Abb. 2-2). Darüberhinaus können unvollständige Angaben ergänzt und kleinere Fehler korrigiert werden.

Systeme mit einer großen Funktionalität wie FINANZ können von den Benutzern oft nicht beeinflußt und an ihre Bedürfnisse angepaßt werden. Die Modifizierbarkeit des Anwendungssystems durch den Benutzer ist aber von grundlegender Bedeutung:

1. Verschiedene, formularorientierte Anwendungen unterscheiden sich in wesentlichen Punkten, so daß Standardisierungen nur selten möglich sind. Systeme müssen daher den Wünschen der Anwender angepaßt werden. Ist die Anpassung nur auf der Ebene der Programmierung möglich, dann ist dazu ein Programmier-Experte erforderlich.
2. Aufgaben und damit Systemanforderungen sind einem ständigem Wechsel unterworfen. In relativ kurzen Abständen müssen Änderungen aufgenommen und

Abbildung 2-2: Interpretation und Ergänzung von Benutzereingaben

Beim Ausfüllen eines Datumfeldes wird die Eingabe als Datum analysiert und mit plausiblen Werten ergänzt. Auf Grund des Wissens über die Funktion eines Feldes kann entschieden werden, welche Eingaben möglich und sinnvoll sind. Erwartungen über Benutzereingaben erleichtern die Analyse.

durch das System reflektiert werden. Die wechselnden Anforderungen können nicht vom Systemdesigner vorausgedacht werden. Deshalb ist die Modifizierbarkeit des Systems durch den Anwender selbst notwendig.

3. Die Expertise zur Anpassung des Systems liegt auf der Seite des Anwenders. Die Möglichkeit, eigenständig Veränderungen vornehmen zu können, vermeidet auch Kommunikationsprobleme mit Systemherstellern.

Die Möglichkeit, Systeme zu gestalten und sie an veränderte Bedingungen anzupassen, macht sie zu Werkzeugen in den Händen der Anwender. Sie sind keine festgefügtten Blöcke, die, so wie sie sind, akzeptiert werden müssen. Ihre Funktion und ihre Einsatzweise werden vom Anwender bestimmt. Dazu sind keine speziellen Kenntnisse über die Implementationssprache erforderlich. Die Kontrolle liegt damit beim Anwender des Systems. In den folgenden Abschnitten soll dies an Hand eines Beispiels verdeutlicht werden.

3. Benutzermodifikationen in FINANZ

Im Finanzierungsplan für ein Projekt werden die beantragten Personalkosten in einem vorgegebenen Formular dargestellt (Abb. 3-1). In der ersten Spalte werden die Anzahl der Personen einer bestimmten Gehaltsklasse aufgeführt. Zusammen mit der Jahrespauschale ergeben sich die Gesamtausgaben bezogen auf Gehaltsklassen. Die Endsumme weist die gesamten, beantragten Personalmittel aus. Die Jahrespauschalen der einzelnen Gehaltsstufen befinden sich auf einem zweiten Formular.

Mit Hilfe dieser beiden Formulare kann ein weiteres Formular erzeugt werden, das einen einzelnen Personalposten aus einer anderen Sicht darstellt. Im Vergleich zum Papierformular stellt das Bildschirmformular von Abbildung 3-1 nur einen Ausschnitt dar. Dieser Ausschnitt zeigt die für das augenblickliche Bearbeitungsstadium relevanten Informationen. Es konzentriert sich auf die Personalkosten der Projekt-

Abbildung 3-1: Die Personalkosten in einem Finanzierungsplan

Personal

von 1.1.1986 bis 1.1.1988

0	Bat I	0	0
2	Bat Ia	110000	110000
9	Bat IIa/Ib	360000	360000
0	Bat Ub-IIa	0	0
0	Bat K-Uc	0	0
0	Lohnempf.	0	0
0	Entgelte	0	0
			470000

Jahresgehaelter

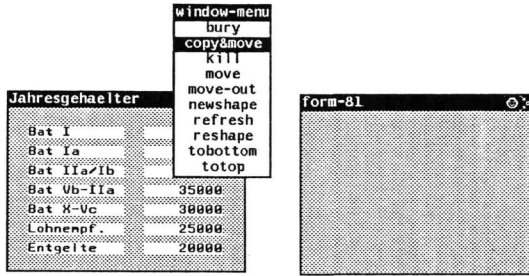
Bat I	60000
Bat Ia	55000
Bat IIa/Ib	40000
Bat Ub-IIa	35000
Bat K-Uc	30000
Lohnempf.	25000
Entgelte	20000

finanzierung. In einem anderen Bearbeitungsstadium können weitere Informationen relevant werden. Mit FINANZ ist es daher möglich, neue Formulare zu erzeugen, die Daten unter einer anderen Perspektive zeigen.

Allgemein beschreiben Formulare in FINANZ *Sichten auf Datenstrukturen*. Diese Datenstrukturen bilden die *interne Repräsentation* des Systems. Sie enthalten Daten über Projekte, Laufzeiten, Zeitpunkte, Personen, Qualifikationen und Aufgaben. Die *externe Repräsentation* besteht aus Formularen, Texten, Feldern, Fenstern, Menüs und Icons. Zwischen externer und interner Repräsentation bestehen Beziehungen, die festlegen, wie ein Element der internen Repräsentation extern dargestellt oder angesprochen wird. Umgekehrt ist festgelegt, wie sich Veränderungen der externen Repräsentation intern auswirken. Aufgrund dieser Organisation können verschiedene Sichten auf dieselbe interne Struktur existieren. *Interne Strukturen werden über externe Sichten dargestellt und manipuliert*. Das Erzeugen eines neues Formulars bedeutet also für FINANZ die Erzeugung einer neuen Sicht auf interne Datenstrukturen.

Der Benutzer erzeugt ein neues Formular, indem er ein vorhandenes kopiert (Abb. 3-2). Das Erzeugen eines neuen Formulars durch Kopieren eines vorhandenen vermeidet die Probleme, die bei einer formalen Spezifikation der Formulareigenschaften auftreten würden: Der Benutzer muß keine Spezifikationssprache lernen und das System benötigt keine Komponente zur Interpretation der Benutzeräußerung. Die Kopieroperation macht dem Benutzer keine Schwierigkeiten, weil die Interaktionsweise und die Semantik von anderen Anwendungssystemen auf das Formularsystem übertragen werden kann.

Abbildung 3-2: Das Erzeugen eines Formulars

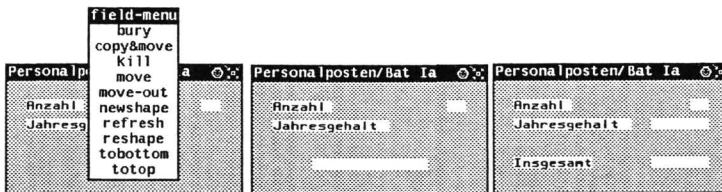


Neue Objekte werden durch Kopieren vorhandener Objekte und anschließendes Editieren erzeugt.

Auf dieselbe grundlegende Weise werden neue Formularfelder kreiert. Der Benutzer erzeugt Zahlen-, Text- oder Datumsfelder, indem er Kopieroperationen auf entsprechende Felder anwendet. Neben den Feldinhalten werden die spezifischen Eigenschaften der Felder übernommen, d.h. die Kopie eines Datumsfelds ist wiederum ein Datumsfeld. Die Darstellung des Datums und die Interpretation der Benutzereingabe als Datum gelten für das kopierte Feld wie für das Original.

Durch verschiedene Darstellungs-Operationen werden die Felder an ihren Platz gerückt (Abb. 3-3). Formularfelder sind Bildschirmobjekte, die Fenstern ähnlich sind: Sie können wie diese bewegt und in ihrer Größe verändert werden.

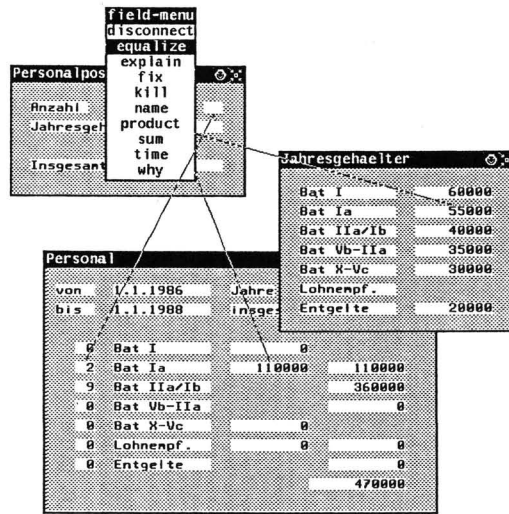
Abbildung 3-3: Das Editieren eines Formulars



Mittels Darstellungs-Kommandos werden die Felder an ihren Platz bewegt. Beim Festlegen der Größe und Position wird vom System eine Rasterung vorgesehen, so daß sich Felder entlang senkrechter und waagrechter Linien ausrichten.

Durch Auswahl aus einem Menü werden die Felder des neuen Formulars inhaltlich mit Feldern der beiden anderen Formulare gleichgesetzt (Abb. 3-4). Mit dieser Aktion etabliert der Benutzer die *Gleichheitsbeziehung* zwischen Feldern. Veränderungen in einem dieser Felder wirken sich auf alle Felder aus, die mit dem modifizierten Feld in der Gleichheitsbeziehung stehen. Im Unterschied zu reinen Darstellungsoperationen wird mit der Spezifikation der Gleichheitsbeziehung die *interne Repräsentation* modifiziert. Der Benutzer *programmiert* das Verhalten des Systems, indem er *deklarativ* eine Abhängigkeit zwischen Feldern festlegt.

Abbildung 3-4: Das Gleichsetzen von Feldern



Zwei Felder werden als Felder gleichen Inhalts spezifiziert, indem aus dem Operationsmenü für Felder der *equalize*-Eintrag ausgewählt wird. Durch zeigen mit der Maus werden die Felder identifiziert. Das System visualisiert die Beziehung durch eine Linie.

Mit dem Einrichten von Beziehungen zwischen Feldern bewirkt der Benutzer intern die Kreation von Strukturen der Implementationsprache ObjTalk, die diese Beziehungen realisieren. Ihre prozeduralen Komponenten sorgen für die Anpassung gleichgesetzter Felder. Erklärungskomponenten nutzen die neue Datenstruktur zur Generierung von Hilfetexten. Insgesamt werden durch die an der Benutzerschnittstelle einfach auszuführende Operation des Gleichsetzens von Feldern reichhaltige ObjTalk-Strukturen erzeugt, die das Systemverhalten insgesamt modifizieren.

Durch Menüauswahl und Identifikation von Formularfeldern mit der Maus etabliert der Benutzer eine Produktbeziehung zwischen drei Feldern des Formulars. In dem Formular wird dadurch festgelegt, wie sich das Jahresgehalt auf die 12 Monate eines Jahres verteilt (Abb. 3-5). Wie bei der Definition der Gleichheitsbeziehung hat der Benutzer durch die deklarative Angabe einer Beziehung neues Systemverhalten programmiert. Diese Beziehung wird zwischen Werten der internen Repräsentation etabliert. Elemente der externen Repräsentation bleiben davon unberührt.

Abbildung 3-5: Die Spezifikation der Produktbeziehung

The screenshot displays the dBase III database environment. A 'field-menu' window is open over the 'Personal' table, showing options: disconnect, equalize, explain, fix, kill, name, and product. The 'Personal' table has columns: Anzahl, Jahresg, Ingesamt, and a calculated field 'Produkt' (Anzahl * Jahresg). The 'Jahresgehaelter' table has columns: Bat I, Bat Ia, Bat Iia/Ib, Bat Vb-IIa, Lohnempf., and Entgelte. A 'Jahr' window is also open, showing the year 1986 and the month 12.

4. Auswirkungen der Modifikationen

Die vom Anwender neu geschaffene Funktionalität steht gleichberechtigt neben den bereits existierenden, d.h. sie kann zum Ausgangspunkt weiterer Manipulationen der internen und externen Repräsentation verwendet werden.

Der Benutzer editiert die Summe der Ausgaben für zwei Personalposten (Abb. 4-1). Ohne zusätzliches Wissen kann weder FINANZ noch eine Person entscheiden, welcher der beiden Summanden und, abhängig davon, welcher der beiden Faktoren des neu eingeführten Produkts angepaßt werden müssen. Die Veränderung der Summe führt das System in einen unentscheidbaren Konflikt, der vom Benutzer aufgelöst werden muß.

Abbildung 4-1: Das Verändern eines abhängigen Feldes

The screenshot displays four overlapping windows from a legacy software application:

- Jahresgehaelter**: A window titled 'Jahresgehaelter' containing a list of personnel and their annual salaries. Visible entries include Bat I (72000), Bat Ia (55000), Bat IIa/Ib (40000), Bat Vb-IIa (35000), and Bat IIc (30000).
- Personalposten/Bat I**: A window titled 'Personalposten/Bat I' showing details for a specific personnel. It includes fields for 'Anzahl' (1) and 'Jahresgehalt' (72000), with a total of 72000.
- Personal**: The largest window, titled 'Personal', showing a list of personnel with their salaries. It includes fields for 'von' (1.1.1986), 'bis' (1.1.1988), 'Jahre' (2), and 'Insgesamt' (1084000). The list includes Bat I (72000), Bat Ia (110000), Bat IIa/Ib (360000), Bat Vb-IIa (0), Bat X-Vc (0), Lohnempf. (0), and Entgelte (0). The total at the bottom is 542000.
- Monate**: A window titled 'Monate' showing monthly salary details. It includes fields for 'Bat I', 'In Jahr' (1986), 'Monate' (12), and 'pro Monat' (6000).

Die Summe der Ausgaben für zwei Personalpostens wird verändert.

Durch Inversdarstellung werden die ursächlich am Konflikt beteiligten Felder angezeigt. Der Benutzer wird aufgefordert, einen der Werte zurückzunehmen (Abb. 4-2). Die für einen Konflikt ursächlich verantwortlichen Benutzereingaben sind nicht in jedem Fall die unmittelbar an der Produktbeziehung beteiligten Felder. Oft sind, wie in diesem Fall, Feldinhalte als Folge der Modifikation anderer Felder in anderen Formularen entstanden, die über benutzerdefinierte Beziehungen miteinander verbunden sind.

Mit der Einführungen von Beziehungen zwischen Felder werden also nicht nur Interferenzprozesse etabliert, sondern auch die Möglichkeit geschaffen, Berechnungen auf

Abbildung 4-2: Die Anzeige eines Konflikts

Ausgaben
 Welcher Wert soll zurückgenommen werden?

Jahresgehalter

Bat I	72000
Bat Ia	55000
Bat IIa/Ib	40000
Bat Vb-IIa	35000
Bat Vc	30000

Personalposten/Bat I

Anzahl	1
Jahresgehalt	72000
Gesamt	72000

Personal

von	1.1.1986	Jahre	2
bis	1.1.1988	Insgesamt	1084000
1 Bat I 72000			
2 Bat Ia 110000 200000			
9 Bat IIa/Ib 360000			
0 Bat Vb-IIa 0			
0 Bat X-Uc 0			
0 Lohnempf. 0			
0 Entgelte 0			
542000			

Monate

Bat I	
In Jahr	1986
Monate	12
pro Monat	6000

Ein vom System nicht auflösbarer Konflikt wird dem Benutzer angezeigt. Der Benutzer hat in eines der Felder einen Betrag eingegeben, der im Widerspruch zu früheren Eingaben und zum Wissen des Systems über die Projektfinanzierung steht. Widersprüche sind oft nicht lokaler Natur: Ihre Ursachen können an entfernter Stelle, z.B. in einem anderen als dem gerade bearbeitenden Formlar liegen. Die ursächlich am Konflikt beteiligten Felder werden durch Inversdarstellung hervorgehoben.

ihre Ursachen zurückzuverfolgen. Das System hat Wissen darüber, welche Feldinhalte selbst Ergebnis von Berechnung sind und wo der Benutzer Veränderungen vorgenommen hat. Dieses Wissen wird abhängig vom Dialog modifiziert. Da nicht *Rechenvorschriften* sondern *Feldabhängigkeiten* definiert werden, ist die Richtung der Berechnungen nicht statisch festgelegt.

Der Benutzer hinterfragt das Zustandekommen einzelner Feldinhalte (Abb. 4-3). In dieser Situation wird das *symbiotische* Verhalten (McCracken, 1979; Fischer, 1983) zwischen System und Benutzer deutlich: FINANZ ist zwar in der Lage, komplexe Abhängigkeitsstrukturen zu durchsuchen und Ergebnisse auf ihre Ursache zurückzuverfolgen. Ohne Wissen darüber, daß Personen üblicherweise 12 Monate im Jahr arbeiten, kann FINANZ diesen Konflikt jedoch nicht auflösen. Der Benutzer erhält daher die Kontrolle über das System zurück. FINANZ steht im wieder als Werkzeug zur Verfügung, dessen interne Abhängigkeitsstrukturen sichtbar gemacht werden können.

Abbildung 4-3: Das Hinterfragen von Feldinhalten

Ausgaben

Der Wert 72000 in 'Ausgaben Bat I', wurde so errechnet:
 'Ausgaben Bat I' = (('Monate Bat I' * 'pro Monat Bat I') * 'Personen Bat I')
 'Monate Bat I' = 12
 'pro Monat Bat I' = 6000
 'Personen Bat I' = 1

Jahresgehaelter

Bat I	72000
Bat Ia	55000
Bat Iia/Ib	40000
Bat Ua-Iia	35000

Personalposten/Bat I

Anzahl	1
Jahresgehalt	72000

Personal

von	1.1.19	disconnect	re	2
bis	1.1.19	equalize	gesamt	1004000
1	Bat I	fix	72000	
2	Bat Ia	name	110000	200000
9	Bat II	product		360000
0	Bat Ua	sum		0
0	Bat H	time		0
0	Bat H	why		0
0	Lohnempf.			0
0	Entgelte			0
				542000

Monate

Bat I	
In Jahr	1986
Monate	12
pro Monat	6000

Der Benutzer hinterfragt das Zustandekommen von Feldinhalten. Er kann dazu für jedes Feld den **explain**-Eintrag aus dem Feldmenü selektieren. In dieser Situation, in der das System einen Konflikt nicht lösen kann, steht es dem Benutzer zu verschiedenen Hilfeleistungen zur Verfügung.

Im Anschluß an den Klärungsdialog nimmt der Benutzer den Monatsbetrag für den Personalposten der Gehaltsklasse **Bat I** durch Anzeige mit der Maus zurück. Dadurch kann das System den Konflikt auflösen und die von der Veränderung betroffenen Felder modifizieren.

Nach Auflösen des Konfliktes verankert der Benutzer die Zahl 12 als Anzahl der Monate pro Jahr (Abb. 4-4). Das Wissen des Benutzers, auf Grund dessen er den Konflikt auflösen konnte, wird FINANZ dadurch bekannt gemacht, daß die Anzahl der Monate pro Jahr als konstant deklariert wird. Mit dieser Information ist FINANZ zukünftig in der Lage, ähnlich Konfliktfälle automatisch zu lösen.

Mit der Festlegung des Monatsfeldes hat der Benutzer Kontrolle an das System abgegeben. Letztendlich wird dadurch seine Entscheidungsfreiheit nicht eingeschränkt. Diese Form der Aufgabendelegation kann jederzeit wieder zurückgenommen werden. In den erklärenden Texten über das Zustandekommen von Feldinhalten wird

Abbildung 4-4: Das Festlegen eines Feldes

Jahresgehaelter

Bat. I	90000
Bat. Ia	55000
Bat. IIa/Ib	40000
Bat. Vb-IIa	35000
Bat. Vb-IIb	30000

Personalposten/Bat. I

Anzahl: 1
Jahresgehalt: 90000

Personal

von: 1.1.1986 Jahre: 2
bis: 1.1.1988 insgesamt: 112000

1	Bat. I	90000	
2	Bat. Ia	110000	200000
9	Bat. IIa/Ib		360000
0	Bat. Vb-IIa		0
0	Bat. H-Vc		0
0	Lohnenpf.		0
0	Entgelte		0
			560000

Monate

Bat. I
In Jahr
Monate
pro Mona

field-menu

disconnect
equalize
explain
fix
kill
name
product
sum
time
why

1986
12
7500

die zusätzliche Systemleistung reflektiert.

5. Schlußbemerkungen

Das beispielhafte Erstellen eines Formulars, die Definition einer neuen Sicht auf interne Strukturen und die Definition neuer Beziehungen zwischen Feldern machen die Prinzipien ObjTalk-basierter Anwendungen deutlich:

1. Durch Trennung von interner und externer Repräsentation können mehrere Sichten auf dieselben, internen Strukturen erzeugt werden.
2. Interne und externe Repräsentation sind über deklarativ beschriebene Beziehungen verbunden. Änderungen der internen Repräsentation kann an der Systemoberfläche unterschiedlich dargestellt werden. Änderungen an verschiedenen Teilen der Benutzerschnittstelle kann sich auf dieselben internen Strukturen auswirken.
3. Kreieren von Formularen und Formularfeldern wird durch *Kopieren und Editieren* (Lenat, Prakash, Shepherd, 1986) realisiert. Dies hat eine Reihe von Vorteilen:
 - Komplexe Eigenschaften der internen und externen Repräsentation werden übernommen, ohne daß sie explizit vom Benutzer spezifiziert werden müßten.
 - Für das Erstellen von Feldern und Formularen ist die Kenntnis einer Spezifikationssprache nicht erforderlich. Die Definition erfolgt interaktiv und durch direkte Manipulation (Hutchins, Hollan, Norman, 1986) von Objekten

des Bildschirms.

- Die Konsistenz der Interaktionsform und der Objekteigenschaften wird aufrechterhalten.
 - Der Prozeß des Kopierens und Editierens wird durch ObjTalk-Strukturen unterstützt.
4. Durch Auswahl und Anwendung von Beziehungen zwischen Formularfeldern werden reichhaltige, interne Strukturen aufgebaut:
- ObjTalk-Objekte mit einer Vielzahl von ererbter Information werden instantiiert.
 - Inferenzprozesse werden zwischen Formularfeldern etabliert.
 - Informationsstrukturen zur Repräsentation von Begründungen für Feldinhalte werden eingerichtet. Dadurch können Werte auf ihre Ursachen zurückgeführt und Erklärungen für den Benutzer dynamisch erzeugt werden.
 - Der Benutzer beschreibt nicht mehr das *Wie* der vom System zur Verfügung gestellten Funktionalität, sondern nur noch das *Was*. Dies vollzieht sich auf einer höheren Abstraktionsebene und ist näher an den Problemen des Benutzers.
5. Die Aufteilung der Aufgaben und der Kontrolle liegt beim Benutzer. Er entscheidet, welche Arbeiten vom System übernommen werden sollen.

Mit FINANZ kann also eine *formularbasierte* Benutzerschnittstelle an Aufgabe und Benutzer angepaßt werden. Sowohl *Aussehen* wie *Struktur* der Formulare werden *interaktiv* gestaltet. Die so gestaltete Benutzerschnittstelle vermag den Benutzer bei seiner Aufgabe wirkungsvoll zu unterstützen.

FINANZ erlaubt die prototypische Entwicklung eines wissensbasierten Systems, das z.B. einen Fachmann im Bereich der Projektfinanzierung oder der Steuergesetzgebung bei der Konzeption und der Bearbeitung von Formularen unterstützt. FINANZ dient dabei als Bindeglied zwischen den Konzepten der Repräsentationssprache - also ObjTalk - und den Konzepten des Anwendungsgebiets. Die Beschränkung auf formularorientierte Anwendungen erlaubt eine wirkungsvolle Unterstützung des Benutzers während der Entwurfsphase, so daß FINANZ von Personen benutzt werden kann, die keine Experten im Bezug auf die ObjTalk-Programmierung sind.

6. Literatur

- G. Fischer: *"Symbiotic, Knowledge-Based Computer Support Systems"*. *Automatica* 19(6), pp 627-637, November, 1983.
- G. Fischer, R. Gunzenhaeuser (editors): *"Methoden und Werkzeuge zur Realisierung menschengerechter Computersysteme"*. Walter de Gruyter, Berlin - New York, 1986.

- E.L. Hutchins, J.D. Hollan, D.A. Norman: *"Direct Manipulation Interfaces". User Centered System Design, New Perspectives on Human-Computer Interaction.* Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1986, pp 87-124, chapter 5
- D. Lenat, M. Prakash, M. Shepherd: *"CYC: Using Commons Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks". AI Magazine* 6(4), pp 65-85, Winter, 1986.
- D. McCracken: *"Man + Computer: A new Symbiosis". Communications of the ACM* 22(11), pp 587-588, 1979.
- C. Rathke: *"Wissensbasierte Systeme: Mehr als eine attraktive Bildschirmgestaltung". Computer Magazin* 12(3), pp 40-41, March, 1983.
- C. Rathke: *"Objektorientierte Wissenspraesentation (Object-Oriented Knowledge Representation)". In G. Fischer, R. Gunzenhaeuser (editor), Methoden und Werkzeuge zur Gestaltung benutzergerechter Computersysteme. Verlag Walter de Gruyter & Co., Berlin - New York, 1986. Chapter 3.*

Dr. Christian Rathke
 Department of Computer Science und Institute of Cognitive Science
 University of Colorado, Campus Box 430
 Boulder, CO 80309, USA
 CS-Net: chr@boulder, Tel.: (303) 492-7135