

## DETAM - EIN SOFTWARE-SYSTEM ZUR PROZESSAUTOMATISIERUNG

### MIT MIKRORECHNERN UND ENTSCHEIDUNGSTABELLEN

S. NESTEL, K. SCHAPER

STUTT GART

STANDARD ELEKTRIK LORENZ AG  
FORSCHUNGSZENTRUM

17.7.1980

Dissertation TH Aachen, 1977.

## Zusammenfassung

Im SEL-Forschungszentrum wurde mit finanzieller Unterstützung durch das Projekt "Prozesslenkung mit Datenverarbeitungsanlagen (PDV)" das Entscheidungstabellen-Verarbeitungssystem DETAM entwickelt. DETAM (= Decision Table Support for Microcomputers) besteht aus einem Entscheidungstabellen-Aufbereitungssystem, das auf den Mikrorechner-Entwicklungssystemen Intel MDS 800 oder Intel MDS 231 betrieben wird, sowie aus Realzeitverarbeitungsprogrammen, welche auf den Zielrechnern Intel 8085 oder Intel 8086 den automatisch aufbereiteten und in den Arbeitsspeicher des Zielrechners geladenen Tabellen-Code verarbeiten.

Der Anwender kann je nach Speicherplatz- und Laufzeitanforderungen zwischen drei alternativen Verarbeitungsalgorithmen wählen und erhält für Planungs- und Optimierungszwecke den im Zielrechner zu erwartenden Laufzeit- und Speicherplatzbedarf ausgedruckt. Alle notwendigen Eingaben werden über einen interaktiven Dialog am Bildschirm abgefragt. Die eingegebenen Tabellen werden auf Widerspruch und Redundanz geprüft.

DETAM bietet die Möglichkeit, Teilaufgaben der Software-Entwicklung automatisch zu bearbeiten. DETAM liefert damit einen Beitrag zur industriege-rechten Software-Fertigung.

### 1. Einführung

Das Schema einer Entscheidungstabelle zeigt Bild 1. Theorie und Praxis der Entscheidungstabellentechnik seien hier als bekannt vorausgesetzt, siehe z.B. [1], [2], [3]. Über Entscheidungstabellen liegt eine DIN-Norm vor [4].

Vor allem für Anwendungen in der Betriebsorganisation und Verwaltung stehen für die meisten größeren Rechner bereits Entscheidungstabellenum-setzer zur Verfügung, siehe [5] und [6]. Die Entscheidungstabellentechnik eignet sich jedoch auch als übersichtliches und kompaktes Spezifikations-hilfsmittel bei Prozeßsteuerungsproblemen mit umfangreichen logischen Verknüpfungen (Schaltfunktionen) sowie zur Programmierung von Realzeitabläufen, welche durch zeitlich nicht-determiniert auftretende Ereignisse veranlaßt werden (z.B. Hardware- und Software-Unterbrechungen mit Dringlichkeitsstufen).

	R1	R2	R3	R4
Bedingung 1	J	J	N	N
Bedingung 2	J	N	J	N
Bedingung 3	J	-	N	N
Aktion 1	X	X	-	X
Aktion 2	-	-	X	X
Aktion 3	-	X	-	-

**Bild 1 Entscheidungstabelle**

Regeln (Verknüpfungsfälle):

$R_1 \dots R_N$

Bedingungsanzeiger:

J = JA

N = NEIN

- = Nichtrelevanzanzeiger

Im erweiterten Falle:

Beliebige alphanumerische Angaben

Aktionsanzeiger:

X = Aktion ist auszuführen

- = Aktion ist nicht auszuführen

Während Schaltfunktionen zur Prozesssteuerung z.B. in der Nachrichtentechnik, Verfahrenstechnik, sowie in der Industrie- und Konsumelektronik heute immer mehr durch Mikrorechnerprogramme anstatt durch Hardware-Logik realisiert werden, ist es vorteilhaft, auch für Mikrorechner die Entscheidungstabellentechnik nutzbar zu machen. Aus diesen Überlegungen entstand DETAM.

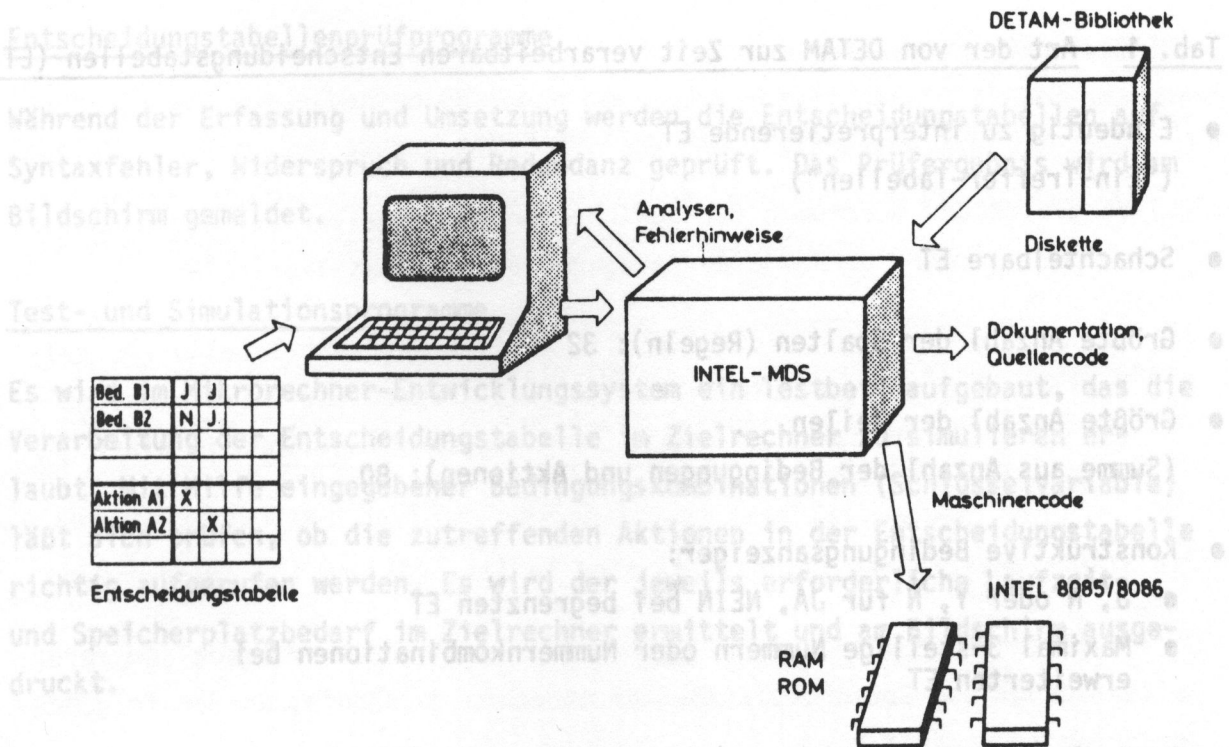
## 2. Ausstattung und Betriebsweise von DETAM

Eine Übersicht zu DETAM zeigt Bild 2. DETAM stellt folgende Programmpakete bereit:

### Eingabedialog

Nach dem Aufruf des Eingabedialogs sind vom Anwender am Bildschirm u.a. folgende Angaben zu liefern:





**Bild 2** Übersicht DETAM

- Art der zu verarbeitenden Entscheidungstabelle (Art der Bedingungsanzeiger, Dimension der Tabelle, Links-vor-Rechts oder Parallel-Interpretation der Entscheidungstabelle). Die Arten der von DETAM zur Zeit verarbeitbaren Entscheidungstabellen zeigt Tabelle 1.
- Einteilung der Schlüsselvariablen in Rechneranteilwörter
- Symbolische Adressen der Aktionen

Mit Hilfe eines eigens dafür entwickelten Tabellen-Editors ist anschließend ein mit Fragezeichen vorbesetztes Tabellenformular am Bildschirm auszufüllen.



Tab. 1 Art der von DETAM zur Zeit verarbeitbaren Entscheidungstabellen (ET)

- Eindeutig zu interpretierende ET  
("Ein-Treffer-Tabellen")
- Schachtelbare ET
- Größte Anzahl der Spalten (Regeln): 32
- Größte Anzahl der Zeilen  
(Summe aus Anzahl der Bedingungen und Aktionen): 80
- Konstruktive Bedingungsanzeiger:
  - J, N oder Y, N für JA, NEIN bei begrenzten ET
  - Maximal 3-stellige Nummern oder Nummernkombinationen bei erweiterten ET
- Neutrale Bedingungsanzeiger (bleiben während der Bearbeitung unberücksichtigt):
  - Nichtrelevanzanzeiger: -
  - Ausschußanzeiger: /  
(Unterschied zum Nichtrelevanzanzeiger: Darf zum Zwecke des Widerspruchstests nicht durch beliebige konstruktive Bedingungsanzeiger ersetzt werden.)
- Aktionsanzeiger (bzw. Ergebnisanzeiger):  
X (Aktion trifft zu) oder - (Aktion trifft nicht zu)

Entscheidungstabellenumsetzungsprogramme

Angepaßt an eine Auswahl von drei verschiedenen Verarbeitungsmethoden im Zielrechner steht eine Reihe von Entscheidungstabellenumsetzungsprogrammen zum Umordnen, Aufbereiten und Optimieren der vom Anwender eingegebenen Ursprungsform der Entscheidungstabelle zur Verfügung. Als Endprodukt entsteht ein Datenmodul, der im Kern die in den Mikrorechnerspeicher ladbare Objektform der Entscheidungstabelle bildet. Diese Objektform besteht entweder aus Sollmustern, Masken und Aktionsadressen (bei Verarbeitungsverfahren auf der Grundlage des vergleichenden Suchens) oder aus Adresstransformationstabellen (beim Adressrechnungsverfahren). Die Umsetzungsprogramme werden nur während der Phase der Software-Entwicklung benötigt und werden auf dem Mikrorechner-Entwicklungssystem betrieben.

### Entscheidungstabellenprüfprogramme

Während der Erfassung und Umsetzung werden die Entscheidungstabellen auf Syntaxfehler, Widerspruch und Redundanz geprüft. Das Prüfergebnis wird am Bildschirm gemeldet.

### Test- und Simulationsprogramme

Es wird im Mikrorechner-Entwicklungssystem ein Testbett aufgebaut, das die Verarbeitung der Entscheidungstabelle im Zielrechner zu simulieren erlaubt. Mit Hilfe eingegebener Bedingungskombinationen (Schlüsselvariable) läßt sich prüfen, ob die zutreffenden Aktionen in der Entscheidungstabelle richtig aufgerufen werden. Es wird der jeweils erforderliche Laufzeit- und Speicherplatzbedarf im Zielrechner ermittelt und am Bildschirm ausgedruckt.

### Entscheidungstabellenverarbeitungsprogramme für den Zielrechner

In der Betriebsphase arbeitet ein in den Programmspeicher des Zielrechners geladenes Entscheidungstabellen-Verarbeitungsprogramm die im Datenspeicher des Zielrechners abgelegte Objektform der Entscheidungstabelle unter Realzeitbedingungen ab.

Es sind drei verschiedene kurze Verarbeitungsroutinen für unterschiedliche Verarbeitungsmethoden verfügbar, die sich im Speicherplatz- und Laufzeitbedarf sowie in der Änderungsflexibilität unterscheiden. Die Verarbeitungsroutinen umfassen ca. 60 ... 80 Byte und werden bedarfsweise aus der DETAM-Bibliothek abgerufen und als Unterprogramme in das Hauptprogramm des Anwenders eingebunden. Für mehrere Entscheidungstabellen mit gleicher Verarbeitungsmethode reicht eine einzige Verarbeitungsroutine im Programmspeicher aus.

### 3. Die Verarbeitungsverfahren

Bei der Entwicklung von DETAM standen folgende Verarbeitungsverfahren zur Auswahl:

- (1) Entscheidungsbaumverfahren
- (2) Spalten-Masken-Verfahren

### (3) Zeilen-Masken-Verfahren

### (4) Adressrechnungsverfahren (Veinott-Methode)

### (5) Vergleich der Quersumme der gewichteten Wertigkeiten der Bedingungsanzeiger mit der Quersumme der eingegebenen Schlüsselvariablen

### (6) Umwandlung der Entscheidungstabelle in eine Karnaugh-Tabelle (Aufteilung der Bedingungskombinationen in zwei Gruppen und zweidimensionale Zuordnung der Aktionen zu diesen Bedingungskombinationen in einer neuen Tabelle)

Die Standard-Verfahren (1) bis (4) sind in [1], [2] und [3] beschrieben.

Um verschiedene Realisierungsanforderungen im Zielrechner bezüglich Verarbeitungsdauer und Speicherplatzbedarf abdecken zu können, wurden folgende Verfahren ausgewählt:

- Standard-Spalten-Masken-Verfahren (SSM)  
(Vergleichendes Suchen mit Standard-Suchweg von links nach rechts, Regel für Regel)
- Laufzeitoptimiertes Spalten-Masken-Verfahren (OSM)  
(Vergleichendes Suchen mit optimiertem Suchweg durch Umordnung der Regeln und Teilwörter mit dem Ziel, übergeordnet und untergeordnet abzufragende Gruppen von Bedingungsanzeigen zu bilden)
- Adressrechnungsverfahren (ADR)  
(Berechnung der Adresse der zutreffenden Aktion aus der Codewertigkeit der Schlüssel-Teilwörter unter Ausnutzung von Optimierungsmöglichkeiten durch eine günstige Aufteilung in Teilwörter)

Der Anwender muß sich jeweils nach der Eingabe einer Entscheidungstabelle für eines der drei Verfahren entscheiden.

### 4. Analyse-, Test- und Optimierungsmöglichkeiten

Zum Test der Entscheidungstabellenverarbeitung im Zielrechner wird durch DETAM im Dienstrechner ein Testbett erzeugt, welches das Hauptprogramm ersetzt. Der Anwender kann nach Eingabe von Test-Schlüsselvariablen die Ent-



Insgesamt stehen dem Anwender von DETAM folgende Optimierungsmöglichkeiten zur Verfügung:

- Wahl zwischen drei Verarbeitungsverfahren
- Wahl einer geeigneten Aufteilung der Schlüsselvariablen auf Rechner-Teilwörter (z.B. zur Verminderung des Speicherplatzbedarfs beim Adreßrechnungsverfahren oder zur Anpassung an die Gegebenheiten der Rechner-peripherie)
- Auslagerung von Entscheidungstabellen in Untertabellen

## 5. Realisierung von DETAM

Einige Daten über die Realisierung von DETAM sind in Tabelle 2 zusammenge-stellt.

Tab. 2 Realisierung von DETAM

Zielrechner:	Intel 8085/8086
Dienstrechner:	Intel MDS 800 oder Intel MDS 231 mit DD-Diskette
Erforderlicher Speicherausbau im Dienstrechner:	64 kByte
Sprache der DETAM-Dienstrechner- programme:	PL/M-80
Sprache der DETAM-Zielrechner- programme:	ASSEMBLER 80/86
Mögliche Sprachen des Anwender- Hauptprogramms:	PL/M-80, FORTRAN-80, ASSEMBLER 80/86
Betriebssystem:	ISIS II 3.4
Erforderliche Intel-Objektdateien:	ASM 80/86 PL/M-80 SUBMIT LINK LOCATE

## 6. Beispiele zur Anwendung von Entscheidungstabellen für Prozeßsteuerungsaufgaben

### 6.1. Allgemeines

Entscheidungstabellen lassen sich bei der Software-Entwicklung überall dort einsetzen, wo die auszuführenden Funktionen wesentlich durch logische Verknüpfungen bestimmt sind. Solche Zusammenhänge lassen sich in Entscheidungstabellen sehr kompakt und übersichtlich formulieren. Wichtig ist jedoch, daß bereits in frühen Phasen des Software-Entstehungsprozesses alle Funktionen mit umfangreichen logischen Verknüpfungen identifiziert und unmittelbar in Form von Entscheidungstabellen formuliert werden. In Verbindung mit weiterverarbeitenden maschinellen Hilfsmitteln wie DETAM ist die Entscheidungstabellentechnik dann ein wirkungsvolles Planungsinstrument und trägt zur Rationalisierung der Software-Entwicklung bei.

### 6.2. Koordination von Prozeßabläufen

In einer Realzeitumgebung, in der Prozesse (= asynchrone Verwaltungseinheiten des Ablaufgeschehens) durch Ereignisse gesteuert werden, ist je nach der Art des Ereignisses (z.B. durch Hardware oder Software veranlaßte Unterbrechungsanforderungen) und des augenblicklichen Systemzustandes (z.B. Priorität laufender Prozesse) zu entscheiden, welche Prozesse zu aktivieren, zu verdrängen oder zur Ruhe zu setzen sind. Aus der logischen Verknüpfung der Art der auftretenden Ereignisse und herrschenden Systemzustände können aus einer Entscheidungstabelle die notwendigen Aktionen für den Prozeßmultiplexbetrieb abgeleitet werden. Entscheidungstabellen eignen sich in diesem Falle zur Realisierung von Prozeßkoordinatoren, die Teil eines Rechnerbetriebssystems sein können und dann die bekannten Funktionen "scheduling" und "dispatching" übernehmen. Das Schema eines Systems von Entscheidungstabellen zur Prozeßkoordination zeigt Bild 4.

### 6.3. Auftragssteuerung in Rechnerverbundsystemen

Werden in einem Verbund von Rechnern (insbesondere Mikrorechnern) Teilaufgaben verteilt bearbeitet, die zu einer größeren Gesamtaufgabe gehören, so ist eine Auftragssteuerung erforderlich, welche den logischen Zusammenhang im Ablauf der Teilaufgaben koordiniert und die Zuweisung von Teil-

aufgaben zu Rechnern bestimmt. Zur Realisierung einer solchen Globalbetriebs-systemfunktion lassen sich Entscheidungstabellen anwenden.

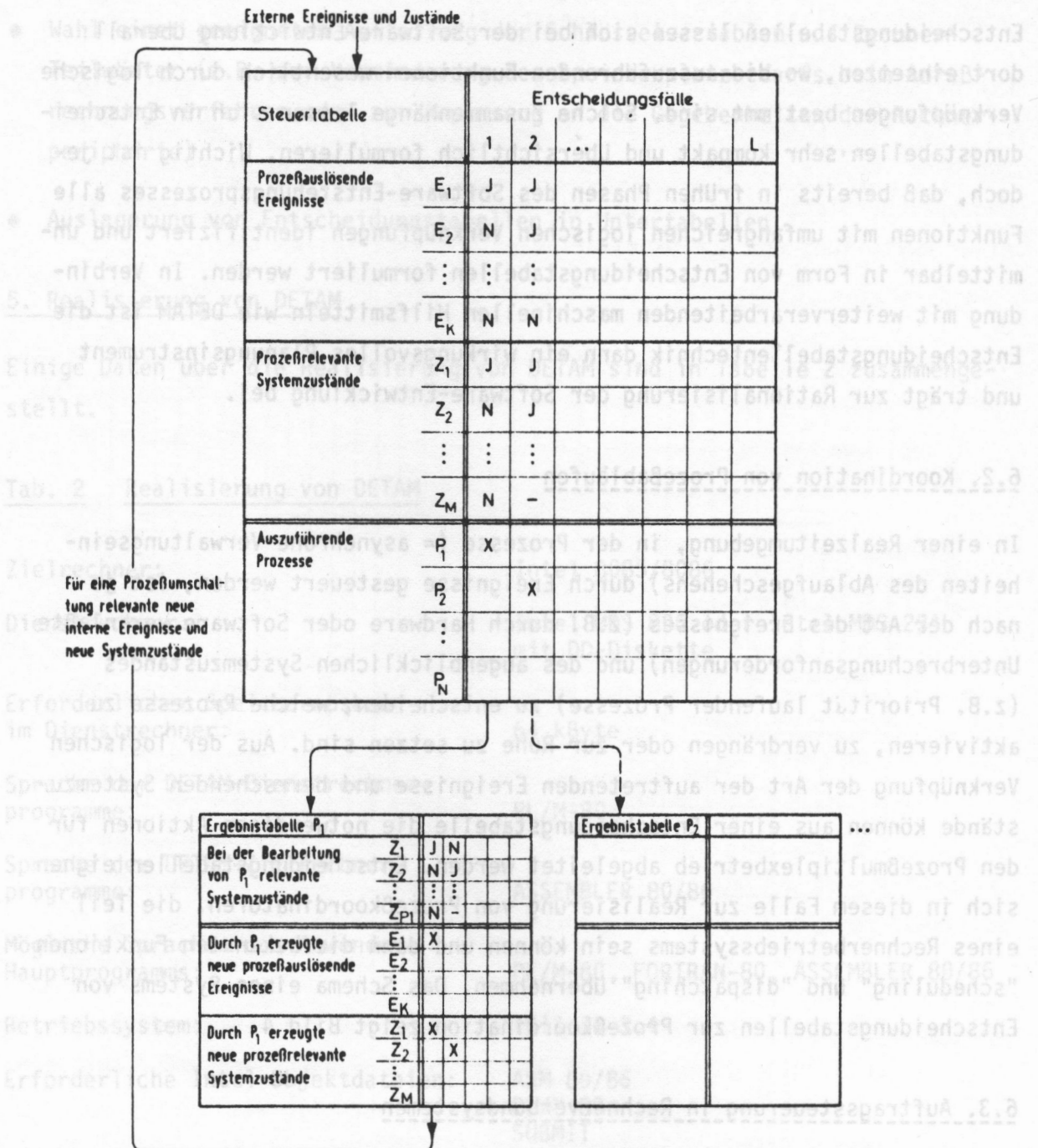
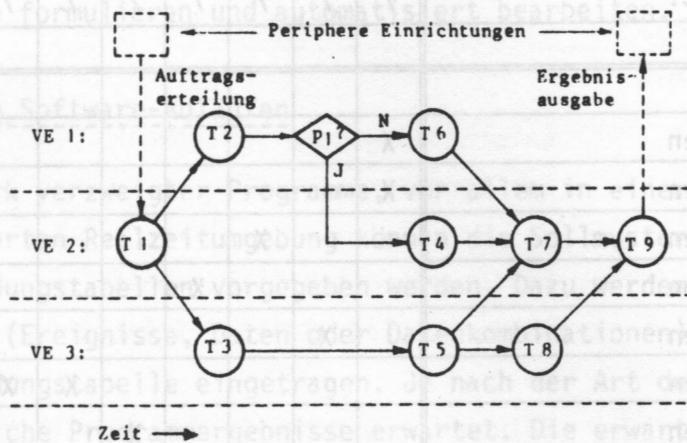


Bild 4 Koordination von Prozessabläufen durch Entscheidungstabellen



Das Beispiel Bild 5 zeigt in Form eines Graphen, wie die Teilaufgaben  $T_i$  einer Gesamtaufgabe in Funktionsteilung auf drei Mikrorechnern (= Verarbeitungseinheiten) ablaufen. Im Beispiel ist vorausgesetzt, daß die Verarbeitungseinheit VE1 den Auftrag zur Bearbeitung der Gesamtaufgabe annimmt und die Ergebnisse wieder zur Peripherie ausgibt, wobei kein neuer Auftrag angenommen wird, bevor der alte nicht vollständig bearbeitet wird. Teilaufgabe T7 kann erst bearbeitet werden, wenn die Ergebnisse von T5 und T4 oder von T5 und T6 vorliegen. Ähnlich kann mit der Bearbeitung von T9 erst begonnen werden, sobald T7 und T8 abgeschlossen sind.



**Bild 5** Darstellung einer Gesamtaufgabe eines verteilten Rechnersystems als gerichteter Graph

( $T_i$  = Teilaufgabe,  $VE_j$  = Verarbeitungseinheit)

Bild 6 zeigt die zugehörige Entscheidungstabelle zur Auftragssteuerung.

Die Entscheidungstabelle läßt sich von DETAM unmittelbar automatisch weiterverarbeiten.

Wie bei allen Schaltwerken oder endlichen Zustandsautomaten durchläuft auch die Auftragssteuerung eine Kette von Systemzuständen. Um bei der wiederholten Abfrage der Entscheidungstabelle (z.B. bei jeder Beendigung einer Teilaufgabe) zu gewährleisten, daß die Steuerung immer beim aktuellen Systemzustand fortgesetzt wird, wird in der Entscheidungstabelle vorbeugend abgefragt, ob die nächste zu bearbeitende Teilaufgabe nicht bereits schon früher erledigt wurde. Damit ist eine Wiederholsperrre realisiert, die erst nach Abschluß des Gesamtauftrags wieder aufgehoben wird.

	1	2	3	4	5	6	7	8
T1 bearbeitet?	J	/	/	/	/	/	/	/
T2 bearbeitet?	N	J	J	/	/	/	/	/
Verzweigungsbed. $p_1$ erfüllt?	/	N	J	/	/	/	/	/
T3 bearbeitet?	N	/	/	J	/	/	/	/
T4 bearbeitet?	/	/	N	/	/	J	/	/
T5 bearbeitet?	/	/	/	N	J	J	J	/
T6 bearbeitet?	/	N	/	/	/	/	J	/
T7 bearbeitet?	/	/	/	/	/	N	N	J
T8 bearbeitet?	/	/	/	/	N	/	/	J
T9 bearbeitet?	/	/	/	/	/	/	/	N
T2 veranlassen	X							
T3 veranlassen	X							
T4 veranlassen			X					
T5 veranlassen				X				
T6 veranlassen		X						
T7 veranlassen						X	X	
T8 veranlassen					X			
T9 veranlassen								X
Grundstellung d. Bearbeitungszustände herbeiführen								X

Bild 6 Entscheidungstabelle zur Auftragssteuerung im Beispiel nach Bild 5

Die Entscheidungstabelle kann durch DETAM soweit automatisch aufbereitet werden, daß sie als Datenmodul in einen der Mikrorechner geladen werden kann. Dieser Mikrorechner kann dann die Auftragssteuerung zentral koordinieren. (Es sei bemerkt, daß auch eine dezentrale Lösung möglich ist, wenn die Entscheidungstabelle in drei kleinere Tabellen aufgeteilt wird, welche auf die drei Mikrorechner verteilt sind.)

Werden mehrere Aufträge überlappend im System bearbeitet (Fließbandprinzip), so ist in den Entscheidungstabellen zusätzlich zu berücksichtigen, welche Teilaufgaben für welchen Auftrag momentan bearbeitet wurden bzw. noch zu bearbeiten sind.

#### 6.4. Steuerung vermittlungstechnischer Vorgänge

In der Fernsprech-, Fernschreib- und Datenvermittlungstechnik sowie in Auskunftssystemen sind Zugriffsberechtigungen und andere Kennungen von Teilnehmern zu prüfen, bevor die Zugriffserlaubnis erteilt wird. Weiterhin ist aus einer Vielzahl von möglichen Verbindungsarten, Belegungsarten, Dienstarten und Systemzuständen in einer Realzeitumgebung die jeweils zutreffende Aktion (z.B. Verbindungsaufbau, Verbindungsabbau, Rückfrage einleiten, Signaltöne an- oder abschalten) zu veranlassen.

Solche Software-Aufgaben lassen sich zu einem großen Teil durch Entscheidungstabellen formulieren und automatisiert bearbeiten.

#### 6.5. Test von Software-Abläufen

Zum Test stark verzweigter Programme, vor allem in einer simulierten ereignisgesteuerten Realzeitumgebung können die Sollmuster des Tests in Form von Entscheidungstabellen vorgegeben werden. Dazu werden die Eingaben für den Testlauf (Ereignisse, Daten oder Datenkombinationen) im Bedingungsteil der Entscheidungstabelle eingetragen. Je nach der Art der Eingaben werden unterschiedliche Programmergebnisse erwartet. Die erwarteten Soll-Ergebnisse werden im Aktionsteil der Entscheidungstabelle eingetragen und dienen zum Vergleich mit den erhaltenen Ist-Werten. Auf dieser Grundlage lassen sich Testautomaten realisieren.

#### 6.6. Test digitaler Schaltungen und Geräte

Ähnlich wie beim Test von Software-Abläufen werden die Testeingaben in Form von Bitmustern oder Kombinationen von Signalzuständen im Bedingungsteil einer Entscheidungstabelle und die erwarteten Soll-Ergebnisse im Aktionsteil spezifiziert und der Reihe nach durchgeprüft.

#### 6.7. Überwachung und Interpretation der Eingaben an Mensch-Maschine-Schnittstellen

Bei der menschlichen Eingabe von Kommandos, Meldungen, Daten und Bedienungsanweisungen an Datenverarbeitungsanlagen sind die Eingaben zu interpretieren, zu überprüfen und gegebenenfalls automatisch richtigzustellen, damit eine fehlerhafte Weiterverarbeitung oder unzulässige Zugriffe ver-



mieden werden. Da die Berücksichtigung aller statthafter und unstatthafter Fälle schnell zu umfangreichen logischen Verknüpfungen führt, ist auch in diesem Falle die Verwendung der Entscheidungstabellentechnik vorteilhaft.

## 7. Quellenhinweise

- [1] Elben, W.: Entscheidungstabellentechnik  
De Gruyter-Verlag, 1973
- [2] Erbesdobler, R.;  
Heinemann, J.;  
Mey, P.: Entscheidungstabellentechnik  
Springer-Verlag, 1976
- [3] Strunz, H.: Entscheidungstabellentechnik  
Hanser-Verlag, Reihe Betriebsinformatik, 1977
- [4] DIN 66241 Entscheidungstabellen für den Informations-  
austausch, 1/79
- [5] Mc Daniel, H.: Decision Table Software  
Brandon/Systems Press Inc., Princeton, New York,  
London, 1970
- [6] Köhler, N.: Technische und ökonomische Folgerungen des  
Entscheidungstabelleneinsatzes  
Online 1976, Heft 10, S. 611-617