ECONSTOR
Make Your Publications Visible.

A Service of

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Schmidt, Reinhart; Janowski, Wolfgang

Working Paper — Digitized Version

PLASMA: Planungssystem für mathematische Anwendungen auf Dialogbasis

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 39

Provided in Cooperation with:
Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Schmidt, Reinhart; Janowski, Wolfgang (1976) : PLASMA: Planungssystem für mathematische Anwendungen auf Dialogbasis, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 39, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:
https://hdl.handle.net/10419/193890

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

INSTITUT FÜR BETRIEBSWIRTSCHAFTSLEHRE
DER CHRISTIAN-ALBRECHTS-UNIVERSITÄT KIEL

DIREKTOREN: PROFESSOR DR. KLAUS BROCKHOFF, PROFESSOR DR. REINHART SCHMIDT

Nr. 39

P L A S M A
- Planungssystem für mathematische Anwendungen auf Dialogbasis -

von
Reinhart Schmidt
und
Wolfgang Janowski

November 1976

Copyright Reinhart Schmidt und Wolfgang Janowski

1. Zur Entwicklung von Planungssystemen

Neuere Umfragen über Umfang und Art der Unternehmensplanung haben zu Ergebnissen geführt, aus denen Konsequenzen für die weitere Entwicklung von Planungssystemen gezogen werden müssen. So hat die Umfrage von Naylor und Schauland bei US-amerikanischen, kanadischen und europäischen Unternehmen ergeben, daß ein Modelleinsatz bei der Unternehmensplanung vor allem deshalb als nützlich empfunden wird, weil mehr Alternativen betrachtet werden können, die Entscheidungsqualität sich verbessert und die Planung effektiver gestaltet werden kann (2o, S. 933). Andererseits wird an Unternehmensmodellen insbesondere ausgesetzt, sie seien nicht flexibel genug, zu schlecht dokumentiert und erforderten einen zu umfangreichen Dateninput (2o, S. 934).

Der Umfrage bei deutschen Unternehmen durch Brockhoff entnimmt man als Gründe für die Unzufriedenheit mit der Unternehmensplanung (2, S. 311): zu wenig Modell- und Computereinsatz, ungenügende Ausbildung und Beratung sowie unzureichende Informationsqualität. Unzufriedenheit mit der Planung läßt die Unternehmen verstärkt die Einführung mathematischer Planungsmethoden anstreben (2, S. 312). Dies zeigt sich auch an dem weiter zunehmenden Einsatz von Methoden des Operations Research (OR), den Kirsch et al. (12, S. 44 f.) festgestellt haben. Einsatz oder Nichteinsatz von mathematischen Planungsmethoden sind aber anscheinend nicht bestimmend für die Zufriedenheit mit der Planung (2, S. 312). Damit ist erneut die Frage nach dem Nutzen von Planungsmethoden und -modellen gestellt.

Für die Gestaltung von Planungssystemen sind viele Kriterien von Bedeutung [vgl. solche Kriterien etwa in (4)]. Die zukünftige Entwicklung von Planungssystemen muß mehr Gewicht auf die Flexibilität von Modellen, Methoden und Daten als Elementen des Planungssystems legen:

1. Die Modellflexibilität ist zu erhöhen. Abwandlungen und Verbindungen von Modellen müssen leicht möglich sein. Dadurch wird die Benutzerfreundlichkeit gesteigert, wovon wiederum die Motivation zur Modellanwendung positiv beeinflußt wird.

2. Die Methodenflexibilität ist zu erhöhen. Es muß die Möglichkeit gegeben sein, von niederen auf höhere Methoden überzugehen, z.B. von der Simulation auf die Optimierung. Zwischen verschiedenen Algorithmen innerhalb einer Methode muß eine Wahlmöglichkeit bestehen.
3. Die Datenflexibilität ist zu erhöhen. Die Dateneingabe muß erleichtert werden, und es müssen unterschiedliche Methoden der Datengewinnung, insbesondere der Prognose, Verwendung finden können.

Die sich auf Modelle, Methoden und Daten beziehende Planungsflexibilität hängt wesentlich von der Art des Computereinsatzes ab. Dabei tritt unter dem Flexibilitätsaspekt die Gestaltung der Mensch-Maschine-Kommunikation in den Vordergrund. Bei Planungsflexibilität läßt sich die in der Literatur (17, S. 24) getroffene Unterscheidung zwischen einem programmkontrollierten Dialog ("Sklavensystem") und einem benutzerkontrollierten Dialog ("Herrensystem") nicht mehr eindeutig treffen. So erfordert die Modellflexibilität eine starke Benutzerkontrolle, die nach Entscheidung für eine bestimmte Methode der Modellauswertung in einen programmkontrollierten Dialog übergehen kann. Dies zeigt nur zu deutlich, daß es um eine Partnerschaft zwischen Mensch und Maschine geht, wobei der Trend zum Computer als intelligentem Partner des Menschen nicht zu verkennen ist (11, S. 146 ff.). Damit scheint die Auffassung von Hellmich (7, S. 249), letztlich sei der Mensch das Maß der Flexibilität des Planungssystems, nicht generell gültig zu sein.

Das von den Verfassern entwickelte Programmsystem PLASMA strebt als Planungssystem für mathematische Anwendungen die Flexibilität im Bereich von Modellen, Methoden und Daten an. Im folgenden werden Grundgedanken und Arbeitsweise eines solchen Systems dargestellt. Wert gelegt wird im Gegensatz zu Winand (26) auch auf die praktische Realisation des Systems und seine schrittweise Weiterentwicklung.

2. Aufbau und Arbeitsweise von PLASMA

2.1. Die Aufgabenstellung

PLASMA ist aus der Erfahrung entstanden, daß die Praxis eine Scheu bei der Einführung wissenschaftlich begründeter Planungsmodelle und -methoden zeigt. Der Betriebswirtschaftslehre gelingt es bisher wegen abstrakter Modellformulierungen nur schwer, die Praxis von Anwendbarkeit und Nutzen bestimmter Modelle und Methoden zu überzeugen. Daher muß sich die betriebswirtschaftliche Forschung verstärkt um Nachweise der Anwendbarkeit und Machbarkeit kümmern. PLASMA will solche Nachweise erbringen helfen und unterstützt damit zunächst das Eigeninteresse der Forschung nach Verwertung ihrer Ergebnisse.

Eine zweite Aufgabe von PLASMA ist die Verbesserung des betriebswirtschaftlichen Forschungsprozesses. Durch Modell-, Methoden- und Datenflexibilität eines Programmsystems sind Rationalisierungseffekte für die angewandte Forschung zu erwarten; denn wünschenswerte Vergleiche alternativer Modellstrukturen und Methoden sind bisher auf große Schwierigkeiten gestoßen. Außerdem ist eine Beschleunigung des Forschungsprozesses durch Einsatz eines solchen Programmsystems zu erwarten. Zur Erfüllung dieser Zwecke ist besonders auf Erweiterungsfähigkeit des Systems in modell- und methodenmäßiger Hinsicht zu achten.

Wenn mit Hilfe des Programmsystems Anwendbarkeit und Machbarkeit von Konzepten demonstriert werden können, so kann dem System PLASMA zugleich eine dritte Aufgabe gestellt werden: Software-Unterstützung für Planungsprobleme in der Praxis zu geben. Diese Aufgabe führt zu einer Betonung der Benutzerfreundlichkeit, der Programmkompatibilität und der Fähigkeit, auch große Planungsprobleme praktisch lösen zu können.

Eine vierte Aufgabe von PLASMA wird bei Erfüllung der ersten beiden Aufgaben lösbar: die Fortentwicklung des computergestützten Unterrichts im Bereich der Planungsmodelle und -methoden. Es wird angestrebt, daß Dozent und Teilnehmer einer Lehrveranstaltung OR-Modelle während einer Veranstaltung

nicht nur formulieren, sondern auch lösen. In diesem Fall sind insbesondere die Dialogstruktur, die Verarbeitungsgeschwindigkeit und die Dokumentation der einzelnen Schritte von Bedeutung. In dieser Hinsicht ist auch eine Verbindung mit bestehenden Systemen [z.B. (21)] denkbar.

Es wäre nicht nur vermessen, sondern falsch, die vorliegende Version von PLASMA als endgültig anzusehen. Dies ergibt sich aus dem Flexibilitätsaspekt, der prinzipiell eine Integration anderer existierender oder neu aufkommender Modelle und Methoden erfordert. Insofern handelt es sich bei der jetzigen Version eher um den Prototyp von PLASMA.

2.2. Die Systemstruktur

PLASMA ist ein in FORTRAN IV geschriebenes Programmpaket, welches eine Integration von drei Bereichen ermöglicht:

- Modellbank
- Methodenbank
- Datenbank

In den Abbildungen 1 bis 3 ist der Aufbau dieser Bereiche mit Hilfe der Dezimalklassifikation geordnet worden. Die ersten beiden Stellen der Klassifikation dienen nur der sachlogischen Strukturierung, wogegen die dritte Stelle zusätzlich die Existenz einer entsprechenden Datei auf externen Speichern anzeigt.

2.2.1. Die Modellbank

Unter Modell verstehen wir die Abbildung der Realität, wobei diese Abbildung in PLASMA in der jetzigen Ausbaustufe in Form von Gleichungen oder Ungleichungen erfolgt. Die Modellbank enthält eine oder mehrere Modelldateien, die unter ihrem Dateinamen aufgerufen werden können. Jede Modelldatei wiederum enthält ein oder mehrere Modelle, die unter ihrem Modellnamen aufgerufen werden können. Ein Modell besteht aus einem oder mehreren Modellsätzen. Jeder Modellsatz repräsentiert eine Gleichung oder Ungleichung, es können bei Verwendung der mathematischen Kurzsprache auch mehrere Gleichungen bzw. Ungleichungen in einem Modellsatz wiedergegeben werden (Indizierung).

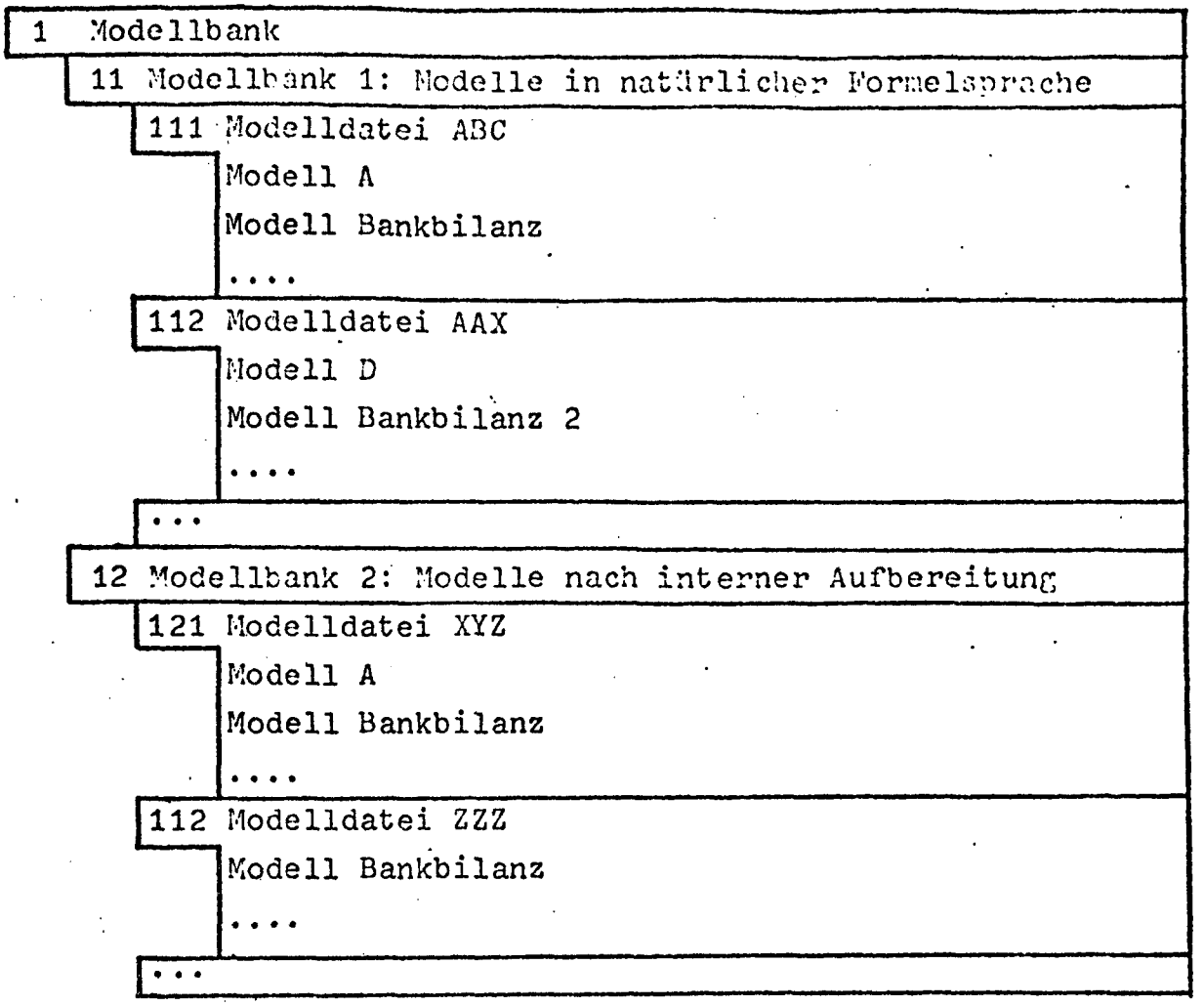

Abbildung 1

PLASMA - Dateiorganisation
- Modellbank -

Der sachliche Inhalt eines Modells ist in PLASMA ohne jede Bedeutung [vgl. dazu (26, S. 506)], woraus sich die Flexibilität des Systems ergibt. Es kann sich handeln um

- Modelle verschiedener Benutzer
- Modelle eines Benutzers für verschiedene Probleme
- Submodelle eines Modells eines Benutzers
- Alternativmodelle eines Benutzers für dasselbe Problem.

Die Speicherung von Modellen ist in PLASMA auf verschiedene Weise möglich. Die Modellbank 1 enthält Modelle in der üblichen mathematischen Formelsprache. Dabei werden Indizierungen nicht tief gestellt, sondern in Klammern angegeben; außerdem werden der Multiplikationspunkt durch das Sternsymbol, das Dezimalkomma durch den Punkt ersetzt.
Jeder Modellsatz weist eine Satzkennzeichnung, also z.B. die Formelnummer (11), auf:

```
(11)   Z (1977) + KOEFF * WERT (T-1) + A = 4.5 + X (T)
                - 3.47 * YY (T+6)        FUER T=1976,1978.
```

In der Beispielgleichung (11) kommen folgende Größen vor:

- die mit 1977 indizierte Variable Z
- der Koeffizient KOEFF, dessen Wert später einzugeben oder einer Datenbank zu entnehmen ist
- die mit T-1 indizierte Variable WERT
- die Variable A
- die numerische Konstante 4.5
- die mit T indizierte Variable X
- die mit T + 6 indizierte Variable YY, welche mit dem numerischen Koeffizienten 3.47 versehen ist.

Sämtliche Variablen können echte Modellvariablen sein oder später zu Konstanten werden. Die Beziehung (11) soll für die Perioden 1976 bis 1978 gelten.

Die Modellbank 1 enthält vom Prinzip her Dokumente, welche in PLASMA aus Formeln bestehen.

Neben der Modellbank 1 existiert eine für die weitere Verarbeitung erforderliche Modellbank 2. In dieser Modellbank 2 sind die von dem Programm PROTOPLASMA interpretierten Modellsätze enthalten. Dabei erfolgt eine interne Numerierung der Modellsätze, Variablen und Konstanten unter Berücksichtigung des mehrfachen Auftretens einer Größe in einem Modell. Die Modellbank 2 beinhaltet also sowohl Thesaurusdateien als auch Strukturdateien.

Bei der Modellbank 1 empfiehlt sich eine Gliederung in Subsysteme, wobei bei einer solchen Aufteilung die entsprechenden Konsequenzen im Bereich der Datenbank zu ziehen sind [vgl. dazu (3), S. 165 ff.] .

Weil die Modelle im allgemeinen nicht unveränderlich sind und also abänderbar sein müssen, wird man eine permanente Speicherung in der Modellbank 1 immer vorsehen. Denn Änderungen müssen in der natürlichen Formelsprache vorgenommen werden können. Die Modellbank 2 kann dagegen - bei häufigem Modelländerungsdienst oder bei extremen Speicherbeschränkungen - auch nur temporär aufgebaut werden.

2.2.2. Die Methodenbank

Die Methodenbank besteht aus drei Teilen (vgl. Abbildung 2):

- PLASMA-Systemverwaltung (Methodenbank 1)
- Datengewinnung und -verwaltung (Methodenbank 2)
- OR-Methoden (Methodenbank 3).

Die Methodenbank 1 enthält dabei nur zwei Programme (Dateien): das Systeminformationsprogramm PLASMAIS sowie das für PLASMA wesentliche Programm PROTOPLASMA, welches Modelle methodisch und numerisch aufbereiten und rechnen kann.

Die Methodenbank 2 beinhaltet das Dateneingabe- und Update-Programm PLASMA.221 und das Programm PLASMA.222 zur Datenmanipulation, insbesondere zur Kennzahlenbildung. Vorgesehen sind weiterhin eine Programmdatei PLASMA.223 mit statistischen Prognosemethoden und ein Reportgenerator PLASMA.224 zur flexiblen Berichtsgestaltung. Die letztgenannte Aufgabe ist insbesondere für Anwendungen in der Unternehmenspraxis unerläßlich [vgl. auch die Dokumentation über Reportgeneratoren in (25)]. Andererseits ist darauf hinzuweisen, daß ein Reportgenerator noch kein Planungssystem darstellt [vgl. (13), S. 152]; diese Einsicht scheint sich noch nicht überall durchgesetzt zu haben.

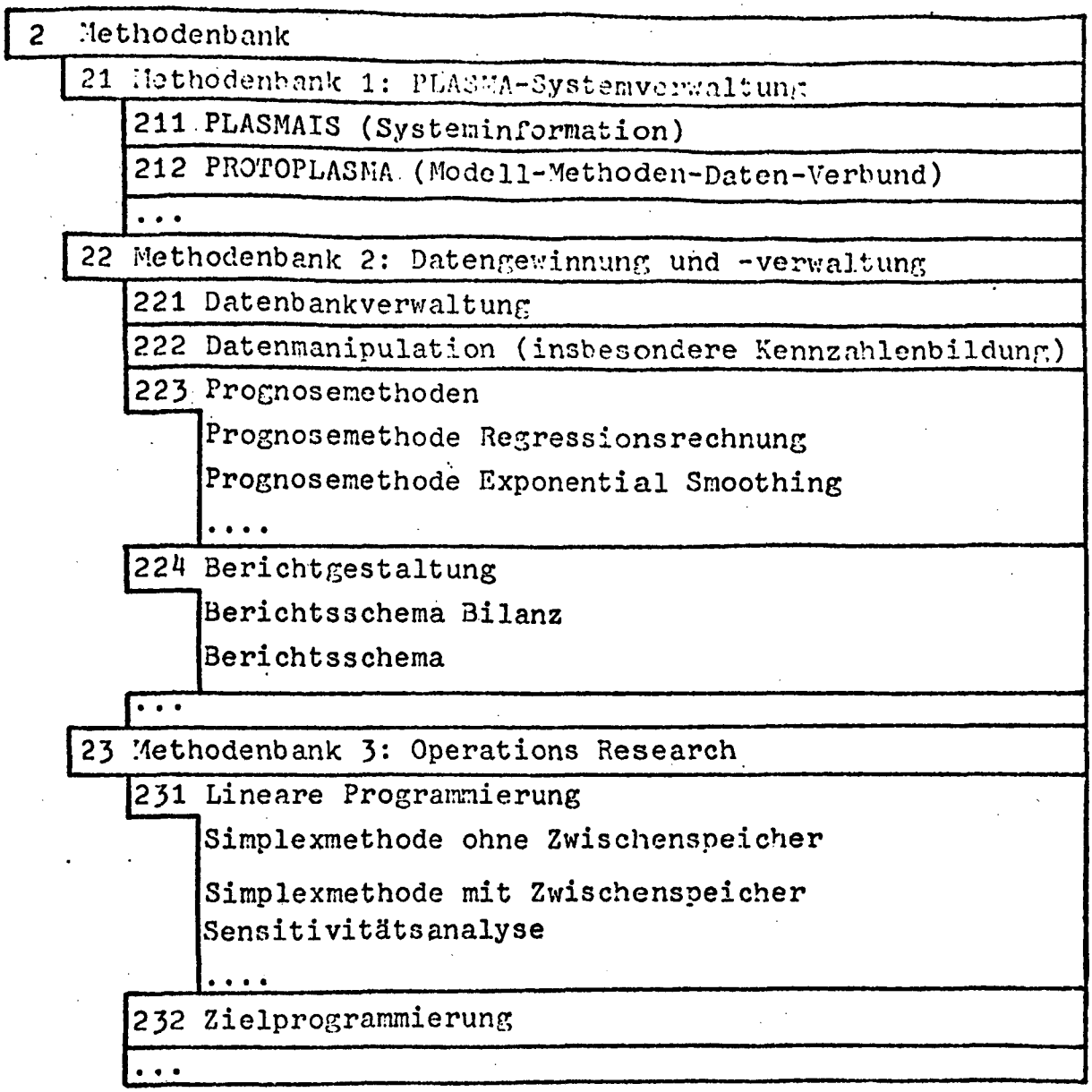

Abbildung 2

PLASMA - Dateiorganisation
- Methodenbank -

Die Methodenbank 3 enthält OR-Methoden, wobei die Programmdateien methodenweise abgegrenzt sind: z.B. Programme der linearen Programmierung (PLASMA.231) und der Zielprogrammierung (PLASMA.232). In PLASMA.231 kann dabei je nach Größe und Struktur des Problems zwischen mehreren LP-Algorithmen ausgewählt werden, wobei auch ein selbst geschriebener Algorithmus für große Probleme bei geringem Kernspeicher implementiert ist. Bei PLASMA.232 handelt es sich um das von den Verfassern adaptierte und für den Dialog ausgebaute Programm des Goal Programming von Lee (14).

Das Ausmaß der Dateiuntergliederung der Methodenbank hängt zunächst von der Hard- und Software der benutzten EDV-Anlage ab. Auf lange Sicht ist zu erwarten, daß die Methodenauswahl verstärkt vom Computer übernommen werden wird, was entweder nur eine Programmdatei oder einen starken Dateiverbund mehrerer Dateien impliziert.

2.2.3. Die Datenbank

Die Datenbank umfaßt hauptsächlich formatierte Daten, die sich in folgenden Dateitypen befinden (vgl. Abbildung 3)

- Vergangenheitsdaten, welche insbesondere aus dem Rechnungswesen und der Betriebsstatistik stammen (Datenbank 1)
- Prognosedaten, welche Ergebnisse der Anwendung einer Prognosemethode sind oder auf andere Weise geschätzt werden (Datenbank 2)
- Modelldaten, welche zu den methodisch und numerisch vollständig aufbereiteten Modellen gehören (Datenbank 3)
- Planungsdaten, welche Ergebnisse von Modellrechnungen - z.B. mit PLASMA - umfassen (Datenbank 4)
- Entscheidungs- und Vorgabedaten, welche im Unterschied zu den Planungsdaten die tatsächlich getroffenen Entscheidungen abbilden (Datenbank 5)
- PLASMA-Systemdaten, welche für Beschreibung und Ablauf von PLASMA erforderlich sind (Datenbank 6).

Aus dieser Aufzählung wird deutlich, daß Datenbanken verstärkt Zukunftsdaten speichern müssen. Es ist selbstverständlich, daß die Datenbank Dateien gleichen sachlichen, aber von verschiedenen Benutzern stammenden Inhalts enthalten kann. Dies wird hier nur betont, um zu zeigen, daß PLASMA mehrere Benutzer des gleichen Computersystems bedienen kann. Bei dem einzelnen Benutzer wiederum können Prognose-, Modell- und Planungsdateien sich auf mehrere Probleme oder auf dasselbe Problem beziehen, wobei Alternativdateien im letztgenannten Fall insbesondere für wissenschaftliche Vergleiche von Bedeutung sind.

In der Datenbank 2 können auch mit Wahrscheinlichkeiten versehene Daten enthalten sein [vgl. (19), S. 1137]. Die in der Literatur [(10), S. 184] angesprochene Zieldatei wird hier der Datenbank 5 zugeordnet.

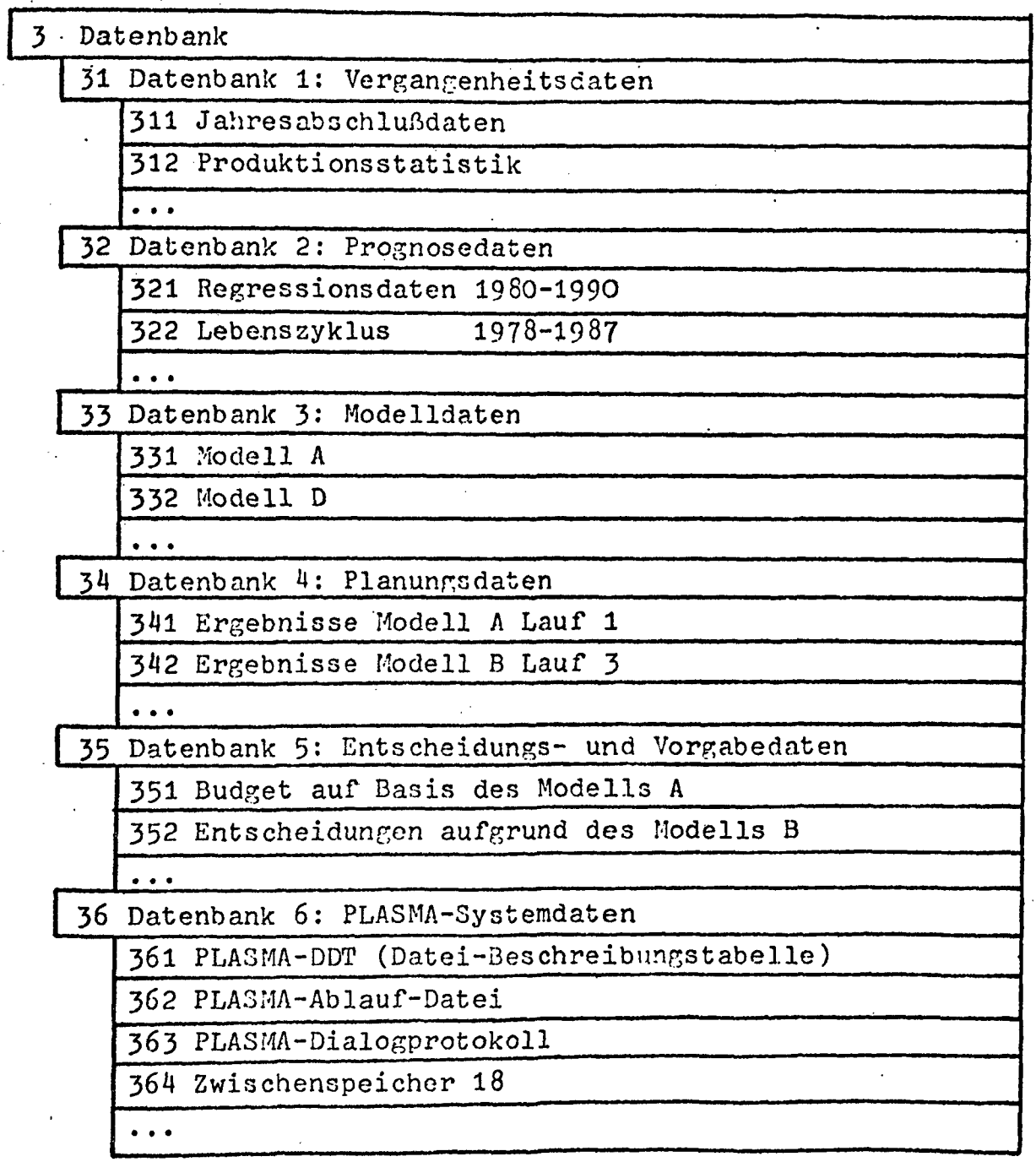

Abbildung 3

PLASMA - Dateiorganisation
- Datenbank -

Die Speicherorganisation der Datenbank ist nicht einheitlich, sondern problemabhängig vorgenommen worden. Aufdie meisten Dateien muß in Anbetracht der schnellen Verarbeitung bei einem Dialogprogramm direkt zugegriffen werden können. Der Direktzugriff in PLASMA erfordert die Angabe des Dateinamens, die Nennung des Namens des Datenfeldes (entsprechend der Modellformulierung) und die Angabe einer eventuellen Indizierung.

Bei den PLASMA-Systemdaten in Datenbank 6 enthält die Datei-Beschreibungstabelle PLASMA.361 u. a. die Information der Abbildungen 1 bis 3. Zur Analyse des Benutzerverhaltens ist in PLASMA.362 ein vollständiges Dialogprotokoll abgespeichert. Solche Protokolle sind für die Analyse des computergestützten Unterrichts und für die empirische Entscheidungsforschung relevant. Insbesondere wird dadurch ein Beitrag zur Datengewinnung für verhaltenstheoretische Modelle der computergestützten Planung [vgl. dazu (15)] geleistet.

2.3. Der Planungsablauf

2.3.1. Die Zerlegung in Teilprozesse

Der mit PLASMA mögliche computergestützte Planungsprozeß besteht aus Teilprozessen [vgl. ähnlich (18), S. 6o ff.], deren Reihenfolge vom Planenden selbst bestimmt werden kann. Insofern liegt ein benutzergesteuerter Dialog vor. Innerhalb der einzelnen Teilprozesse allerdings gilt das Prinzip des programmgesteuerten Dialogs. Der Benutzer kann also den Ablauf des jeweiligen Planungsprozesses frei gestalten, in den einzelnen Teilprozessen unterliegt er jedoch den Beschränkungen, welche durch die Programmsteuerung des Teilprozesses hingenommen werden müssen.

Die Beschränktheit der Abläufe in dem programmgesteuerten Dialog eines Teilprozesses ist kein Nachteil, denn dadurch können zunächst einmal die EDV-Kenntnisse des Benutzers auf ein Minimum reduziert werden. Darüber hinaus kann auf eine detaillierte Methodenkenntnis des Benutzers verzichtet werden, weil die im Zusammenhang mit einer Methode relevanten Fragen zum großen Teil maschinenintern beantwortet werden können. Auf lange Sicht werden auch im Bereich der Problemstellung, d.h. der Modellbildung, die Anforderungen an die Modellierungskenntnisse zurückgeschraubt werden können - sofern man die Benutzerfreundlichkeit für wesentlich hält. Es ist dafür eine Modellformulierung nicht in mathematischer Formelsprache, sondern in der natürlichen Fachsprache anzustreben. Eine solche Entwicklung bedeutet eine stärkere Verbindung der bisherigen Ansätze der computergestützten Planung, wobei bisher entweder sehr problembezogene und inflexible oder vom Ansatz her nicht problemorientierte und allgemeine Ansätze überwiegen [vgl. dazu etwa (5), (9), (23), (25)].

In Tabelle 1 sind alle in PLASMA möglichen (und sinnvollen) Folgen von Teilprozessen dargestellt. Die Zahl der sich daraus ergebenden Kombinationen von vollständigen oder unvollständigen Planungsprozessen ist so groß, daß für die Tabelle 1 die Form einer Übergangsmatrix gewählt wurde. Aus Tabelle 1 geht nicht hervor, welche Teilprozeßfolgen durchlaufen sein müssen, bevor eine bestimmte Folge von Zeile zu Spalte zulässig ist. Für solche Fragen könnte die Entscheidungstabellentechnik herangezogen werden.

Tabelle 1

Mögliche Planungsabläufe in PLASMA

von \ nach		0	1	2	3	4	5	6	7	8	9
0	PLASMA-System-Steuerung		ja	ja	ja	ja	ja		ja	ja	ja
1	PLASMA-System-Information	ja		ja	ja	ja	ja		ja	ja	ja
2	Datenverwaltung und Gewinnung	ja	ja		ja	ja	ja			ja	ja
3	Modellkonstruktion und -verwaltung	ja	ja	ja		ja					ja
4	Interne Modellaufbereitung	ja	ja		ja		ja	ja			ja
5	Dateneingabe	ja	ja	ja	ja			ja			ja
6	Methodische und numerische Modellaufbereitung	ja	ja		ja		ja		ja		ja
7	Modellrechnung	ja	ja	ja	ja	ja	ja			ja	ja
8	Bericht über Modellergebnisse	ja	ja	ja	ja	ja	ja				ja
9	Bericht über den Planungsablauf	ja	ja	ja	ja	ja	ja		ja	ja	

Das koordinierende Element von PLASMA bildet die PLASMA-Systemsteuerung. Dabei handelt es sich um ein die Teilprozesse im Dialog steuerndes und dokumentierendes Programm, das auch die für die Ausführung erforderlichen Dateien festlegt. Durch die PLASMA-Systemformation kann der Benutzer Auskünfte über das ganze System oder seine Teile erhalten. Mit dem Programm PLASMA.221 zur Datenverwaltung und -gewinnung kann die PLASMA-Datenbank beschickt und verändert werden, wobei im Falle der Existenz einer anderen Datenbank des Benutzers noch für den Datenaustausch zu sorgen ist.

Im folgenden werden die für den Modell-, Methoden- und Datenverbund wesentlichen Teilprozesse diskutiert. In dem Dialog werden Mitteilungen der Maschine beidseitig mit doppeltem Ausrufungszeichen, Fragen an den Benutzer beidseitig mit doppeltem Fragezeichen versehen. Die Antworten des Benutzers erscheinen auf dem Bildschirm und im Dialogprotokoll (vgl. unten) möglichst in der gleichen Ausgabezeile wie die betreffende Frage. Ein Doppelpunkt zeigt den Beginn einer Antwort des Benutzers an.

2.3.2. Modellkonstruktion und -verwaltung

Die Modellkonstruktion und -verwaltung basiert auf einem eigenen Datei-Manipulationssystem, so daß PROTOPLASMA nicht nur in Planungs-, sondern z.B. auch in Dokumentationssystemen anwendbar erscheint. Die Konstruktionsmöglichkeit von Modellen im Dialog in mathematischer Formelsprache [vgl. einen anderen Ansatz in (6)] trägt wesentlich zur Flexibilität des Planungssystems bei. In PROTOPLASMA wurde dabei eine Syntax ähnlich MATRAN [vgl. dazu (22), S. 32 ff.] gewählt. Die maximale Länge einer Variablen oder eines nichtnumerischen Koeffizienten beträgt sechs Zeichen. Folgende Möglichkeiten stehen dem Benutzer offen:

- Wahl der Modelldatei
- Erstmaliges Anlegen eines Modells unter einem bestimmten Namen und Speicherung dieses Modells
- Zusammenfügen von zwei oder mehr Modellen zu einem neuen Modell

- Eingabe einzelner Sätze eines Modells, wobei der Name des Modellsatzes und der Satzinhalt anzugeben sind - die Eingabe erfolgt dabei segmentweise (variable Satzlänge)
- Änderung eines Modells
  - durch Einfügen oder Löschen ganzer Modellsätze
  - durch Einfügen oder Löschen von einzelnen Zeichen oder Zeichenfolgen in einem Modellsatz
  - durch Überschreiben von einzelnen Zeichen oder Zeichenfolgen in einem Modellsatz
- Löschen ganzer Modelle.

Der Ablauf einer Modellierungsarbeit ist dem folgenden Dialogprotokoll zu entnehmen, welches die Eingabe eines Modells wiedergibt.

```
?? WELCHE MODELLDATEI WOLLEN SIE BENUTZEN ??                    :TESMO.
!! GEBEN SIE BITTE DAS GLEICHUNGSSYSTEM EIN !!
?? KENNZEICHNUNG DES EINZUGEBENDEN SATZES ??                    :KLAAB
?? SATZ  1    SEGMENT  1 ??
W + S = R
?? SATZ  1    SEGMENT  2 ??

?? WEITERER SATZ ??                                             :J
?? KENNZEICHNUNG DES EINZUGEBENDEN SATZES ??                    :KLBAB
?? SATZ  2    SEGMENT  1 ??
T + S = W
?? SATZ  2    SEGMENT  2 ??

?? WEITERER SATZ ??                                             :J
?? KENNZEICHNUNG DES EINZUGEBENDEN SATZES ??                    :KLCAB
?? SATZ  3    SEGMENT  1 ??
W + T = S
?? SATZ  3    SEGMENT  2 ??

?? WEITERER SATZ ??                                             :N
?? AUSDRUCK DES MODELLS ??                                      :J
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                       :J
KLAAB
 W + S = R
KLBAB
 T + S = W
KLCAB
 W + T = S
?? AUSDRUCK DES MODELLS ??                                      :N
?? SATZ HINZUFUEGEN ??                                          :N
?? MODELL SPEICHERN ??                                          :J
?? NAME DES MODELLS ??  :KL(ABC)AB
```

Das folgende Dialogprotokoll zeigt, wie ein neues Modell durch Zusammenfügen eines Submodells "KL(ABC)AB" mit dem Satz "KLXAB" des Submodells "KL(ABCX)AB" und durch Übernahme des Modells "Indizierung nach 105" entsteht.

```
?? NEUES MODELL GENERIEREN ??                                    :J
?? AUS BESTEHENDEN MODELLEN ZUSAMMENFUEGEN ??                    :J
?? KENNZEICHNUNG DES ZU VERWENDENDEN MODELLS ??
   :
!! FOLGENDE MODELLE SIND GESPEICHERT:
INDIZIERUNG NACH 105
KL(ABCX)AB
KL(ABC)AB
?? WEITERES SUBMODELL VERWENDEN ??                               :J
?? KENNZEICHNUNG DES ZU VERWENDENDEN MODELLS ??
   :KL(ABC)AB
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                        :J
KLAAB
 W + S = R
KLBAB
 T + S = W
KLCAB
 W + T = S
?? MODELL VOLLSTAENDIG UEBERNEHMEN ??                            :J
?? WEITERES SUBMODELL VERWENDEN ??                               :J
?? KENNZEICHNUNG DES ZU VERWENDENDEN MODELLS ??
   :KL(ABCX)AB
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                        :J
KLAAB
 W + S = R
KLBAB
 T + S = W
KLCAB
 W + T = S
KLXAB
 END + D
?? MODELL VOLLSTAENDIG UEBERNEHMEN ??                            :N
?? KENNZEICHNUNG DES AUSZUGEBENDEN SATZES ??                     :KLXAB
?? WEITEREN SATZ AUSGEBEN ??                                     :N
?? WEITERES SUBMODELL VERWENDEN ??                               :J
?? KENNZEICHNUNG DES ZU VERWENDENDEN MODELLS ??
   :INDIZIERUNG NACH 105
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                        :N
?? MODELL VOLLSTAENDIG UEBERNEHMEN ??                            :J
?? WEITERES SUBMODELL VERWENDEN ??                               :N
?? AUSDRUCK DES MODELLS ??                                       :J
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                        :J
KLAAB
 W + S = R
KLBAB
 T + S = W
KLCAB
 W + T = S
KLXAB
 END + D
X105
 S(T) + F(T) + WF* RE(T) =ERG      FUER T=1,3
```

Das folgende Dialogprotokoll zeigt das Einfügen einer Zeichenfolge. Die Stelle des Einfügens wird dadurch festgelegt, daß der unmittelbar vor der Stelle liegende Text zitiert wird.

```
?? AUSDRUCK DES MODELLS ??                                    :J
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                     :N
?? KENNZEICHNUNG DES AUSZUGEBENDEN SATZES ??                  :KLXAB
KLXAB
 END + D
?? AENDERN DES SATZES ??                                      :J
?? LOESCHEN ??                                                :N
?? UEBERSCHREIBEN ??                                          :N
?? EINFUEGEN ??                                               :J
?? WONACH ??     :+ D
   TEXT:* WOLF - AA2 * TY + REG = T
KLXAB
 END + D* WOLF - AA2 * TY + REG = T
?? AENDERN DES SATZES ??                                      :N
?? AUSDRUCK DES MODELLS ??                                    :N
?? SATZ HINZUFUEGEN ??                                        :N
?? MODELL SPEICHERN ??                                        :J
?? NAME DES MODELLS ??  :RECHENVERSION
```

Das Löschen einer Zeichenfolge ist dem folgenden Dialogprotokoll zu entnehmen. Dabei ist die zu löschende Zeichenfolge anzugeben. Kommt die zu löschende Zeichenfolge mehrmals vor und soll nur eine bestimmte dieser Folgen gelöscht werden, so muß die Operation "Überschreiben" mit Leerzeichen ausgeführt werden.

```
?? WELCHE MODELLDATEI WOLLEN SIE BENUTZEN ??                  :PLAN5.DAT
!! ZAHL DER GESPEICHERTEN MODELLE:  5 !!
?? MODELL LOESCHEN ??                                         :N
?? NEUES MODELL GENERIEREN ??                                 :N
?? KENNZEICHNUNG DES ZU VERWENDENDEN MODELLS ??
   :VERSION 03
?? AUSDRUCK DES MODELLS ??                                    :J
?? VOLLSTAENDIGER AUSDRUCK DES MODELLS ??                     :N
?? KENNZEICHNUNG DES AUSZUGEBENDEN SATZES ??                  :LKAA
LKAA
 END + WOLF - AA2*TY + D * REG + VT * REG(4) = T
?? AENDERN DES SATZES ??                                      :J
?? LOESCHEN ??                                                :J
?? GANZEN SATZ LOESCHEN ??                                    :N
?? WAS SOLL GELOESCHT WERDEN ??
   :G(4)
LKAA
 END + WOLF - AA2*TY + D * REG + VT * RE = T
?? AENDERN DES SATZES ??                                      :N
```

Bei der Systemgestaltung wurde darauf geachtet, daß die Änderung eines in einer Modelldatei vorhandenen Modells an einer Modellkopie vorgenommen wird. Das geänderte oder zusammengefügte Modell muß - wenn es gespeichert werden soll - unter einem neuen Namen gespeichert werden. Dadurch wird die Modelldokumentation verbessert, und das alte Modell bleibt erhalten.

2.3.3. Interne Modellaufbereitung

Die interne Aufbereitung von Modellen wird von einem in PROTOPLASMA enthaltenen Übersetzer vorgenommen. Dabei handelt es sich im Prinzip um einen Matrixgenerator. In den Modellsätzen sind dabei vier Typen von Elementen zu unterscheiden:

- Variablen (die bei der Datenaufbereitung später eventuell zu numerischen Kostanten werden können)
- nichtnumerische Koeffizienten (deren Werte bei der Datenaufbereitung zu numerischen Koeffizienten werden müssen)
- numerische Koeffizienten
- numerische Kostanten.

Die Zulassung nichtnumerischer Koeffizienten vor einer Variablen bewirkt im Grunde die Trennung von Modellbank und Datenbank. Auch die allgemeine Formulierung von Modellkonstanten als Variablen, die erst später zu Konstanten werden, trägt zu der Trennung bei.

2.3.4. Dateneingabe

Im Anschluß an die interne Modellaufbereitung kann nur dann eine direkte Verbindung zur Methodenbank hergestellt werden, wenn alle numerischen Werte in dem ursprünglichen Modell schon enthalten sind. Dies wird in der Regel nicht der Fall sein, so daß die speziellen Modelldaten noch eingegeben werden müssen.

Für die Dateneingabe bestehen zwei Möglichkeiten:

1. Abruf der benötigten Daten aus einer vom Benutzer zu benennenden Datei. Dies ist sinnvoll, wenn der Datenumfang groß ist oder dieselben Daten mehrmals benötigt werden. Der Abruf der Daten setzt voraus, daß die Daten mit Hilfe des Datenbankverwaltungsprogramms in die Datenbank gelangt sind.
2. Eingabe der benötigten Daten im Dialog. Dies ist sinnvoll, wenn nur wenige Daten einzugeben sind oder wenn zwar ein großer Teil der Daten abgerufen werden kann, aber ein geringer Teil noch nicht vorhanden ist.

Nur die Dateneingabe über das Datenbankverwaltungsprogramm stellt einen eigenen Teilprozeß im Rahmen von PLASMA dar. Der Abruf der Daten aus der Datenbank oder die Dateneingabe im Dialog sind schon mit der methodischen und numerischen Modellaufbereitung verknüpft, worauf im folgenden eingegangen wird.

2.3.5. Methodische und numerische Modellaufbereitung

Methodische und numerische Datenaufbereitung sind unmittelbar miteinander verbunden, weil die aufgrund einer Methode erforderliche Verwandlung von Variablen in Konstanten die numerische Angabe der Werte der Konstanten erfordert.

Entscheidet man sich für die Anwendung der linearen Programmierung, so ist zunächst die Satzkennung der Zielfunktion anzugeben. Dies führt zu einer Umsortierung der Modellsätze mit dem Ergebnis, daß die Zielfunktion den letzten Modellsatz - wie für die Modellrechnung erforderlich - bildet.

Danach werden zunächst sämtliche Koeffizienten des Modells vom Programm her auf ihren Wert abgefragt - soweit sich der Wert nicht schon direkt aus dem Modellsatz ergibt. Werden die Koeffizienten einer Datei der Datenbank entnommen, so kann der Benutzer durch Entscheidung für eine Abfrageoption erreichen, daß er bei jedem Koeffizienten noch einmal gefragt wird, ob der in der Datenbank enthaltene und ihm zur Kenntnis gebrachte Wert in das Modell eingehen soll. Alle nach Suchen in der Datei nicht gefundenen oder alle in das Modell nicht übernommene Koeffizienten sind auf Anforderung des Programms vom Benutzer im Dialog wertmäßig zu spezifizieren.

Im folgenden ist die Datenaufbereitung anhand des Dialogprotokolls wiedergegeben.

```
?? SOLL DATENAUFBEREITUNG ERFOLGEN ??                          :J
?? LINEARE PROGRAMMIERUNG ??                                   :J
?? SATZKENNUNG DER ZIELFUNKTION ??                             :LZIEL
?? NUMERISCHE DATEN AUS DATENBANK ENTNEHMEN ??                 :J
?? NAME DER DATEI ??          :DATA.
!! KONSTANTE AA2      INDEX:   -1   WERT:      12.00000 !!
?? EINFUEGEN ??                                                :J
!! KONSTANTE D        INDEX:   -1   WERT:      11.00000 !!
?? EINFUEGEN ??                                                :J
?? WERT DES KOEFFIZIENTEN VT     INDEX:   -1                   :     19.00000
?? VARIABLE ZU KONSTANTEN ??                                   :J
!! ERD      INDEX :    -1 WERT :   28.0000
?? EINFUEGEN ??                                                :N
!! FGA      INDEX :  1948 WERT :   21.0000
?? EINFUEGEN ??                                                :N
!! FGA      INDEX :  1949 WERT :   22.0000
?? EINFUEGEN ??                                                :N
!! REG      INDEX :    -1 WERT :   20.0000
?? EINFUEGEN ??                                                :J
!! RE       INDEX :     1 WERT :   25.0000
?? VARIABLE ZU KONSTANTEN ??                                   :N
!! ENDE DATENAUFBEREITUNG !!
!! DATENAUSGABE !!
?? NAME DER AUSGABEDATEI ??          : OUTPUT.
!!  W A R N U N G  !!
    ZAHL DER VARIABLEN ( 17)
    UNGLEICH ZAHL DER GLEICHUNGEN (  7)
?? DATENAUSGABE FORTSETZEN ??                                  :J
!! ENDE DER DATENAUSGABE !!
?? WEITERER PROGRAMMLAUF ??                                    :N
!! ENDE DES PROGRAMMS ' P R O T O P L A S M A ' !!
```

Für die praktikable Gestaltung von Modellen ist es - etwa bei Zeitindizierung - sinnvoll, das Modell zuerst allgemein für alle Perioden zu formulieren und dann die Anfangswerte durch Umwandlung von Variablen in Konstanten festzulegen. In PROTOPLASMA geschieht dies durch die in den obigen Dialogprotokoll enthaltene Abfrage: "Variable zu Konstanten?". Durch Angabe einer Variablen in diesem Zusammenhang wird der konstante Wert einer solchen Variablen zunächst in der Datenbank gesucht und bei erfolgloser Suche im Dialog erfragt.

Bei Anwendung der Zielprogrammierung kommt der Umwandlung von Variablen zu Kostanten besondere Bedeutung zu, weil auf diese Weise mehrere Zielgrößen numerisch im Dialog spezifiziert werden können. Gleichzeitig müssen die erforderlichen Prioritäten und Gewichte festgelegt werden [vgl. dazu (14)]. Die Verbindung von Dialog und Zielprogrammierung einerseits und schneller Modellgenerierung andererseits wird vor allem für Gruppenentscheidungen interessant sein. Es kann schnell überprüft werden, welche Folgen die durch formale Abstimmungen bestimmten Zielwerte und Prioritäten für die zu ergreifenden Maßnahmen haben.

Im Rahmen der methodischen und numerischen Aufbereitung werden alle Variablen und ihre Koeffizienten auf die linke Seite, alle Konstanten auf die rechte Seite des Gleichheits- oder Ungleichheitszeichens gebracht. Mehrere Konstanten werden dabei vom Programm her addiert. Gleichzeitig wird festgehalten, ob es sich um eine Gleichung, eine Mindestbedingung ($\geq$) oder eine Maximalbedingung ($\leq$) handelt. Damit sind alle Informationen vorhanden, um die Modelldaten einer Ausgabedatei zu übergeben. In dieser Ausgabedatei der Modelldaten werden wie üblich nur von Null verschiedene Werte gespeichert.

Die Möglichkeit einer Modellrechnung wird durch Plausibilitätsprüfungen getestet. Bei der Lösung von What-if-Modellen ist z.B. auf Variablen- und Zeilenzahl des Systems zu achten [vgl. dazu das Beispiel in (24)]. Die Plausibilitätsprüfungen sind bei hohen Rechenzeitgen des Modells besonders auszubauen, um unnötige CPU-Zeiten zu vermeiden. Auch ist an programmgesteuerte Umformungen des aufbereiteten Modells zu denken, z.B. durch Reduktion der Zeilenzahl oder durch Dekomposition des Modells.

2.3.6. Modellrechnung

Die Modellrechnung mit den vorher erzeugten Modelldaten kann zwar sofort erfolgen, bei großen Problemen wird man jedoch hier den Dialog beenden und die Modellrechnung über Stapelverarbeitung erfolgen lassen. Die Entscheidung für eine der beiden Vorgehensweisen hängt dabei auch von der Art des eingesetzten Programms ab. So wird in PLASMA festgestellt, ob die Problemgröße es erlaubt, die Matrix im Kernspeicher zu halten, also auf die Auslagerung von Werten während der Rechnung zu verzichten. Ist das Problem entsprechend klein, so wird man auf eine Stapelverarbeitung verzichten können. Das Verbleiben in PLASMA hat den Vorteil, daß Ergebnisse - mit Variablenbezeichnung und zeitlicher Indizierung - sofort zur Kenntnis genommen werden können und andere Teilprozesse gemäß Tabelle 1 folgen können. Dies ist ein wesentlicher Gewinn beim Verbund von Modellen, Methoden und Daten.

Die Speicherung der Modellergebnisse in der Datenbank hat zwei Vorteile:

1. Ein separables Problem kann unter Verwendung der Modellergebnisse des vorherigen Schritts für eine anschließende Periode durchgerechnet werden.
2. Der Vergleich von Modellergebnissen untereinander, von Modellergebnissen mit den getroffenen Entscheidungen sowie von Modellergebnissen, Entscheidungen mit der späteren Realisation ermöglicht die Entwicklung eines lernenden Systems. Hier liegt die Zukunft von computergestützten Entscheidungen.

2.3.7. Zum praktischen Einsatz von PLASMA

Das Programmpaket PLASMA besteht in der ersten Ausbaustufe aus über 3000 FORTRAN-Befehlen, wovon ca.1500 Befehle in PROTOPLASMA enthalten sind. Man kommt mit 32 K Worten im Kernspeicher aus. Eine solche Beschränkung ergibt sich nicht nur aus dem Time-Sharing-Betrieb der PDP 10 des Rechenzentrums der Universität Kiel. Es muß vielmehr generell auf den Kernspeicherbedarf geachtet werden, damit auch mittlere Unternehmen nicht vor einer computergestützten Planung zurückschrecken. Es ist auch eine Aufgabe der Betriebswirtschaftslehre, die Konzentration in der Wirtschaft nicht noch durch Entwicklung von Planungssoftware zu fördern, welche nur von Spezialisten einsetzbar und auf Großcomputern durchführbar ist. Das Konzept der Zerlegung des Planungsprozesses in Teilprozesse (vgl. Tabelle 1) erfüllt damit nicht nur den Zweck eines flexiblen Planungsablaufs, sondern es nimmt auf Hardware-Beschränkungen des Benutzers Rücksicht.

Auch die Konzeption von PLASMA in bezug auf die Software geht in die gleiche Richtung. Die Benutzer sollen nicht mit immer neuen, meist fremdsprachigen Makrosprachen dem Computerhersteller weiter ihre Referenz erweisen müssen. Vielmehr muß das Ziel eine Verständigung mit dem Computer in natürlicher Sprache sein [vgl. dazu auch (8)]. Flexibilität ist programmintern zu gewährleisten. Von der Dialoggestaltung mit Hilfe der Menu Selection-Technik [vgl. dazu (16), S. 111 ff.] wird weitgehend Abstand genommen, weil so die Anforderungen an den Benutzer niedriger gehalten werden können. Der Unternehmensplaner hat sich mit den betriebswirtschaftlichen Problemen zu befassen: Systemzusammenhang, Zielsetzung, Unsicherheit [vgl. dazu (1)].

Bisherige Einsätze von PLASMA [z.B. in (24)] brachten die Erfahrung, daß der Benutzer eher bereit ist, eigene Modellformulierungen in Frage zu stellen oder alternative Datenkonstellationen zu untersuchen. Damit wird tendenziell der sog. Fehler dritter Art bekämpft, nämlich die richtige Lösung des falschen Problems zu präsentieren.

Der Computer behält auch im Rahmen eines programmgesteuerten Dialogs die dienende Funktion, weil die Programmsteuerung selbst vorprogrammiert ist oder der Mensch maschinelle Lernprozesse und daraus folgende Systemanpassungen kontrollieren wird.

3. Ergebnis

Anhand des Planungssystems für mathematische Anwendungen (PLASMA) wurde untersucht, welche Probleme sich bei der Verknüpfung von Modellen, Methoden und Daten ergeben und wie diese Probleme in den Griff zu bekommen sind. Wegen der wechselnden Problemstellungen der Planung, wegen des ständigen Fortschritts im Bereich der Methoden sowie wegen der Fülle zukünftiger, unsicherer Datenkonstellationen ist auf die Flexibilität des Planungssystems der größte Wert zu legen. PLASMA will dafür ein Beispiel geben, indem die Starrheit von Abläufen durch vom Benutzer ansteuerbare Teilprozesse beseitigt wird und die einzelnen Teilprozesse für Änderungen offen sind. Innerhalb der Teilprozesse läuft ein programmgesteuerter Dialog ab, der den Gesichtspunkt der Benutzerfreundlichkeit in den Vordergrund stellt.

Die Anwendungsmöglichkeiten eines solchen Systems werden nicht nur in der Unternehmensplanung selbst gesehen. Auch betriebswirtschaftliche Forschung und Lehre benötigen ein solches Instrument, damit nicht nur jeweils ein Modell für eine akademische Problemstellung formuliert wird. Vielmehr ist für ein gegebenes realitätsnahes Problem das beste Modell zu suchen.

Literatur

1. Albach, Horst, Beiträge zur Unternehmensplanung, Wiesbaden 1969.

2. Brockhoff, Klaus, Planung in mittelgrossen Industrieunternehmen, Die Unternehmung 29 (1975), S. 3o3-317.

3. Döringer, Hartmut, Integrierte computer-unterstützte Informations- und Entscheidungssysteme, Diss. Mannheim 1975.

4. Emde, Wilhelm, und Ulrich Hasenkamp, Modell- und methodenorientierte Anwendungs-Software in entscheidungsorientierten Informationssystemen - Beurteilungskriterien und Auswahl, Köln 1972.

5. Grochla, Erwin, und Norbert Szyperski (Hrsg.), Modell- und computer-gestützte Unternehmungsplanung, Wiesbaden 1973.

6. Groh, H., und M. Trittelvitz, GRAMAG, Ein Ansatz zur Rationalisierung der Linearen Optimierung, in: Proceedings in Operations Research 4 Würzburg und Wien 1974, S. 85-95.

7. Hellmich, Reinhard, Entwicklung eines Planungssystems für Unternehmungen, Winterthur 197o.

8. Hudetz, W., A Graphical Interactive System Simulation Program for Mini-Computers, Paper presented at the DECUS Symposium Europe, München, 8.-1o. Sept. 1976.

9. Hussey, D. E. (Hrsg.), The Corporate Planner's Yearbook 1974-5, Oxford et al. 1974.

1o. Kayser, Volkmar, Die Planung des Unternehmungswachstums-Entwurf eines Modells zur computergestützten Wachstumsplanung, Diss. Köln 1974

11. Kieser, Heinz-Peter, Mensch-Maschine-Kommunikation in Organisationen, Diss. Mannheim 1973.

12. Kirsch, Werner, Eduard Gabele et al., Planung und Organisation in Unternehmen, Bericht aus einem empirischen Forschungsprojekt, München 1975.

13. Lande, H. F., Planning-data systems, IBM Systems Journal 12 (1973), S. 145-16o.

14. Lee, Sang M., Goal Programming for Decision Analysis, Philadelphia 1972.

15. Marock, Jürgen, Ein verhaltenstheoretisches Modell als Hilfsmittel bei der organisatorischen Implementierung computergestützter Planungssysteme, Diss. Köln 1974.

16. Martin, James, Design of Man-Computer Dialogues, Englewood Cliffs 1973.

17. Mertens, Peter, und Joachim Griese, Industrielle Datenverarbeitung, Band II, Informations- und Planungssysteme, Wiesbaden 1972.

18. Morton, Michael S. Scott, Management-Entscheidungen im Bildschirmdialog, Essen 1972.

19. Moses, Michael A., Implementation of Analytical Planning Systems, Management Science 21 (1975), S. 1133-1143.

2o. Naylor, Thomas H., und Horst Schauland, A Survey of Users of Corporate Planning Models, Management Science 22 (1976), S. 927-937.

21. Reitbauer, F., H. Österle und J. Griese, Ein System zur Unterstützung von interaktiven Programmen, Angewandte Informatik 15 (1973), S. 157-173.

22. Robichek, A. A., D. Teichroew und J. M. Jones, Optimal Short Term Financing Decision, Management Science 12 (1965), S. 1-36.

23. Schild, Heinz G., Überlegungen zur Software für die strategische Planung, in: Grochla, Erwin, und Norbert Szyperski (Hrsg.), Modell- und computergestützte Unternehmungsplanung, Wiesbaden 1973, S. 463-471.

24. Schmidt, Reinhart, Zur Planungsflexibilität bei der Planung von Bankbilanzen, in: Manuskripte aus dem Institut für Betriebswirtschaftslehre der Universität Kiel, Nr. 38, September 1976.

25. Steinacke, Volkmar, Olaf Seifert und Dietrich Ohse, Lineare Planungsmodelle im praktischen Einsatz, Berlin et al. 1973.

26. Winand, Udo, Zur Gestaltung von Modell - Methoden - Verbund - Programmen im Rahmen von quantitativen modellgestützten Problemlösungen, Angewandte Informatik 17 (1975), S. 5o3-513.