Ein System zur interaktiven Bearbeitung umfangreicher Meßdaten

Ulrich Schauer, IBM Deutschland GmbH, Wiss. Zentrum Heidelberg

Zusammenfassung

Bei der Bearbeitung von Meßdaten muß man unterscheiden zwischen einer Standardauswertung der Messungen, bei der eine bestimmte Modellvorstellung zugrunde liegt und einer Analyse mit dem Ziel, logische Zusammenhänge zu erkennen und ein erklärendes Modell zu finden. Während die Standardauswertung durchaus im Stapelbetrieb ablaufen kann mit einem Datenmodell, das abgestimmt ist auf die im Modell ablesbaren Verknüpfungsmöglichkeiten, ist für die Analyse ein interaktives System wünschenswert mit einem Datenmodell, das beliebige Verknüpfungen ermöglicht und mit einer Datenmanipulationssprache, die möglichst deskriptiv sein sollte, aber komplexe Auswahlkriterien erlaubt. Verfügbare Systeme werden den Anforderungen der Analyse nur teilweise gerecht, meist mangelt es der Datenmanipulationssprache an Fähigkeiten zur rechnerischen Datenbearbeitung.

Im folgenden wird ein experimentelles System für die Bearbeitung von Meßdaten beschrieben, an dem im Wissenschaftlichen Zentrum der IBM in Heidelberg gearbeitet wird.

1. EINFÜHRUNG

Umfangreiche Sammlungen von Meßdaten können erst in vollem Maße nutzbar gemacht werden, wenn die für die Analyse zuständigen Fachleute (z. B. Wissenschaftler, Techniker - meist ohne große Programmiererfahrung) in die Lage versetzt werden, ohne Zuhilfenahme von Programmierern selbst die Bearbeitung vorzunehmen. Dazu ist ein interaktives System erforderlich, das erlaubt, Teilmengen der Daten unter komplexen Auswahlkriterien zu bilden und in vorhandene oder neu zu schreibende Bearbeitungsprogramme zu stecken und die Ergebnisse tabellarisch oder graphisch darzustellen.

Schon bei den Auswahlkriterien können recht verwickelte Berechnungen anfallen, die zweckmäßig mit Bausteinen aus einer Programmbibliothek durchgeführt werden. Anpassung des Systems an bestimmte Fachgebiete ist damit möglich durch Anpassung der zugrundeliegenden Programmbibliothek.

Da nur eine begrenzte Anzahl von vorgefertigten Programmen zur Verfügung stehen kann, wird häufig noch Datenmanipulation durch eine Trägersprache (host language) notwendig sein. Als Trägersprache ist APL für die angestrebte Zielsetzung besonders geeignet durch ein hohes Maß an Interaktivität, durch Anpassungsfähigkeit an die Programmiererfahrung des Benützers und eine Vielzahl von Operationen zur Datenmanipulation.

Figur 1 vermittelt einen Überblick über den Systemaufbau.

FIGUR 1: System-Aufbau

Die Datenbank enthält sowohl Problemdaten als auch beschreibende Daten. Programmbibliothek steht symbolisch für eine Sammlung von Programmen, die in PL/1, FORTRAN oder Assembler geschrieben sein können und die von APL aus mit Daten aus dem APL-Arbeitsspeicher oder der Datenbank angestoßen werden können und ihre Ergebnisse wieder im APL-Arbeitsspeicher abliefern. Die Benutzer-Kommunikation erfolgt mit APL oder mit einem der in APL eingebetteten Systeme zur Manipulation von Meßdaten, Pro-

grammen und zugehöriger Dokumentation. Als Benützerstation (Terminal) kommen in erster Linie Bildschirm und Schreibmaschine in Frage.

Einen Überblick über die Datenkomponenten, die vom System zu verwalten sind, gibt Figur 2.

FIGUR 2: Datenkomponenten

Das System muß drei in Wechselbeziehung stehende Klassen von Daten verwalten:

- a) Algorithmen (Programmbibliothek)
- b) Problem Daten (Meßdaten)
- c) Beschreibende Daten (Katalog der Daten und der Programme)

Im allgemeinen wird der Benützer bei einem konkreten Bearbeitungsfall erst anhand der Kataloginformation feststellen, aus welchen Datenaggregaten (Tabellen) seine Problem Daten auszuwählen sind und welche Programme bei der Bearbeitung eingesetzt werden können, und erst dann die vollständige Problemlösung festlegen mit Hilfe des Datenmanipulationssystems.

2. ÜBERBLICK ÜBER DIE SYSTEMKOMPONENTEN

2.1 Algorithmen

Die verfügbaren Hilfsmittel zur Bearbeitung der Meßdaten und zur Darstellung von Resultaten und Zusammenhängen zwischen den Daten lassen sich in drei Klassen einteilen:

- a) Arithmetische und logische Operationen zum Ausdrücken von logischen Beziehungen (z. B. $a > 5$) und zur Datenmanipulation (z. B. $x \leftarrow y \div z - 100$) für numerische und abgesehen von arithmetischen

Operationen auch für nicht numerische Daten. Die Verwendung von APL als Trägersprache erlaubt insbesondere auch bequeme Manipulation von Rechtecksstrukturen von numerischen und von Textdaten (Vektoren, Matrizen).

- b) Unterprogramme zur Lösung von standardisierten Problemen aus Gebieten wie Mathematik (z. B. numerische Integration und Differentiation) und Statistik (z. B. lineare Regression, Testverfahren, Darstellung von Häufigkeitsverteilungen etc.).
- c) Anwendungsbezogene Standardverfahren (z. B. Analyse von EKG-Aufzeichnungen, Klassifizierung von Fingerabdrücken etc.).

Die Trägersprache APL mit einer Vielzahl von verfügbaren APL-Bibliotheksprogrammen und der Möglichkeit, von APL aus graphische Darstellungen zu initiieren, bietet schon alle Möglichkeiten zur Datenmanipulation. Trotzdem sind die Klassen b) und c) notwendige Bestandteile des Systems. Die Klasse b) erlaubt Ausweichen auf FORTRAN, PL/1 oder Assembler geschriebene Unterprogramme, was besonders bei großen Datenmengen bessere Rechenzeiten bringen kann. Programme der Klasse c) existieren vorwiegend in FORTRAN oder PL/1, weil sie meistens für Anwendung im Stapelbereich entwickelt werden.

2.2 Problemdaten

Das System benützt ein relationales Datenmodell; die Datenbank besteht aus einer Sammlung umfangreicher Tabellen, die mit leicht verständlichen Operationen manipuliert werden können (Codd /1,2,3/). Datenattribute sind den Spalten einer Tabelle fest zugeordnet wie beim SEQUEL-System (Boyce, Chamberlin /4,5/). Spezifikation von Teilmengen von Daten aus einer oder mehreren Tabellen erfolgt mit einer an Beispieleintragungen in die fraglichen Tabellen orientierten deskriptiven Sprache, die sich gleichermaßen für den Einbau von Unterprogrammaufrufen in den Programmablauf eignet (Zloof /6/).

Die Datenelemente in einer Tabellenspalte können dimensionierte Daten sein (z. B. Vektoren, die eine Meßreihe darstellen oder Matrizen, die mehrere Meßreihen oder eine Funktion von zwei Veränderlichen darstellen können etc.). Die offensichtliche Mehrdeutigkeit wird durch eine der Tabellenspalte zugeordnete Interpretierung behoben.

- a) Interpretierungsattribut: Regelt die Deutung einer Matrix, z.B. als Werte einer Funktion von zwei Veränderlichen in den Punkten eines gleichabständigen Gitters. Die Definition der Gitterpunkte

$$(x_0 + i.h, y_0 + j.k) \quad i = 0, 1, \dots, m-1 \\ j = 0, 1, \dots, n-1$$

erfolgt durch Angabe von x_0 , y_0 , h , k und m , n .

- b) Darstellungsattribut: Erlaubt Spezifikation von Verdichtungsmechanismen für Datendarstellungen in Ergänzung zu beispielsweise 1, 2, 4 byte integer.
- c) Speicherungsattribut: Die meisten Daten werden in der XRM-Datenbank gespeichert (Lorie /7/). Umfangreiche Datenelemente (z. B. digitalisierte Bilder) können jedoch auch in von CMS (Conversational Monitor System) verwalteten Band- oder Platten-Dateien abgelegt und in XRM nur durch Angabe ihres Dateinamens und einer Zugriffsroutine bekannt gemacht werden.

Das System besorgt automatische Umwandlung physikalischer Einheiten und automatische Datenkonversion entsprechend Interpretierungs-, Darstellungs- und Speicherungsattribut sowie Beachtung von durch logische Bedingungen definierten Konsistenzregeln bei neuen Eintragungen oder Änderungen in einer Tabelle.

2.3 Beschreibende Daten

Das System zur Manipulation der unformatierten Kataloginformation ist eine selbständige Komponente mit Fähigkeiten für Generierung, Wartung und für rechnerunterstütztes Auffinden der relevanten Katalogeintragungen über Daten und Algorithmen (Erbe, Walch /8/). Formatierte Datenbeschreibung wird in der XRM-Datenbank gespeichert und umfaßt jeweils ein Verzeichnis von:

- a) Umwandlungstabellen für physikalische Einheiten.
- b) Methoden mit Programmidentifikation.
- c) Datenattributen mit Tabellen und Spaltenbezeichnern.

Mittels b) und c) können Programme und Tabellen rasch identifiziert werden, wenn die Bezeichnung der Methode bzw. der Attribute der fraglichen Tabellenspalte bekannt sind.

3. DIE DATENMANIPULATIONSSPRACHE

Zunächst sind zwei Sprachebenen vorgesehen.

3.1 Prozedurale Sprachebene

Die folgenden Eigenschaften kennzeichnen die prozedurale Datenmanipulation:

- a) Der Datenzugriff erfolgt durch APL-Befehle (Lorie, Symonds /9/).
- b) Umwandlungen zwischen der externen Datendarstellung in der XRM-Datenbank und der internen Datendarstellung erfolgen automatisch (z. B. Darstellung und Speicherung).
- c) Konsistenzregeln werden automatisch kontrolliert bei Datenzugängen oder Veränderungen.
- d) Die Daten werden tabellenweise oder zeilenweise verarbeitet.
- e) Der Benützer ist verantwortlich für korrektes Verarbeiten der Daten hinsichtlich physikalischer Einheiten und Interpretation.

3.2 Deskriptive Sprachebene

Die nicht prozedurale Sprache EQBE stellt eine Erweiterung dar von QBE (Query by Example, Zloof /6/). Sie eignet sich auch für Benützer mit geringen Kenntnissen in APL (Erfahrung im Umgang mit APL als Tischrechner genügt) und ohne Programmiererfahrung. Die Sprache ist in hohem Maße deskriptiv. Relationen und in der Programmbibliothek verfügbare Unterprogramme werden als Tabellen dargestellt, und der Benützer formuliert seine Datenauswahl, indem er entsprechende Zeileneintragungen vornimmt, die Ausgabewerte bezeichnet und Auswahlkriterien - soweit erforderlich - durch APL-Statements definiert. EQBE läßt sich am besten anhand von Beispielen erklären.

3.3 Beispiele zu EQBE

R	R1	R2
r	x	y

ist ein Schema für eine Tabelle mit dem Namen R und zwei Spalten mit den Bezeichnern R1 und R2.

Die Werte x, y stellen eine Tabellenzeile dar, r ist ein Bezeichner für diese Zeile.

r, x, y werden vom Benützer eingetragen in das Schema

R	R1	R2
---	----	----

, das vom System geliefert wird, wenn man Tabelle R anfordert. Die Datenvariablen x, y können alle in R gespeicherten Tupelwerte annehmen.

$$\{(x, y) \mid (x, y) \in R\}$$

1. Auswahl einer Spalte (Projektion)

R	R1	R2
u	x	

$\square \leftarrow x$

Die Angabe eines Zeilenbezeichners ist nicht notwendig.

$\square$ ist als Symbol für Ausgabe zu verstehen.

Die Abfrage lautet:

Gesucht ist die Menge der x Werte aus R1.

Eine mögliche Formulierung im Prädikatenkalkül wäre

$$\{x \mid \exists_y (x, y) \in R\}$$

Selbstverständlich erstreckt sich der Definitionsbereich von y nur auf Werte aus der R2-Spalte von R.

Im folgenden schreiben wir dafür auch kürzer

$$\{x \mid u(x,)\}$$

und fassen $u(x,)$ als Prädikat auf, das wahr ist, wenn ein Tupel in R existiert, dessen erste Komponente gleich x ist.

2. Einfache Abfrage mit einschränkenden Bedingungen, die in der Trägersprache formuliert werden.

R	R1	R2	R3
u	x	y	z

$$x > 5 + y \times y$$

$$(z < 25) \vee (z > 50)$$

$$\square \leftarrow x$$

$$\{x \mid \exists_y \exists_z u(x,y,z) \wedge (x > 5 + y \times y) \wedge ((z < 25) \vee z > 50) \}$$

3. Schnittmenge

R	R1	R2
r	x	y

S	S1	S2
s	x	z

$$x > y$$

$$z = 10$$

$$\square \leftarrow x$$

Trägt man in S anstatt z den konstanten Wert 10 ein, so entfällt das APL-Statement $z = 10$.

$$\{x \mid \exists_y r(x,y) \wedge (x > y)\} \cap \{x \mid \exists_z s(x,z) \wedge (z=10)\}$$

oder

$$\{x \mid \exists_y \exists_z r(x,y) \wedge s(x,z) \wedge (x > y) \wedge (z=10)\}$$

4. Vereinigungsmenge

R	R1	R2	R3
u	x1	y	
v	x2		z
w	x:x1		
w	x:x2		

$$x1 > y$$

$$z = 10$$

$$\square \leftarrow x$$

$$\{ x \mid \exists y \, u(x,y) \wedge (x > y) \} \cup \{ x \mid \exists z \, v(x,z) \wedge (z=10) \}$$

oder

$$\{ x \mid (\exists y \, u(x,y) \wedge (x > y)) \vee \exists z \, v(x,z) \wedge (z=10) \}$$

5. Differenzmenge

R	R1	R2
r	x	y

S	S1	S2
$\sim s$		x

$$\square \leftarrow x$$

$$\{ x \mid \exists y \, r(x,y) \wedge \sim s(x) \}$$

Selbstverständlich muß jede Datenvariable, die in einer negierten Tupelvariable auftritt, auch in einer nicht negierten Tupelvariablen auftreten (oder als globale Variable bekannt sein).

6. Kartesisches Produkt

R	R1	R2
r	x	y

S	S1	S2
s	x1	z

$$\square \leftarrow x,y,x1,z$$

$$\{ (x,y,x1,z) \mid r(x,y) \wedge s(x1,z) \}$$

7. Equijoin (Restriktion im Kartesischen Produkt)

R	R1	R2
r	x	y

S	S1	S2
s	x	z

$$\square \leftarrow x,y,z$$

$$\{ (x,y,z) \mid r(x,y) \wedge s(x,z) \}$$

8. Verallgemeinerter Join mit nachfolgender Projektion

R	R1	R2
r	x	y

S	S1	S2
s	x1	z

$$x \geq y$$

$$\square \leftarrow z$$

$$\{z \mid \exists x \exists x_1 \exists y \ r(x,y) \wedge s(x_1,z) \wedge (x \geq y)\}$$

Anstelle des $\geq$ Operators könnte eine beliebige Boolesche Funktion stehen.

9. Division

R	R1	R2
r	x	y

S	S1	S2
s	x	z

T	T1	T2
t	.y	z

$$\square \leftarrow x$$

$$\{x \mid \exists z \forall y \in r \ r(x,y) \wedge s(x,z) \wedge t(y,z)\}$$

.y steht für $\{y \mid \exists x \exists z \ r(x,y) \wedge s(x,z)\}$,

wobei $\exists x$ bedeuten soll, daß x fest zu wählen ist, und das Auftreten von .y in t ist so zu verstehen, daß gilt $\forall y \in .y \ t(y,z)$

10. Gruppierung

$$\{x \mid \forall y \exists z \ r(x,y) \wedge s(x,z) \wedge t(y,z)\}$$

kann bis jetzt noch nicht formuliert werden. Man braucht ein Hilfsmittel, um Abhängigkeit zwischen Variablen anzugeben. Mit der Vereinbarung, daß y.z bedeuten soll $\exists y \exists z$, sind die entsprechenden Eintragungen:

R	R1	R2
r	x	y

S	S1	S2
s	x	z

T	T1	T2
t	.y	y.z

$$\square \leftarrow x$$

Wir sind jetzt in der Lage, jede Operation der Relationalalgebra auszuführen. Die Vollständigkeit von QBE in der vorgestellten erweiterten Form ist damit für einfache Abfragen, die nur eine Operation der Relationalalgebra umfassen, erwiesen.

Sie folgt auch für beliebig zusammengesetzte Operationen: Jede Abfrage von QBE etabliert bei ihrer Definition eine logische Datensicht, die der Resultattabelle entspricht. Erst bei Ausführung eines APL-Programmes, das von einem Abfrageprozessor aus der logischen Datensicht erzeugt wird,

entsteht die Resultattabelle. Eine neue Abfrage kann auf der logischen Datensicht von schon definierten Abfragen aufgebaut werden, und damit kann eine komplexe Abfrage in Einzelschritte aufgelöst werden.

3.4 Diskussion der Erweiterungen von QBE

Die nachfolgend beschriebenen Erweiterungen erlauben die Behandlung von recht komplexen Abfragen, wie sie bei Meßdaten zu erwarten sind, ohne die Einfachheit für elementare Abfragen zu beeinträchtigen.

- a) In einer Programmbibliothek erfaßte Algorithmen (APL-Funktionen, FORTRAN-Unterprogramme, PL/1-Prozeduren oder Assemblerroutrinen) können für Datenauswertung oder Datenselektion innerhalb einer Abfrage eingesetzt werden.
- b) Beliebige APL-Befehle können innerhalb einer Abfrage zur Daten-selektion und Auswertung verwendet werden. QBE erlaubt außer den Vergleichsoperationen nur eine begrenzte Anzahl eingebauter Funktionen wie COUNT, SUM etc.
- c) Die Resultattabelle einer Abfrage kann durch Angabe von format-beschreibenden Formularen auf verschiedenste Art dargestellt werden, auch in graphischer Form und wiederholt mit wechselnden Formularen.
- d) Durch jede Abfrage wird eine logische Datensicht definiert, die zur Entkoppelung komplexer Abfragen in einer Folge von einfacheren Abfragen verwendet werden kann.
- e) Jede Abfrage kann zu wiederholten Malen ausgeführt werden. Dabei können von Mal zu Mal die Werte globaler Variablen geändert werden. Für APL-erfahrene Benützer eröffnen sich dadurch interessante Möglichkeiten zur Datenbearbeitung mit anpassungsfähigen Bausteinen.
- f) Der Entkopplungseffekt von QBE, daß die Zeileneintragungen in beliebiger Reihenfolge möglich sind, wurde noch verstärkt (Verwendung der Gruppierungsmöglichkeit).
- g) Durch die Gruppierungsmöglichkeit können auch Abfragen ohne Zerlegung in aufeinanderfolgende Schritte bearbeitet werden, die sich der Behandlung durch QBE entziehen.
- h) Als Gegenstück des ALL D-Operators (all different) von QBE dient in EQBE ein vorgesetzter Punkt, entsprechend beim ALL-Operator (alle mit Wiederholungen) ein vorgesetzter Punkt und Angabe des Tupelbezeichners in Klammern gesetzt. Eine Pseudovariable wie .y oder .x (r) kann in APL-Befehlen verwendet werden und steht stellvertretend für einen Bereich gleichartiger Werte.

4. MESSDATENBEARBEITUNG

4.1 Das Datenbearbeitungssystem

APL ist zur interaktiven Analyse von Meßdaten, die im APL-Arbeitsspeicher Platz finden, hervorragend geeignet (Schatzoff /10/). Bei großem Datenumfang verliert APL an Attraktivität, weil Datenselektion aus Tabellen dann aus Platzgründen nicht im APL-Stil durch eine Operation über einen dimensionierten Bereich dargestellt werden kann, sondern nur durch eine Rekursionsvorschrift über alle Tabellenzeilen. Eine prozedurale Sprachebene mit APL als Trägersprache ist daher noch nicht voll zufriedenstellend.

Ein weiterer Gesichtspunkt bei Meßdaten ist, daß Messung häufig für die Zusammenfassung von vielen Einzelwerten steht (z. B. digitalisierte Meßkurve). Für die Bearbeitung solcher Messungen ist es wünschenswert von der Trägersprache APL aus, Programme, die in einer anderen Sprache (FORTRAN, PL/1, Assembler) entwickelt wurden, aufrufen zu können.

Andere experimentelle Datenbanksysteme, die APL als Trägersprache verwenden, sind meist nur für geringen Datenumfang konzipiert (Palermo /11/), Klebanoff, Lochovsky, Tsichritzis /12/) und erlauben den Einsatz von Programmen, die nicht in APL geschrieben wurden, entweder gar nicht oder nur mit ineffizienter Datenkommunikation (über externe Dateien).

Bei der in Figur 3 beschriebenen Architektur erhalten wir ein System zur Problemlösung mit

- Datenbankzugriff auf zwei Sprachebenen (prozedural und deskriptiv)
- Einsatzmöglichkeit von vorgefertigten Programmen aus einer leicht erweiterbaren Programmbibliothek (FORTRAN, PL/1 oder Assemblerprogramme)
- Hilfsmitteln zur Verwaltung der Dokumentation über Daten und Programme
- Automatischer Datenumwandlung in gewünschte physikalische Einheiten
- Automatischer Datenkonversion, soweit durch Implementierung, Darstellung und Speicherung erforderlich
- Unterstützung graphischer Ein/Ausgabegeräte
- Verfügbarkeit von Programmen zur graphischen Darstellung
- einer Schnittstelle für leichte Substitution von Ein/Ausgabegeräten

VM /370 Conversational Monitor System

FIGUR 3: Systemarchitektur

4.2 Beispiele zur Datenbearbeitung

Die folgenden zwei Beispiele sollen die Fähigkeiten zur Problemlösung illustrieren. Im ersten Beispiel wird die Verbindung mit Programmen aus einer Programmbibliothek gezeigt, im zweiten Beispiel unter anderem die Benützung von globalen Variablen.

1. Welches in der Datenbank erfaßte Material hat einen mittleren Reflexionsbeiwert (zwischen 250 und 300 nm) größer als 60?

MATERIAL- SPEKTREN	MATERIAL- NAME	REFLEXIONSSPEKTRUM	
		STARTWERT	SCHRITTWEITE
		150 NM	5 NM
	material	reflexion	

SIMPSON- REGEL	AUSGABE	EINGABE				
	INTEGRALWERT	INTEGRAND			GRENZEN	
		x_0	Δx	$y(x)$	von	bis
	integral	150	5	reflexion	x1	x2

x1 ← 250

x2 ← 300

60 < integral ÷ (x2-x1)

□ ← material

Die obigen Eintragungen in das Tabellenschema der Materialspektren und eine schematische Darstellung des Programmes SIMPSON-REGEL zusammen mit einigen APL-Befehlen definieren die Ergebnisliste der gesuchten Materialien.

Kommt als zusätzliche Bedingung hinzu, daß für das gesuchte Material das Produkt aus spezifischem Gewicht γ [kg/dm³], spezifischer Wärme c [cal/(grad.g)] und Wärmeleitfähigkeit λ [cal/(cm.grad.sec)] größer als 0.5 sein muß, so ist das obige Schema wie folgt zu ergänzen:

MATERIAL- WERTE	SPEZ. GEWICHT	SPEZ. WÄRME	WÄRME LEITFÄHIGKEIT	MATERIAL- NAME
	gamma	c	lambda	material

$$0.5 > \text{gamma} [\text{KG} \div \text{DM}^3] \times c [\text{CAL} \div \text{GRAD} \times \text{G}] \times \text{lambda} [\text{CAL} \div \text{CM} \times \text{GRAD} \times \text{SEC}]$$

Bei dieser Formulierung ist die Existenz einer Eintragung in der Tabelle MATERIALWERTE gesichert. Eine widersprechende Eintragung könnte außerdem existieren (falls MATERIALNAME nicht Schlüsseleigenschaft hat). Bei der folgenden Abänderung ist entweder die zusätzliche Bedingung erfüllt oder nicht entscheidbar (weil keine Eintragung der Materialwerte existiert):

MATERIAL- WERTE	SPEZ. GEWICHT	SPEZ. WÄRME	WÄRME- LEITFÄHIGKEIT	MATERIAL- NAME
~	gamma	c	lambda	material

$$0.5 \leq \text{gamma} [\text{KG} \div \text{DM}^3] \times c [\text{CAL} \div \text{GRAD} \times \text{G}] \times \text{lambda} [\text{CAL} \div \text{CM} \times \text{GRAD} \times \text{SEC}]$$

2. Die Datenbank möge Aufzeichnungen enthalten über die Wirkung verschiedener Behandlungsarten sowie Daten über die behandelten Personen.

Um einen ersten Überblick zu bekommen, ist eine Häufigkeitstabelle gewünscht, die den Zusammenhang zwischen Wirkung und Behandlungsart für ein bestimmtes Kollektiv von Versuchspersonen (Raucher, männlich, älter als 40 und mit Übergewicht) wiedergibt.

BEHANDLUNG	ART	WIRKUNG	PERSON
b	x	y	name

PERSON	NAME	MÄNNLICH	GRÖSSE	GEWICHT	ALTER	RAUCHER
	name	1	h	g	a	1

40 < a [JAHR]

1 < g [KG] ÷ (h[CM]-100)

I ← x(b)

J ← y(b)

P[I;J] ← P[I;J] + 1

Durch das Anhängen von (b) an x und y wird bewirkt, daß alle in b vorkommenden Paare x,y berücksichtigt werden (auch Wiederholungen). Die globale Variable P, welche mit den Werten x(b), y(b) gebildet wird, muß vor Ausführung der Abfrage initialisiert werden, z. B. durch $P \leftarrow 5 \quad 10 \quad 0$, wenn zwischen 5 Behandlungsarten und 10 Wirkungsstufen unterschieden wird.

Würden kontinuierliche Werte für Kennzeichnung von Behandlungsart und Wirkung vorliegen, so müßte noch eine Intervalleinteilung vorgegeben werden für x und y,

z.B.: $x < x_1, x_1 \leq x < x_2, \dots, x_3 \leq x < x_4, x_4 \leq x$

$y < y_1, y_1 \leq y < y_2, \dots, y_8 \leq y < y_9, y_9 \leq y$

durch Angabe der Zahlenfolgen $IX \leftarrow x_1, \dots, x_4$ und $IY \leftarrow y_1, \dots, y_9$.

IX und IY können gleichfalls als globale Variable übergeben werden. Anstelle von $I \leftarrow x(b)$ und $J \leftarrow y(b)$ tritt dann $I \leftarrow IX \text{ INDEX } x(b)$ und $J \leftarrow IY \text{ INDEX } y(b)$. Dabei ist INDEX eine APL-Funktion, die die Intervallnummer feststellt:

$\nabla I \quad IX \text{ INDEX } X$

$[1]I \leftarrow 1 + +/X \geq IX \nabla$

Durch Abänderung der globalen Variablen IX, IY, P kann eine völlig neue Resultattabelle erzeugt werden, ohne neuerliche Kompilation der Abfrage.

4.3 Einsatzmöglichkeiten

Das vorgestellte System ist in erster Linie für die Bearbeitung von Meßdaten in Wissenschaft und Technik konzipiert. Sicher finden sich auch im kommerziellen Bereich Einsatzmöglichkeiten, z. B. für interaktive Datenanalyse mit dem Ziel, Zusammenhänge zu erkennen, die sinnvolle Vorhersagen ermöglichen.

Computerunterstütztes Entwerfen (CAD) als spezielles Anwendungsgebiet dieses Systems wird am Wissenschaftlichen Zentrum Heidelberg untersucht (Kantorowitz /13/).

Wie Figur 4 zeigt, läßt sich das System - je nach Benutzerstandpunkt - charakterisieren als:

- Erweiterung von APL (Datenbankzugriff, Einsatzmöglichkeit von kompilierten Programmen, Auskunftssystem über Methoden, Programme und Daten),
- Erweiterung eines Datenbanksystems zum Problemlösungssystem (Interaktive Datenmanipulationssprache mit Einsatzmöglichkeit für eine Sammlung von Programmen, Auskunftssystem über Methoden, Programme und Daten),
- Erweiterung einer Sammlung von Programmen zum Problemlösungssystem (Datenbankzugriff, interaktive Datenmanipulationssprache, Auskunftssystem über Methoden, Programme und Daten),
- Erweiterung eines Auskunftssystems über Methoden, Programme und Daten zum Problemlösungssystem (Datenbankzugriff, interaktive Datenmanipulationssprache mit Zugriff zu Programmbibliothek).

Funktion	Komponente
Datenmanagementsystem	XRM (Extended Relational Memory)
Interaktive Träger- sprache	APL
Prozedurale Sprachebene (Interface zu XRM)	APL-Funktionen
Deskriptive Sprachebene (mit Übersetzung nach APL)	EQBE (Extended Query by Example)
Programmbibliothek	Scientific Subroutine Package (Benutzerprogramme)
Programmdokumentation	Methodenbank (in APL implementiert)
Kommandosprache für Programmausführung	EQBE
Interaktive graphische Datenmanipulation	GRAPHPAK
Sequentielle Dateien	CMS/APL
Schnittstelle APL - PL/1	Hilfsprozessor
Unterstützung von Ein/Ausgabegeräten	APL-Funktionen

IGUR 4: Komponenten zum Problemlösungssystem

Keine der oben erwähnten Komponenten stellt allein gesehen ein Novum dar, ausgenommen vielleicht das Auskunftssystem über Daten, Methoden und Programme. Im Zusammenwirken entsteht ein System zur interaktiven Bearbeitung umfangreicher Meßdaten, bei dem der Problemlöser selbst (ohne Zwischenschaltung von Programmierern) die für ihn wichtige Information aus einer Datenbank unter anwendungsspezifischen Auswahlkriterien abrufen und für gleichfalls anwendungsbezogene Berechnungen nutzbar machen kann. Bei unserem experimentellen System wird APL als Implementierungs- und Trägersprache verwendet, um entsprechend implementierte Teilsysteme leicht einfügen zu können.

Durch die Wahl der Komponenten bedingt bleiben manche Aspekte eines Datenmanagementsystems zunächst unberücksichtigt. Dennoch kann das System helfen, Erfahrung zu sammeln über die Forderungen, die für die Bearbeitung von Meßdaten an Datenbanksystem, Datenmanipulationssprache und Auskunftssystem zu stellen sind, um ein Problemlösungssystem zu erhalten, das auch für Nichtprogrammierer attraktiv ist.

Literatur

- [1] E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", CACM, Vol. 13, No. 6, June 1970, pp. 377-387
- [2] E.F. Codd, "Normalized Data Base Structure: A Brief Tutorial", Proc. 1971 ACM SIGFIDET Workshop
- [3] E.F. Codd, "Interactive support for Non-Programmers: The Relational and Network Approaches", Proc. 1974 ACM SIGFIDET Workshop
- [4] R.F. Boyce, D.D. Chamberlin, "SEQUEL: A Structured English Query Language", Proc. 1974 ACM SIGFIDET Workshop
- [5] R.F. Boyce, D.D. Chamberlin, "Using a Structured English Query Language as a Data Definition Facility", IBM Research Report RJ 1318, Dec. 1973
- [6] M.M. Zloof, "Query by Example", IBM Research Report RC 4917, July 1974
- [7] R.A. Lorie, "XRM an Extended (n-ary) Relational Memory", IBM Technical Report 320-2096, Jan. 1974
- [8] R. Erbe, G. Walch, "An Interactive Guidance System for Method Libraries", IBM Germany, Wissenschaftliches Zentrum Heidelberg, Technical Report 75.04.001, April 1975
- [9] R.A. Lorie, A.J. Symonds, "A Relational Access Method for Interactive Applications", Courant Computer Science Symposia 6, "Data Base Systems", 1971, Prentice Hall
- [10] M. Schatzoff, "Interactive Statistical Data Analysis - APL Style", IBM Technical Report 320-2079, April 1972
- [11] F.P. Palermo, "An APL Environment for Testing Relational Operators and Search Algorithms", Proc. APL 75
- [12] J. Klebanoff, F. Lochovsky, D. Tsichritzis, "Teaching Data Base Concepts Using APL", Proc. APL 75
- [13] E. Kantorowitz, "A Computer Aided Design Front End for the Measurement Data Base", IBM Germany, Wissenschaftliches Zentrum Heidelberg, Technical Note 75.07, July 1975