

# Rechnerunterstütztes Erfinden

— eine Einführung —

	AN	UE	AR	HS	EN	
OM	OM   DM   EM   ...				KT   BA   ...	
DM					1   0	
AN		1	0	1	1   0	
EM					0   1	
⋮						
UE						
AR						
HS						
EM						

Bezirksneuererzentrum Suhl

**Herausgeber:** Bezirksneuererzentrum Suhl  
**Redaktion:** W. Ball  
**Verfasser:** Dr. sc. techn. Dieter Herring  
**Druck:** Druckerei „Freies Wort“ Suhl  
V-20-15 2 SG 134/15/86 1875  
00800

**Dieter Herrig**

**Rechnerunterstütztes Erfinden - eine Einführung**

**Bezirksneuererzentrum Suhl 1986**

<u>Inhalt</u>	Seite
<b>Vorwort</b>	3
<b>0. Überblick</b>	4
<b>1. Theorie</b>	
<b>1.1. Objekt und Merkmal</b>	
<b>1.1.0. Übersicht</b>	6
<b>1.1.1. Objektbeschreibung</b>	8
<b>1.1.2. Merkmalbeschreibung</b>	10
<b>1.2. Aufgabe und Lösung</b>	
<b>1.2.0. Übersicht</b>	12
<b>1.2.1. Aufgabenfindung</b>	14
<b>1.2.2. Lösungsfindung</b>	16
<b>2. Praxis</b>	
<b>2.1. Hardware</b>	18
<b>2.2. Software</b>	
<b>2.2.0. Übersicht</b>	
<b>2.2.1. HEUREKA - ein Programmrahmen</b>	20
<b>2.2.2. PATPRO1 - ein Patentrechercheprogramm</b>	22
<b>2.2.3. EFFPRO1 - ein Effektkettungsprogramm</b>	26
<b>2.2.4. WIDPRO1 - ein Widerspruchsmerkmalprogramm</b>	34
<b>2.2.5. STAPRO1 - ein Standardsituationsprogramm</b>	42
<b>2.2.6. TREPRO1 - ein Trenddateiprogramm</b>	50
<b>2.2.7. ASSPRO1 - ein Assoziationsprogramm</b>	58
	64
<b>3. Beispiele</b>	
<b>3.0. Übersicht</b>	70
<b>3.1. Feststofftrennung</b>	72
<b>3.2. Rotationsenergiemumformung</b>	74
<b>4. Ausblick</b>	
<b>4.0. Übersicht</b>	78
<b>4.1. STUPRO1 - eine Strukturkonkretisierungsidee</b>	80
<b>4.2. EXPPRO1 - eine Expertensystemidee</b>	84
<b>5. Anhang</b>	
<b>5.1. Programmiersprache BASIC</b>	86
<b>5.2. Programmiersprache dBASE</b>	90
<b>6. Quellen</b>	94

## Vorwort

"Die wissenschaftlich-technische Arbeit ist in allen Bereichen konsequent auf die Erzielung von Spitzenleistungen bei neuen Erzeugnissen und Technologien auszurichten.

... Bis zum Jahre 1990 sollen 85000 - 90000 CAD/DAM-Arbeitsstationen wirksam werden, an denen insbesondere Konstrukteure, Projektanten, Technologen, Formgestalter und weitere ingenieurtechnische Kader tätig sind."

## Direktive zum XI. Parteitag

Als Beitrag zur Verwirklichung dieser Strategie sowie zur weiteren Entwicklung der Erfindertätigkeit und als Anregung für Industriebetriebe, Lehreinrichtungen, Erfinderschulen und Computerclubs soll die vorliegende Broschüre dienen. Der Bezirksvorstand der KDT, die Gewerkschaft Wissenschaft, die AG "Erfindertätigkeit/Schöpfertum" und das Bezirksneuererzentrum des Bezirkes Suhl verfolgen gemeinsam das Ziel, neue Impulse für das kreative Wirken, insbesondere der Jugend, zu geben. Der Kampf um mehr Patente und Erfindungen, die internationales Spitzenniveau darstellen, ist bei der Einführung der Schlüsseltechnologien und der Entwicklung neuer hochwertiger Konsumgüter unerlässlich.

Der Herausgeber erhielt die Anregung für dieses Material durch den Vortrag von Dr.sc.techn. Dieter Herring, Problem-analytiker an der Bezirksnervenklinik Schwerin, auf der ersten Erfinderkonferenz des Bezirkes Suhl im Jahre 1985. Der Verfasser beschäftigt sich seit etwa zwanzig Jahren mit dem kartei- und dateigestützten Konstruieren und seit etwa fünf Jahren mit dem kartei- und dateigestützten Erfinden. Seine Erfahrungen sind in der Broschüre verarbeitet und sollen einem breiten Anwenderkreis vermittelt werden.

B a l l  
Leiter BNZ Suhl

## 0. Überblick

Rechnerunterstützung bei Entwurfs- und Entwicklungsprozessen wird seit etwa dreißig Jahren versucht, ist heute fast schon üblich und als CAD/CAP/CAM wohlbekannt.

Rechnerunterstützung bei Entdeckungs- und Erfindungsprozessen wird seit etwa zwanzig Jahren versucht, ist heute noch nicht üblich und als CAC und CATI kaum bekannt,

Bild 0.1.

In den folgenden Abschnitten wird nun versucht,

- einige Thesen für die Theorie
- einige Programme für die Praxis

aufzustellen und zusammenzustellen.

Mit der Zeit sollen die Thesen zur geschlossenen Theorie und die Programme zum offenen Programmsystem HEUREKA reifen,

Bild 0.2.

Im theoretischen Teil dominieren die Begriffe

- Objekt (als technisches Verfahren und technisches Gebilde)
- Merkmal (als Bestandteil, Eigenschaft und Beziehung),  
denn immer hat der Erfinder ein technisches Objekt zu schaffen,  
"gekennzeichnet dadurch", daß Merkmale etwas Neues, Nützliches  
und Niveauvolles ergeben.

Im praktischen Teil dominieren sehr einfache Programme

- für Kleinstrechner (Personal- und Bürocomputer)
- zum Kombinieren, Konkretisieren und Kreieren.

Zwei Beispiele zeigen, daß die Programme brauchbare Lösungen anregen können.

Ein Ausblick zeigt dann, wie mit Mitteln der künstlichen Intelligenz künftig künstliche Kreativität erzeugbar ist, wie beispielsweise Expertensysteme für das Erfinden entstehen und wirken können.

Zwei Anhänge führen in die verwendeten Programmiersprachen

- BASIC (insbesondere für Rechenaufgaben)
- dBASE (insbesondere für Suchaufgaben)

ein.

Dreißig Quellen dienen als Ideenbelege und sollen zum Weiter- und Bessermachen anregen.

Produktphase		Haupt - ergebnis	Rechnerunterstützung CA ... computer aided ...			
Forschung	Entdeckung	Effekt	CAC	CATI	... C - creativity • (Schöpferkraft)	
	Erfindung	Patent			... TI - technical invention (techn. Erfinden)	
Konstruktion	Entwurf	Zeichnung	CAD	CAD	... D - design (Entwurf)	
Technologie		Arbeits - plan	CAP	CAM	... P - planning (Arb. planung)	
Fertigung , F. - Steuerung		Produkt			... M - manufacturing Fertigung	

Bild 0.1.

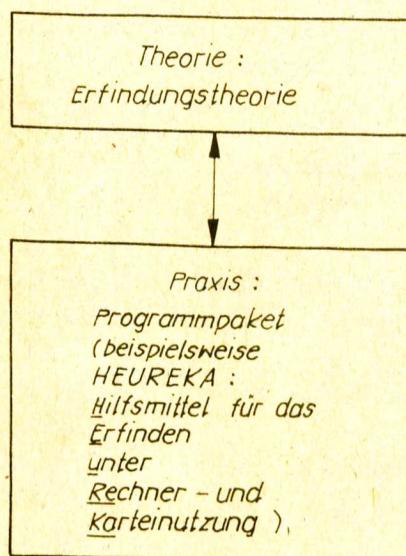


Bild 0.2.

## 1. Theorie

### 1.1. Objekt und Merkmal

#### 1.1.0. Übersicht

Alle Objekte  $O$ , darunter die technischen Objekte

- technische Verfahren (Technologien)

- technische Gebilde (Konstruktionen),

haben Merkmale  $M$ , in denen sie sich

- gleichen  $(M_A = M_B)$

- ähneln  $(M_A \approx M_B, M_A \sim M_B)$

- unterscheiden  $(M_A \neq M_B, M_A \gg M_B, M_A = k \cdot M_B)$ .

Es gibt unendlich viele Merkmale, aber nur drei Merkmalarten:

- Bestandteile ( $M^0$  ist Teil von  $O$ ,  $M^0 \in O$ );

diese Merkmale haben Werte für Existenz und Anzahl:

mit, ohne; viele, wenige; 0, 1, 2, ... usw.

- Eigenschaften ( $M^1$  ist Eigenschaft von  $O$ );

diese Merkmale haben Werte für Grad und Güte:

groß, klein; stark, schwach; 0...1 usw.

- Beziehungen ( $M^2$  ist Beziehung zwischen  $O_a$  und  $O_A$  oder  $O_B$ );

diese Merkmale haben Werte für Ablauf und Anordnung:

vorher, nachher; vor, hinter; Eingang, Ausgang usw.

Alle Objekte  $O$  und alle Merkmale  $M$  sind

- abstrakt  $(O_a = (O_{k1}, O_{k2}, \dots), M_a = (M_{k1}, M_{k2}, \dots),$

$O_a = (M_{a1}, M_{a2}, \dots))$

- konkret  $(O_k = (M_{k1}, M_{k2}, \dots), M_k)$

beschreibbar.

Die Hauptaufgabe des Erfinders besteht nun darin,

- mindestens ein niveauvolles konkretes Merkmal  $M_k$  zu finden,

um damit

- mindestens ein abstraktes Merkmal  $M_a$  nützlich zu beeinflussen

um somit

- mindestens ein neues technisches Objekt  $O$  zu schaffen:

$$! M_a(O) = f(M_k) ,$$

s. Beispiel in Bild 1.1.

Aufgabe	Der Verbrennungsmotor ist zu verbessern.
Objekt O	Verbrennungsmotor
abstraktes Merkmal $M_a$	Effektivität (aus Gebrauchswert und Tauschwert)
konkretes Merkmal $M_k$	Bestandteile ändern: <u>ohne Kurbeltrieb</u> Beispiel: STELZER - Motor 111
Lösungen	
	Eigenschaften ändern: <u>vergrößerte</u> thermische Belastbarkeit durch neue Werkstoffe (Keramik) Beispiel: PARR - Motor 121
	Beziehungen ändern: <u>Eingangsgröße</u> Wasser verbessert Wirkungsgrad Beispiel: LEIPNITZ - Motor 131 (Das Wasser erzeugt einen Kreisprozeß, der ähnlich dem des DIESEL-Motors ist)
	$M_a \uparrow (O) = f(M_k)$ $\text{! Effektivität } \uparrow (\text{Verbr. motor}) = f \left( \begin{array}{l} \text{Anzahl der Bestandteile:} \\ \text{z.B. ohne Kurbeltrieb;} \\ \text{Größe der Eigenschaften:} \\ \text{z.B. wärmebeständige Keramik;} \\ \text{Ablauf der Beziehungen:} \\ \text{z.B. Wasser im Gemisch} \end{array} \right)$

Bild 1.1.

### 1.1.1. Objektbeschreibung

Objekte  $O_k$  werden im einfachsten Falle durch einige ihrer Merkmale  $M_k$  beschrieben:

$$O_k = (M_{ki}, M_{kj}, \dots) .$$

Diese Beschreibung heißt normale Beschreibung:

- ein Objekt
- einige oder alle bekannten Merkmale, die das Objekt hat  
(Gegensatz: inverse Beschreibung, s. Abschnitt 1.1.2.).

Objektteil-Klassen sind abstrakte Merkmale  $M_a$  (Bestandteile der Objektklassen  $O_a$ ) und werden durch diejenigen konkreten Merkmale  $M_k$  beschrieben, die Werte oder Ausprägungen der abstrakten Merkmale  $M_a$  sein können:

$$M_a = (M_{k1}, M_{k2}, M_{k3}, \dots) .$$

Objekt-Klassen sind abstrakte Objekte  $O_a$  und werden durch ihre Objektteil-Klassen  $M_a$  beschrieben:

$$\begin{aligned} O_a &= (M_{a1} (M_{1k1}, M_{1k2}, \dots), \\ &\quad M_{a2} (M_{2k1}, M_{2k2}, \dots), \\ &\quad \dots \quad ). \end{aligned}$$

Es entsteht die morphologische Matrix nach ZWICKY /4/:

Nummern der konkreten Merkmale		0	1	2	3	...
abstrakte Merkmale	konkrete Merkmale					
$M_{a1} = M_1$		$M_{1k0} = M_{10}$	$M_{11}$	$M_{12}$	$M_{13}$	
	$M_2$		$M_{20}$	$M_{21}$	$M_{22}$	$M_{23}$
	$M_3$		$M_{30}$	$M_{31}$	$M_{32}$	$M_{33}$
	.	.				
	.	.				

Beispiele zeigt Bild 1.2.

Objekt	Beschreibung				
Pumpenaggregat , Pumpensatz (Objekt )	Pumpen - aggregat	=	$\left\{ \begin{array}{l} \text{Elektromotor,} \\ \text{Kurbeltrieb,} \\ \text{Kolben} \end{array} \right\}$		
Pumpenantriebe (Objektteil - Klasse )	Pumpen - antriebe	=	$\left\{ \begin{array}{l} \text{Verbrennungsm. (mech.-thermisch),} \\ \text{Elektromotoren (elektromagnet.),} \\ \text{Hydromotoren (hydraulisch),} \\ \dots \end{array} \right\}$		
	Nr. $M_k$	0	1	2	3
	$M_a$ $M_k$				
Antriebs - organ	1	ohne	<u>mech.</u> - <u>therm.</u>	elektro- magn.	hydraul. ...
Über - tragungs - organ	2	ohne	Kurbel - trieb	Zahn - trieb	Riemen - trieb ...
Abtriebs - organ	3	ohne	<u>Kolben</u>	Schaufel Membran	...
STELZER - Motorpumpe bzw. Motorverdichter (Objekt )	$O = \{ M_{11}, M_{20}, M_{31} \}$				

Bild 1.2.

### 1.1.2. Merkmalbeschreibung

Merkmale  $M$  werden im einfachsten Falle durch ihre Objekte  $O$  beschrieben:

$$M = (O_A, O_B, \dots).$$

Diese Beschreibung heißt inverse Beschreibung:

- ein Merkmal
- einige oder alle bekannten Objekte, die das Merkmal haben (Gegensatz: normale Beschreibung, s. Abschnitt 1.1.1.).

Komplexe abstrakte Merkmale  $M_a$  werden durch diejenigen Merkmale  $M$  beschrieben, die Wert oder Ausprägung des abstrakten Merkmals  $M_a$  beeinflussen können:

$$M_a = f(M_1, M_2, M_3, \dots).$$

Es entsteht der Widerspruchsansatz nach THIEL /5/, /6/, mit den Tendenzen für die Werte komplexer abstrakter Merkmale  $M_a$  darstellbar sind:

$$M_a^f = f(M_1^f, M_2^f, M_3^f, \dots)$$

oder normiert

$$M_a^f = f(M_1^f, 1/M_2^f, M_3^f, \dots).$$

Werte von Merkmalen  $M$  werden aus Verknüpfungen von Werten konkreter Merkmale  $M_k$  ermittelt:

$$M = M_1 \circ M_2 \circ M_3,$$

beispielsweise

$$M = c \cdot \frac{M_1 \cdot M_3}{M_2}.$$

Beispiele zeigt Bild 1.3.

Merkmal	Beschreibung
Antriebsleistung $10 < P \leq 100 \text{ kW}$	Antriebs - leistung = $\begin{cases} \text{Verbrennungsmotor, Typ ...} \\ \text{Elektromotor, Typ ...} \\ \text{Hydromotor, Typ ...} \\ \dots \end{cases}$
Effektivität <i>(das ist ein komplexes abstraktes Merkmal)</i>	Effektivität = $f(\text{Leistung, Masse, ...})$ $E = f(P, m, \dots)$ $E^f = f(P^f, m^f, \dots)$ oder $E^f = f(P^f, \gamma m^f, \dots)$ $E = c \cdot \frac{P}{m} \dots$

Bild 1.3.

## 1.2. Aufgabe und Lösung

### 1.2.0. Übersicht

Aufgaben können nur dann entstehen, wenn Istzustand und Sollzustand in mindestens einem Merkmal nicht übereinstimmen:

$$M_{\text{ist}} \neq M_{\text{soll}} .$$

Diese Situation kann nun in Aufgaben verschiedener Güte überführt werden:

- Unterschiedsaufgaben
  - durch Finden und Formulieren des Unterschieds zwischen Istzustand und Sollzustand
    - (einfache Aufgaben, kaum erfinderische Lösungen)
- Gegensatzaufgaben
  - durch Finden und Formulieren des gegensätzlichen Unterschieds zwischen offensichtlich möglichem Sollzustandsansatz und gefordertem Sollzustand
    - (einfache Probleme, oft erfinderische Lösungen)
- Widerspruchsaufgaben
  - durch Finden und Formulieren des ausschließenden und gleichzeitig einschließenden gegensätzlichen Unterschieds zwischen offensichtlich möglichem Sollzustand und idealen Sollzustand
    - (schwierige Probleme, fast zwangsläufig erfinderische Lösungen).

Lösungen sind dann entstanden, wenn Sollzustand und neuer Istzustand in mindestens wesentlichen Merkmalen übereinstimmen:

$$M_{\text{soll}} = M_{\text{ist}} .$$

Diese Situation kann mit Lösungen verschiedener Güte erreicht werden:

- Gleichheitslösungen
  - durch Finden und Wiederverwenden von Objekten
    - (erfinderisch durch Applizieren: "altes Objekt - neue Umgebung")
- Ähnlichkeitslösungen
  - durch Finden und Abwandeln von Objekten und Teillobjekten
    - (erfinderisch durch Kombinieren und Variieren)
- Originallösungen
  - durch Finden und Verwandeln von Objekten und Merkmalen
    - (erfinderisch durch Kreieren)

Beispiele zeigt Bild 1.4.

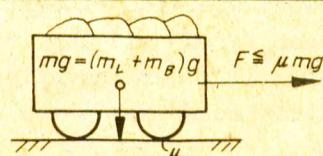
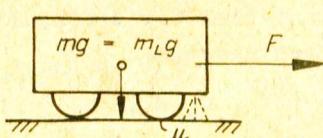
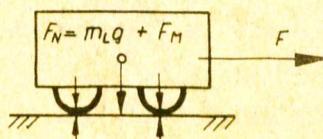
Situation	Grubenlokomotiven sind nicht hinreichend material- und energieökonomisch	
Aufgaben	Beispiele	
Unterschieds-aufgabe	Lokomotive muß - etwas leichter und / oder - etwas zugstärker werden, <u>aber „mehr geht nicht.“</u>	
Gegensatz-aufgabe	Lokomotive muß - viel leichter oder - viel zugstärker werden, <u>aber „beides geht nicht“, weil</u> größere Zugkraft auch schwerere Organe (Antriebsorgane, Stützorgane) erfordert.	
Widerspruchsaufgabe	Lokomotive muß - viel leichter und - viel zugstärker werden, <u>obwohl</u> die Zugkraft widersprüchlich von der Masse abhängt: - geringe Masse - geringe Trägheit - große Zugkraft, - große Masse - große Radkraft - große Zugkraft.	
Lösungen	Beispiele für die Lösung der Widerspruchsaufgabe	
Gleichheits-Lösung	Leichtbau bei Herstellung, Ballast bei Nutzung ( $m$ variabel)	
Ähnlichkeits-Lösung	Leichtbau bei Herstellung, streuen beim Anfahren ( $\mu$ variabel)	
Original-Lösung	Leichtbau Räder als Elektromagnete, einschalten beim Anfahren! ( $g$ „variabel“ durch Zusatz-Feld)	

Bild 1.4.

### 1.2.1. Aufgabenfindung

Aufgaben finden Leiter und Erfinder bei der

- Untersetzung und Umsetzung der Beschlüsse und Pläne  
(z.B. Material-, Energie- und Informationsökonomie)
- Analyse der menschlichen Bedürfnisse  
(z.B. Nahrung, Kleidung, Wohnung, Gesundheit, Verteidigung)
- Analyse der gesellschaftswissenschaftlichen Entdeckungen  
(z.B. dialektische Gesetze)
- Analyse der naturwissenschaftlichen Entdeckungen  
(z.B. physikalische, chemische, biologische Effekte)
- Analyse der technikwissenschaftlichen Entdeckungen  
(z.B. Funktion-Struktur-Analyse und -Synthese)
- Analyse der technischen Trends  
(z.B. Funktionsintegration, Strukturflexibilität)
- Analyse der technischen Objekte  
(z.B. Gebrauchswert-Tauschwert-Verhältnis, Mängel).

Aufgaben formulieren Leiter und Erfinder am besten

- mindestens als Unterschiedsaufgaben
- möglichst als Widerspruchsaufgaben, und zwar
  - zunächst informal (verbal) nach der Satzform  
"Ein Objekt O soll so geschaffen werden, daß die Merkmale  $M_i$  bzw.  $M_j$  existieren bzw. verbessert werden, obwohl sie sich widersprüchlich beeinflussen, denn ..."
  - danach formal mit den drei Formeln:  
Feststellen der Abhängigkeit eines geforderten komplexen abstrakten Merkmals von mehreren elementaren Merkmalen:

$$M_a \not\models (0) = f(M_1 \not\models, M_2 \not\models, \dots, M_i \not\models, M_j \not\models, \dots, M_n \not\models) \quad (1)$$

Feststellen der Abhängigkeiten elementarer Merkmale untereinander:

$$M_i \not\models = f_i(M_j \not\models) \quad (2)$$

Feststellen der Auswirkung auf das komplexe Merkmal:

$$M_i \not\models = f_i(M_j \not\models) \quad (3a)$$

und somit im Widerspruch zu (1) auch

$$M_a \not\models (0) = f_j(M_j \not\models) \quad (3b)$$

Beispiele zeigt Bild 1.5.

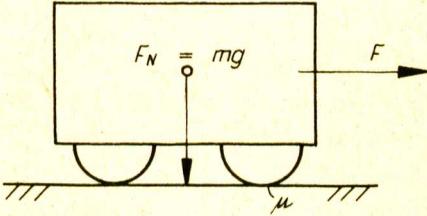
Situation	<p>Grubenlokomotiven sind nicht hinreichend material- und energieökonomisch.</p> <p>Aufgaben - formulierung als Widerspruchs - aufgabe</p> <p> <math>E</math> — Effektivität  <math>F</math> — Zugkraft  <math>F_N</math> — Normalkraft  <math>g</math> — Erdbeschleunigung  <math>m</math> — bewegliche Masse  <math>\mu</math> — Reibungskoeffizient     </p>  <p>Es ist</p> $\begin{aligned} E &= f(F, m, \dots) \\ Ef &= f(F^f, m^f, \dots) \\ Ef &= f(F^f, 1/m^f, \dots) \end{aligned} \quad (1)$ $\begin{aligned} F &= f_1(m), \\ &\text{nämlich} \\ F &\leq \mu \cdot m \cdot g, \\ &\text{also} \\ F^f &= f_1(m^f), \\ &\text{also auch} \\ F^f &= f_1(1/m^f) \end{aligned} \quad (2)$ <p>und somit im Widerspruch zu (1) auch</p> $Ef = f_2(1/m^f) \quad (3a)$ $(3b)$
Lösungs - ansatz	<p><math>m</math> und <math>F</math> sind widersprüchliche Merkmale;  <math>m</math> ist verringerbar und <math>F</math> ist erhöhbar, wenn</p> <ul style="list-style-type: none"> <li>• <math>\mu</math> wächst (Extrem: Zahnradbahn)</li> <li>• <math>g</math> „wächst“ (Zusatz - Kraft, Zusatz - Feld).</li> </ul>

Bild 1.5.

### 1.2.2. Lösungsfindung

Lösungen finden Erfinder durch

- Recherchieren

und Wiederverwenden in Gleichheitslösungen, z.B.

- gegeben : Beziehungsmerkmale  $M^2 = M_{\text{soll}} = M_A$
- gefunden : Objekt  $O_B$  mit Beziehungsmerkmalen  $M_B$
- verwendet: Objekt  $O_B$  als  $O_A$  mit Beziehungsmerkmalen  $M_A$   
(Applizieren)

- Recherchieren

und Abwandeln zu Ähnlichkeitslösungen, z.B.

- gegeben : Merkmale  $M_{\text{soll}} = M_A$
- gefunden : abstraktes Objekt  $O_{AB}$  mit allen  $M_{AB}$  und  
mit einigen konkreten Merkmalen  $M_{KB}$  für  $M_{AB}$
- abgewandelt: Objekte  $O_A$  mit verschiedenen Belegungen  $M_A$   
(Kombinieren, Strukturkonkretisieren)

oder

- gegeben : Merkmale  $M_{\text{soll}} = M_A$
- gefunden : Objekt  $O_B$  mit Merkmalen  $M_B \approx M_A$
- abgewandelt: Objekt  $O_A$  mit Merkmalen  $M_A$   
(Variieren im engeren Sinne)

oder

- gegeben : Merkmale  $M_{\text{soll}} = M_A$  und abstraktes Objekt  $O_{AA}$
- gefunden : Vorschrift für das Konkretisieren/Komplettieren
- abgewandelt: Objekt  $O_A$   
(Konkretisieren, Komplettieren)

- Kreieren

durch Verwandeln zu Originallösungen, z.B.

- gegeben : Merkmale  $M_{1\text{soll}}$  und  $M_{2\text{soll}}$ , die sich widersprechen
- gefunden : Regel, Vorschrift, Formel zur Widerspruchslösung
- verwandelt: Objekt  $O_A$  mit  $M_1$  und  $M_2$   
(Merkmalkreieren)

oder

- gegeben : Beziehungsmerkmale  $M_{Esoll}$  und  $M_{Asoll}$  (Eingang/A.)
- gefunden : abstrakte Teillobjekte mit  $M_{Esoll}$  und/oder  $M_{Asoll}$
- verwandelt:  $O_A$  als Punkt, Kette, Ring, Baum oder Netz  
(Strukturkreieren mit Effekten, Prinzipien usw.)

oder

- gegeben : Merkmal  $M_{\text{soll}}$ ; gefunden : Objekt  $O_B$  mit  $M_{\text{soll}}$
- verwandelt:  $O_A$  mit  $M_{\text{soll}}$ ; (Objektkreieren, Assoziieren).

Beispiele zeigt Bild 1.6.

Lösungsfindung		Beispiele											
Gleichheitslösung	Applizieren	$O_B$	$O_A$										
Ähnlichkeitslösung	Kombinieren	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Nr. M<sub>k</sub></td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>M<sub>a</sub> / M<sub>k</sub></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> An. 1 ohne dreh. schieb. allg. Üb. 2 ohne dreh. schieb. <u>allg.</u> Ab. 3 ohne dreh. <u>schieb.</u> <u>allg.</u>	Nr. M <sub>k</sub>	0	1	2	3	M <sub>a</sub> / M <sub>k</sub>					<b>Kühlschrank</b>  <b>Wärmepumpe</b>   <i>z.B.</i> $O_A = \{M_{11}, M_{23}, M_{32}\}$
Nr. M <sub>k</sub>	0	1	2	3									
M <sub>a</sub> / M <sub>k</sub>													
Variieren (i.e.S.)													
Konkretisieren, Komplettieren													
Originallösung	Merkmalkreieren	Leistung soll verbessert werden! Masse steht dem entgegen!	Vorschrift: Nutze Gegen- oder Zusatzkräfte! Beispiel: Grubenlok										
	Strukturkreieren												
	Objektkreieren, Assoziieren	Reptil: Zunge schnellt vor und erbeutet Insekt.	Plattspeicher: Zugriffskamm „schnellt vor und erbeutet“ Information.										

Bild 1.6.

## 2. Praxis

### 2.1. Hardware

Rechnerunterstütztes Erfinden erfordert als Arbeitsmittel

- Hardware, also technische Gebilde:

- Geräte zur Informationseingabe, -speicherung, -ausgabe  
(Rechnerperipherie)
- Geräte zur Informationsverarbeitung  
(Rechner)

- Software, also textliche Gebilde:

- Texte aus Aufforderungen (Szenarien, Programme)
- Texte aus Aussagen (Daten),

Bild 2.1.

Rechnerperipherie gibt es für

- Ein- und Ausgabe

- lautlich (eindimensional auf t-Achse)
- schriftlich (eindimensional auf x-Achse)
- grafisch (zweidimensional in x-y-Ebene)
- körperlich (dreidimensional im x-y-z-(t)-Raum)

- externe Speicherung

- Band, Kassette (eindimensional, sequentieller Zugriff)
- Platte, Diskette (zweidimensional, wahlfreier Zugriff).

Rechner gibt es als

- Kleinstrechner (Heim-, Personal-, Bürocomputer),  
z.B. PC A 1715, BC A 5120
- Kleinrechner (Arbeitsplatz-, Stationsrechner),  
z.B. AC A 7100, KBR A 64xx
- Mittelrechner, z.B. EC 1055
- Großrechner.

Rechnerunterstütztes Erfinden erfordert möglichst

- Rechnerperipherie als

- Bildschirm, schriftlich mit 80 Zeichen/24 Zeilen
- Drucker mit etwa 1 Zeile/s und Format A4 oder A3
- Disketten mit zusammen etwa 1 MByte  
(1 M = 1024 K = 1024 x 1024 ≈ 1 Mill., 1 Byte = 1 Zeichen)

- Rechner mit

- Hauptspeicherkapazität von etwa 64 KByte
- Rechengeschwindigkeit von etwa 100 KOPS  
(1 OPS = 1 Operation pro Sekunde).

Diese Merkmale werden von Personal- und Bürocomputern erfüllt  
(PC A 1715, BC A 5120) und von Arbeitsplatzcomputern übererfüllt,  
s.a. Bild 2.2.

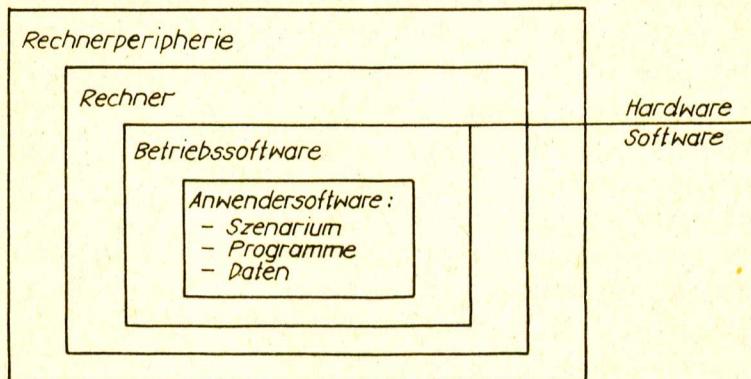


Bild 2.1.

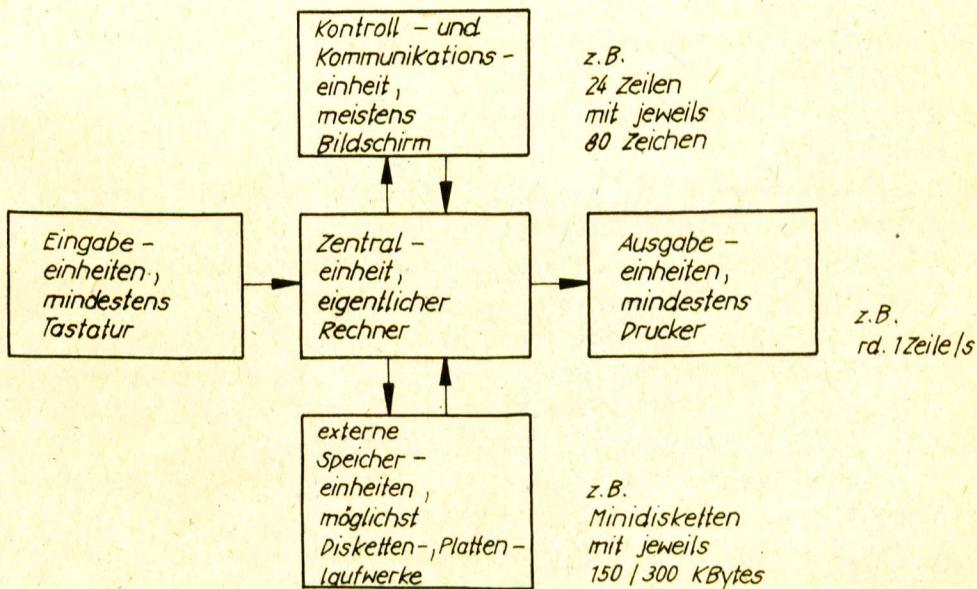


Bild 2.2.

## 2.2. Software

### 2.2.0. Übersicht

Hardware, die "harte Ware" ist potentiell universell, ist also Mittel zur Lösung aller Aufgaben.

Software, die "weiche Ware" ist aktuell spezifizierend, ist also Mittel zur Lösung jeweils einer Aufgabe oder einer Aufgabenklasse.

Software gibt es als

- Betriebsssoftware zur Verknüpfung von
  - . Rechner und Rechnerperipherie (Betriebssystem i.e.S.)
  - . Mensch und Maschine (Dialogsystem)
  - . Hardware und Software (Sprachsystem)
- Anwendersoftware als
  - . Szenarium (Programmrahmen, Szenenbasis)
  - . Programme (Programmbibliothek, Programmbasis)
  - . Daten (Dateien, Datenbasis),

Bild 2.3. nach /8/ und Bild 2.4.

Anwendersoftware sind Texte in Sprachen, für die das Betriebssystem i.w.S.

- verstehende, interpretierende (Interpreter)  
und/oder
- übersetzende, compilierende (Compiler)

Programme enthält.

Für Kleinstrechner gibt es viele, darunter zwei besonders wichtige, weltweit verbreitete Sprachen:

- BASIC (Beginner's All Purpose Symbolic Instruction Code), insbesondere für das Rechnen, z.B.  
`1000 LET verg=k1 * nutz/100 + k2`  
(Zeilennummer, es sei die Vergütung aus dem Nutzen zu errechnen), Einführung s. Abschnitt 5.1.
- dBASE (Data Base, DDR-Produkt heißt REDABAS), insbesondere für das Suchen, z.B.  
`LOCATE FOR land='DD' .AND. .patnr > 250000`  
(lokalisieren, suche DDR-Patente neueren Datums), Einführung s. Abschnitt 5.2.

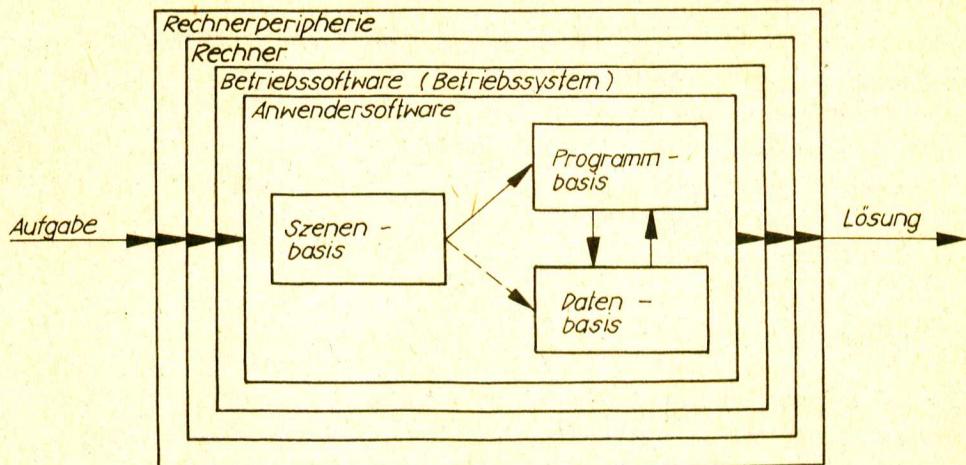


Bild 2.3.

Anwender - software	Zweck , Funktion	Beispiel
Szenarium (Szenen - basis )	Was soll getan werden ?	Erfindungsszenario HEUREKA1 bietet an : - Assoziieren (A) - Kombinieren (K) usw. Nutzer wünscht : A . Das Szenarium ruft das Programm ASSPR01 auf .
Programme (Programm - basis )	Wie soll es getan werden ?	Assoziationsprogramm ASSPR01 ermittelt Zufallszahlen für die Suche. Das Programm ruft die Datei ASSDAT1 auf .
Daten (Daten - basis )	Womit soll es getan werden ?	Assoziationsdatei ASSDAT1 wird durchsucht . Die Daten werden ausgegeben .

Bild 2.4.

### 2.2.1. HEUREKA - ein Programmrahmen

Ein einfacher Programmrahmen, ein einfaches Szenarium für die Verknüpfung von Programmen und Dateien (sowie Karteien), die das Erfinden unterstützen, ist ein Menüangebot der Art "Heureka1 enthält verschiedene Programme:

A - Assoziationsprogramm

K - Kombinationsprogramm

R - Rechercheprogramm

...

Wähle aus (A, K, R, ...) !"

Aber:

"Jede Aufgabe findet ihren Meister" - und künftig sollte jede Aufgabe beim rechnerunterstützten Erfinden auch das Programm finden, das die besten Lösungsansätze liefern kann.

Ein besserer Programmrahmen sollte also

- den Vorgang des Erfindens
- den Zustand der Aufgabe

berücksichtigen.

Der Erfindungsvorgang ist nach Grundlagen, die WOJDZIAK, HERRIG/MÜLLER und HERRIG /8/ lieferten, von HERRLICH /9/ modelliert worden:

Der Weg von der Aufgabe zur Lösung führt

- bei Abstraktion der Funktion
  - : von der Aufgabe über das Gebildeprinzip
  - . zum Verfahrensprinzip und zum Widerspruch
- bei Übergang vom Objektrand zum Objektganzen
  - . von einem lösungsfernen Zustand
  - . zu einem lösungsnahen Zustand
- bei Konkretion der Struktur
  - . von der Widerspruchslösung und vom Verfahrensprinzip
  - . zum Gebildeprinzip und zur Lösung.

Ein Beispiel - die Erfindung und Entwicklung eines Wendegetriebes - zeigt Bild 2.5.

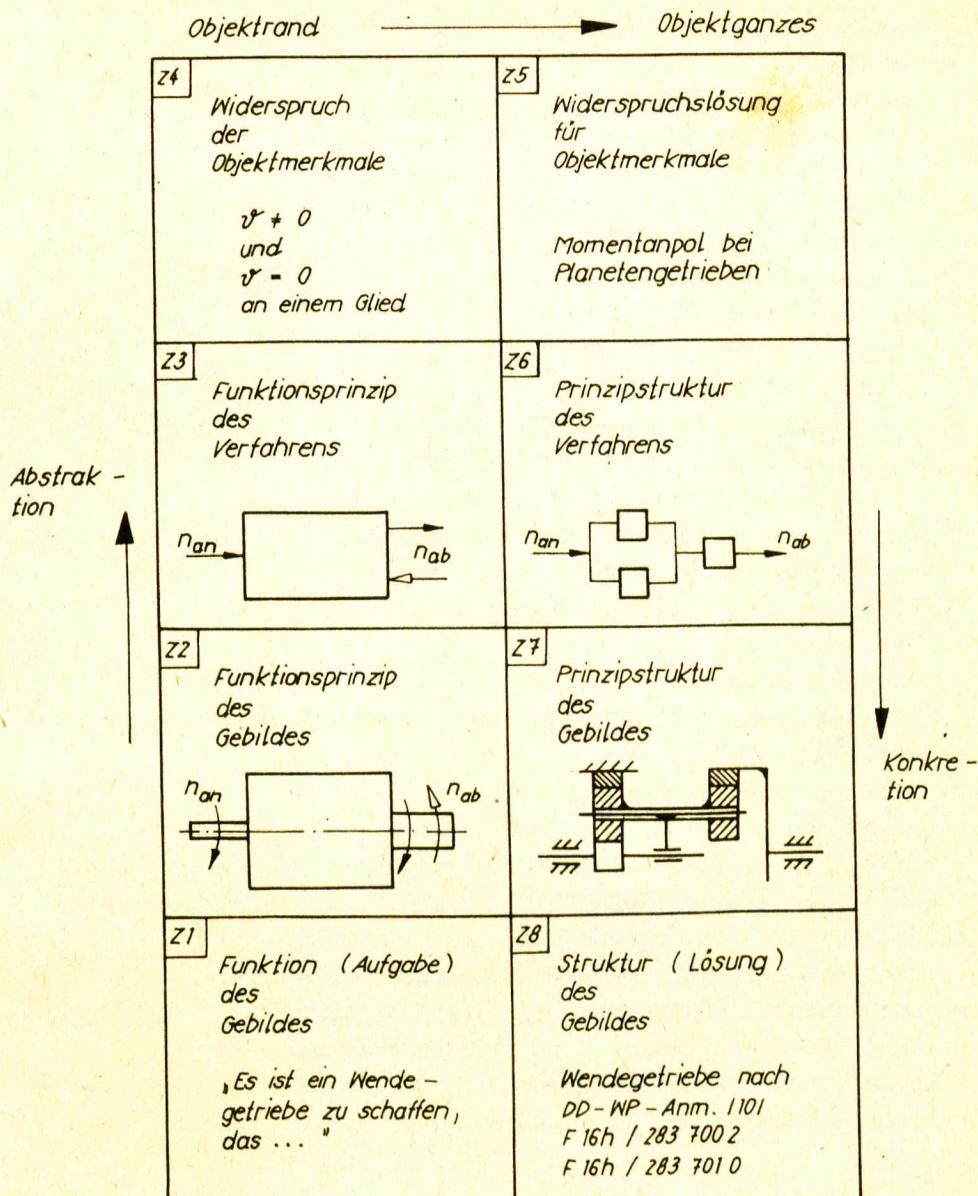


Bild 2.5.

Den Aufgabenzustand kennzeichnen beispielsweise

- die Aufgabengüte
  - . Gleichheitsaufgabe
  - . Ähnlichkeitsaufgabe
  - . Widerspruchsaufgabe
- die Aufgabenpräzision
  - . unklare Aufgabe
  - . klare Aufgabe
- die Aufgabenabstraktion
  - . abstrakt als Trendaufgabe
  - . abstrakt als Stoff-Feld-Aufgabe
  - . abstrakt als Merkmal-Widerspruchsmerkmal-Aufgabe
  - . abstrakt als Eingang-Ausgang-Aufgabe
  - . konkret als Patentrecherche-Aufgabe.

Der Zusammenhang zwischen Erfindungsvorgang und Aufgabenzustand ist noch nicht aufgeklärt, so daß der vorläufige Programmrahmen HEUREKA die Programmauswahl nach der

- Aufgabenpräzision
  - Aufgabenabstraktion
    - (und damit ungefähr nach den Zuständen beim Erfindungsvorgang)
- unterstützt,

s. Bild 2.6.

Die Programme

- ASSPRO1 und PATPRO1 sind BASIC-Programme
- TREPRO1, STAPRO1, WIDPRO1 und EFFPRO1 sind dBASE-Programme.

Alle Programme sind ausdrücklich

- kurz gehalten (eine Seite, höchstens zwei Seiten)
- ohne Komfort (keine Eingabekontrollen usw.)
- nicht optimiert (keine Vielfachschachtelungen usw.).

Alle Programme sind verbesserbar, und alle Dateien sind erweiterbar; das geschieht bereits bei ...PRO2 und ...DAT2.

WENN Aufgabe formuliert . . .	. . . DANN nimm Programm . . .
beliebig abstrakt , unklar , allgemein	ASSPR01 : Assoziationsprogramm : Zufallszahlen weisen auf Objekte mit Merkmalen
abstrakt , allgemein	TREPR01 : Trenddateiprogramm : Entwicklungsrichtungen von Objekten als Regeln
abstrakt als Stoff - Feld - Aufgabe	STAPR01 : Standardsituationsprogramm : Veränderung der S - F - Systeme als Regeln
abstrakt als Merkmal - Widerspruchsmerkmal - Aufgabe	WIDPR01 : Widerspruchsmerkmalprogramm : Lösen von W. bei Merkmal - paaren als Regeln
abstrakt als Eingang - Ausgang - Aufgabe	EFFPR01 : Effektkettenprogramm : naturrechtliche Effekte wandeln Eing. in Ausgänge
konkret	PATPR01 : Patentrechercheprogramm : Patente einzelner Klassen erhalten präzise Klassifizierung

Bild 2.6.

### 2.2.2. PATPRO1 - ein Patentrechercheprogramm

#### a) Programmidee

PATPRO1 unterstützt das Recherchieren nach Objekten mit Merkmalen:

Jedes Patent hat

- eine Patentnummer und ist damit identifizierbar
- eine Patentklassennummer und ist damit klassifizierbar.

Erweitert man nun die Patentbeschreibung in jeder Klasse um 10 bis 15 Merkmale, also die Klassifikationsvorschrift um 10 bis 15 Stellen, so führt die erweiterte Klassifizierungsnummer beinahe stets zur Identifikation des Patents.

Auf diese Weise können

- Patentinhalte numerisch gespeichert werden
- Patentinhalte einfach verglichen werden
- Patentideen auf Neuheit geprüft werden.

Beispiel:

Die Patentklasse B 66 d 1/36 (und 38 sowie 39) betrifft die Seil- und Kettenführungen für geordnetes, lagengerechtes Auf- und Abwickeln bei Trommeln. Eine morphologische Matrix aus

- 14 abstrakten Merkmalen
- durchschnittlich 5 konkreten Merkmalen je abstraktes Merkmal

lässt die Speicherung von  $5^{14} \approx 10^{10}$  (!) verschiedenen Wickelinrichtungen zu (patentiert sind etwa  $10^2$  Wickelprinzipien), Bild 2.7.

#### b) Programmdaten

PATDAT1 enthält

- Patentklasse, -art, -nummer, -titel und Anmeldejahr
- Patentinhalt als 14-stellige Nummer,  
s. Bild 2.8.

(Mit "!" sind Merkmale gekennzeichnet, die deutlich vom Stand der Technik abweichen; Objekt 6 ist eine noch nicht patentierte Idee).

Bearbeitername: -  
 Objektnummer : 0  
 Patentklasse :  
 Land-Patentart:  
 Patentnummer :  
 Patenttitel :  
 Anmeldejahr :  
 Patentinhalt :  
 Schluesselnummer: 0 - null, ohne  
                   6 - kombiniert  
                   7 - beliebig  
 Trommelantrieb : ...1... ...2... ...3... ...4... ...5...  
 - Bewegungsricht: drehen schieb schraub taumeln pendeln  
 - Kraft auf Seil: Zug Druck Biegung Verdreh Scherung  
 Seilfuehrung :  
 - Fuehrungsort : bei Tr. an Tr. vor Tr. hint.Tr.  
 - Freiheitszahl : 1        2        3        4        5  
 - Freiheitsart : drehen schieb schraub taumeln pendeln  
 - Schluszzahl : 1        2        3        4        5  
 - Schluszzart : Kraft Form Stoff  
 - Schluszobjekt : Oeffng Rolle Rad Schlitten  
 - Lagerung : starr federnd bewegl  
 Wickelsteuerung : ...1... ...2... ...3... ...4... ...5...  
 - Meszort : bei Tr. an Tr. vor Tr. hint.Tr.  
 - Meszart : mechan elektr elektro optisch akust  
 - Uebertragung : mechan elektr elektro optisch akust  
 - Stellart : stereo hydro pneumo elektr sonst  
 - Steuerungsart : Z-Lauf S-Kette R-Kreis

Bild 2.7.

Merkmale	Objekt 1	Objekt 2	Objekt 3	Objekt 4	Objekt 5	Objekt 6	Obj.
M <sub>1</sub>	drehend (1)	1	1	1	1	2	
M <sub>2</sub>	ziehend (1)	1	1	1	1	2	
M <sub>3</sub>	vor Tromm. (3)	3	3	1!	2	4	
M <sub>4</sub>	1 1	1	2!	2	2	3	
M <sub>5</sub>	schraubend (3)	2!	4!	6	5!	6	
M <sub>6</sub>	1 (1)	1	1	1	2	3	
M <sub>7</sub>	Formschl. (2)	2	6!	2	2	2	
M <sub>8</sub>	Öffnung (1)	6!	2	2	2	3	
M <sub>9</sub>	starr (1)	1	1	1	1	3	
M <sub>10</sub>	ohne (0)	0	0	0	0	1	
M <sub>11</sub>	ohne (0)	0	0	0	0	1	
M <sub>12</sub>	mechan. (1)	1	1	1	1	2	
M <sub>13</sub>	mechan. (1)	1	1	1	1	2	
M <sub>14</sub>	Zwangslauf (1)	1	3	1	1	3	
Patent :	Stand d.Techn.	DD-AP 152 521	GB ...	DE-OS 23 64 919	PD-WP 129 894	?	
Skizze:							
		Hülltrieb		Profilrolle als Andrückrolle	kleine Rollen, großer Radius		

Bild 2.8.

### c) Programmbeschreibung

PATPRO1 ist ein einfaches Musterprogramm für die Patentklasse B 66 d 1/36. Deshalb ist die Klassifikationsvorschrift direkt in das Programm, nicht jedoch in eine Datei aufgenommen worden.

PATPRO1 ist ein BASIC-Programm, s.a. Anlage 5.1.

Patentrecherchen lassen sich aber auch gut mit dBASE ausführen.

PATPRO1 besteht aus

- einem Vorspann zum Dimensionieren der Variablen und zum Definieren der Konstanten
- der Szene 1: Vorstellung
- der Szene 2: Dateiaufbau
- der Szene 3: Dateinutzung
- der Szene E: Endebehandlung.

Vorspann:

- Hauptspeicherbereiche löschen (CLEAR)
- Variable dimensionieren (DIM)
  - . S-Variable betreffen Texte für Standardangaben
  - . T-Variable betreffen Texte für Klassifizierungsangaben
  - . M-Variable betreffen Werte der Standardangaben
  - . N-Variable betreffen Werte der Klassifizierungsangaben
- Konstante definieren (LET)
  - . S-Konstante betreffen Texte für Standardangaben
  - . T-Konstante betreffen Texte für Klassifizierungsangaben
  - . ZV-Konstante betrifft das Zeilenlöschen  
(24 Leerzeilen)
- Bildschirm löschen (PRINT).

Szene 1: Vorstellung

- Programm kurz beschreiben (PRINT)
- Weiterarbeit erfragen (INPUT)
- Bildschirm löschen (PRINT)
- Menü anbieten (PRINT)
- Menüauswahl (INPUT)
- Programm verzweigen (IF ...),

Bild 2.9.

```
1000 REM Programm PATPRO1
1010 CLEAR
1020 DIM S$(10), T$(20), M$(20), N$(20)
1030 LET S$(1) = "Bearbeitername: "
1040 LET S$(2) = "Objektnummer : "
1050 LET S$(3) = "Patentklasse : "
1060 LET S$(4) = "Land-Patentart: "
1070 LET S$(5) = "Patentnummer : "
1080 LET S$(6) = "Patenttitel : "
1090 LET S$(7) = "Anmeldejahr : "
1100 LET S$(8) = "Patentinhalt : "
1110 LET T$(1) = "Schluesselnummer: 0 - null, ohne"
1120 LET T$(2) = "6 - kombiniert"
1130 LET T$(3) = "7 - beliebig"
1140 LET T$(4) = "Trommelantrieb : ...1... ...2... ...3... ...4... ...5... "
1150 LET T$(5) = "- Bewegungsricht: drehen schieb schraub taumeln pendeln"
1160 LET T$(6) = "- Kraft auf Seil: Zug Druck Biegung Verdreh Scherung"
1170 LET T$(7) = "Seilfuehrung :"
1180 LET T$(8) = "- Fuehrungsort : bei Tr. an Tr. vor Tr. hint.Tr. "
1190 LET T$(9) = "- Freiheitszahl : 1 2 3 4 5 "
1200 LET T$(10) = "- Freiheitsart : drehen schieb schraub taumeln pendeln"
1210 LET T$(11) = "- Schluszzahl : 1 2 3 4 5 "
1220 LET T$(12) = "- Schluszart : Kraft Form Stoff"
1230 LET T$(13) = "- Schluszobjekt : Oeffng Rolle Rad Schlitten"
1240 LET T$(14) = "- Lagerung : starr federnd bewegl"
1250 LET T$(15) = "- Wickelsteuerung : ...1... ...2... ...3... ...4... ...5... "
1260 LET T$(16) = "- Meszort : bei Tr. an Tr. vor Tr. hint.Tr. "
1270 LET T$(17) = "- Meszart : mechan elektr elektro optisch akust"
1280 LET T$(18) = "- Uebertragung : mechan elektr elektro optisch akust"
1290 LET T$(19) = "- Stellart : stereo hydro pneumo elektr sonst"
1300 LET T$(20) = "- Steuerungsart : Z-Lauf S-Kette R-Kreis"
1310 LET ZV$=STRING$(24,10)
1320 PRINT ZV$
1330 REM Szene 1: Vorstellung
1340 PRINT "Das Programm PATPRO1 ist ein Bestandteil"
1350 PRINT "des Programmpakets H E U R E K A : "
1360 PRINT "Hilfsmittel fuer das Erfinden unter "
1370 PRINT "- Rechnernutzung (Buero-, Heimcomputer)"
1380 PRINT "- Karteinutzung (Steil-, Sichtlochk.)."
1390 PRINT "PATPRO1 ermoeglicht:"
1400 PRINT "- Datenerfassung von Patenten"
1410 PRINT "- Datenspeicherung strukturierter Patentbeschreibungen"
1420 PRINT "- Datensuche nach Patentinhalten"
1430 PRINT "- Datenvergleich Patente - Ideen."
1440 PRINT "Das Musterbeispiel betrifft Wickeleinrichtungen"
1450 PRINT "fuer Seile und Ketten (s. B 66 D 1/36 ff.)."
1460 PRINT
1470 INPUT "weiter ? ", EG$
1480 IF EG$=" " OR EG$="j" OR EG$="J" THEN 1490 ELSE 1320
1490 PRINT ZV$
1500 PRINT "A - Aufbau der Datei"
1510 PRINT "N - Nutzung der Datei"
1520 PRINT "E - Ende"
1530 INPUT "Eingabe (A, N oder E): ", EG$
1540 IF EG$="a" OR EG$="A" THEN 2000
1550 IF EG$="n" OR EG$="N" THEN 3000
1560 IF EG$="e" OR EG$="E" THEN 4000 ELSE 1490
```

Bild 2.9.

Szene 2: Dateiaufbau

- Datei eröffnen (OPEN)
  - . "R" bedeutet wahlfreier Satzzugriff
  - . "1" bedeutet Datei 1
  - . "PATDAT1" ist der Dateiname
  - . "128" ist die maximale Satzlänge
- Bearbeiternamen anfordern (PRINT)
- Bildschirm löschen (PRINT)
- Schleife: Standardangaben anfordern (FOR I ... NEXT I)
- Bildschirm löschen (PRINT)
- Schleifen: Klassifizierungsangaben anfordern (FOR I ...)
- Zeichenkette aufbauen (LET ... LET bei 2180):  
jeder Eingabe wird ein Komma als Trennzeichen angefügt;  
die Eingabewerte sind sämtlich Zeichenketten (strings)
- Satznummer gewinnen (LET, VAL, vorsichtshalber INT)
- Satz auf Diskette ausgeben:
  - . Feld mit 128 Zeichen für K-Zeichenkette (FIELD)
  - . Zeichenkette links einordnen, Abschlußkomma (LSET)
  - . Ausgabe (PUT)
- Satz zur Kontrolle auf Drucker ausgeben (LPRINT)
- Weiterarbeit erfragen (INPUT, weiterer Dateiaufbau).

Szene 3: Dateinutzung

- Datei eröffnen (OPEN)
- Bildschirm löschen (PRINT)
- Objektnummer anfordern (INPUT)
- Satz von Diskette eingeben:
  - . Feld vorbereiten (FIELD)
  - . Eingabe (GET)
- Schleifen: Satz in Felder (von Komma zu Komma) zerlegen  
(FOR J ... NEXT J bei 3150)
- Schleifen: Klassifikationsvorschrift und Satz ausdrucken  
(FOR J ... NEXT J bei 3230)
- Weiterarbeit erfragen (INPUT, weitere Dateinutzung)

Szene E: Endebehandlung

- Programm und Datei schließen (END, enthält CLOSE),  
Bild 2.10.

```
2000 REM Szene 2: Dateiaufbau
2010 OPEN "R",#1,"PATDAT1",128
2020 PRINT S$(1);TAB(18) : INPUT "",M$(1)
2030 PRINT ZV$
2040 FOR I=2 TO 7
2050   PRINT S$(I);TAB(18) : INPUT "",M$(I)
2060 NEXT I
2070 PRINT ZV$
2080 PRINT S$(8)
2090 FOR I=1 TO 4 : PRINT T$(I) : NEXT I
2100 FOR I=5 TO 6 : PRINT T$(I);TAB(60) ; : INPUT "",N$(I) : NEXT I
2110 PRINT T7$
2120 FOR I=7 TO 13: PRINT T$(I);TAB(60) ; : INPUT "",N$(I) : NEXT I
2130 PRINT T14$
2140 FOR I=14 TO 18:PRINT T$(I);TAB(60) ; : INPUT "",N$(I) : NEXT I
2150 LET K1$=M$(1)
2160 FOR I=2 TO 7 : LET K1$=K1$+" "+M$(I): NEXT I
2170 LET K2$=N$(5)
2180 FOR I=6 TO 18: LET K2$=K2$+" "+N$(I): NEXT I
2190 LET KA$=K1$+" "+K2$
2200 LET OZ%=INT(VAL(M$(2)))
2210 FIELD #1,128 AS K$
2220 LSET K$=KA$+
2230 PUT #1,OZ%
2240 LPRINT K$
2250 INPUT "weiter (j oder n) ? ",EG$
2260 IF EG$="j" OR EG$="J" THEN 2030 ELSE 4000
3000 REM Szene 3: Dateinutzung
3010 OPEN "R",#1,"PATDAT1",128
3020 PRINT ZV$
3030 INPUT "Objektnummer: ",OZ%
3040 FIELD #1,128 AS K$
3050 GET #1,OZ%
3060 FOR J=1 TO 7
3070   I=INSTR(K$,",")
3080   M$(J)=LEFT$(K$,I-1)
3090   K$     =RIGHT$(K$,LEN(K$)-I)
3100 NEXT J
3110 FOR J=5 TO 18
3120   I=INSTR(K$,",")
3130   N$(J)=LEFT$(K$,I-1)
3140   K$     =RIGHT$(K$,LEN(K$)-I)
3150 NEXT J
3160 FOR J=1 TO 7 : LPRINT S$(J);M$(J) : NEXT J
3170 LPRINT S$(8)
3180 FOR J=1 TO 4 : LPRINT T$(J) : NEXT J
3190 FOR J=5 TO 6 : LPRINT T$(J);N$(J) : NEXT J
3200 LPRINT T7$
3210 FOR J=7 TO 13: LPRINT T$(J);N$(J) : NEXT J
3220 LPRINT T14$
3230 FOR J=14 TO 18:LPRINT T$(J);N$(J) : NEXT J
3240 INPUT "weiter (j oder n) ? ",EG$
3250 IF EG$="j" OR EG$="J" THEN 3020 ELSE 4000
4000 REM Szene E: Ende
4010 END
```

d) Programmmutzung

Nach dem Aufruf

LOAD "PATPRO1" RUN

stellt sich PATPRO1 vor.

Das Menüangebot enthält

- A - Aufbau der Datei
- N - Nutzung der Datei
- E - Ende (der PATPRO1-Nutzung).

Auf die Eingabe a oder A erscheinen die Texte für die Standardangaben (Bearbeitername bis Anmeldejahr), danach die Klassifikationsvorschrift mit dem Cursor am Ende der jeweiligen Zeile: das Patent kann nun klassifiziert werden, die Patentkurzbeschreibung wird auf Diskette gespeichert.

Auf die Eingabe n oder N wird nach der Objektnummer (Speichernummer) gefragt. Für sie werden dann die Standard- und Klassifizierungsangaben ausgegeben,

Bild 2.11.

Die inhaltlich orientierte Suche, der Vergleich von gespeicherten Patenten und der Vergleich zwischen Idee und Patenten sind einfach realisierbar, aber in PATPRO1 (aus Platzgründen) nicht dargestellt,

s. dazu Bild 2.8.

e) Programmverbesserungen

Das Programm PATPRO1 ist vielfältig verbesserbar, z.B.

- Programminhalt: Inhaltssuche, Patentvergleich
- Programmform : Komfort, z.B. Anbieten der nächsten Speichernummer beim Erfassen, Dateiendemeldung beim Nutzen, Korrekturmöglichkeiten beim Eingeben
- Programmumfang: (entsprechend Inhalt)
- Dateinhalt : Klassifikationsvorschriften
- Dateiform : mehrstufig
- Dateiumfang : mehr Merkmale, z.B. Patentgültigkeit, Patentinhaber usw.

Bearbeitername: Herrig

Objektnummer : 1

Patentklasse : B 66 D 1/36

Land-Patentart: DD-WP

Patentnummer : 129 894

Patenttitel : Wickeleinrichtung

Anmeldejahr : 1978

Patenthinhalt :

Schlüsselnummer: 0 - null, ohne

6 - kombiniert

7 - beliebig

Trommelaantrieb : ...1... ...2... ...3... ...4... ...5...

- Bewegungsricht: drehen schieb schraub taumeln pendeln 1

- Kraft auf Seil: Zug Druck Biegung Verdreh Scherung1

Seilfuehrung :

- Fuehrungsort : bei Tr. an Tr. vor Tr. hint.Tr. 2

- Freiheitszahl : 1 2 3 4 5 2

- Freiheitsart : drehen schieb schraub taumeln pendeln 5

- Schluszzahl : 1 2 3 4 5 2

- Schluszart : Kraft Form Stoff 2

- Schluszobjekt : Oeffng Rolle Rad Schlitten 2

- Lagerung : starr federnd bewegl 1

Wickelsteuerung : ...1... ...2... ...3... ...4... ...5...

- Meszort : bei Tr. an Tr. vor Tr. hint.Tr. 0

- Meszart : mechan elektr elektro optisch akust 0

- Uebertragung : mechan elektr elektro optisch akust 1

- Stellart : stereo hydro pneumo elektr sonst 1

- Steuerungsart : Z-Lauf S-Kette R-Kreis 1

Bild 2.11.

### 2.2.3. EFFPRO1 - ein Effektkettungsprogramm

#### a) Programmidee

EFFPRO1 unterstützt das Strukturkreieren von Objekten durch Kombinieren von Effekten:

Jede Struktur-, Natur- und Technikwissenschaft kennt  $10^2$  bis  $10^3$  Effekte, die sich technisch nutzen lassen.

Beispiele:

- Aus der Geometrie stammt das MÖBIUS-Band, ein einfach gewundenes endloses Band, das nur eine Oberfläche hat und deshalb zum Verdoppeln der wirksamen Länge von Riemen und Bändern (Schleifbänder!) verwendbar ist.
- Aus der Physik stammt der WEISSENBURG-Effekt, eine Art von Antizentrifugaleffekt, der entsteht, wenn in einem Gefäß mit nichtnewtonscher, also zäher Flüssigkeit ein Zylinder rotiert und dabei die Normalspannungen ein Hochkriechen der Flüssigkeit am bewegten Zylinder bewirken, so daß man die Drehzahl messen kann.

Alle Effekte lassen sich als Eingangs-Ausgangs-Wandler beschreiben und zu Ketten, Ringen, Bäumen und Netzen so koppeln, daß Eingangsgrößen  $E_1$  soll in Ausgangsgrößen  $A_n$  soll gewandelt werden können,

Bild 2.12.

Die Effekte sind anschaulich in Matrizen einordnenbar,

Bild 2.13.

Effektmatrizen sind in /11/, /12/, /13/ und /14/ dargestellt; andere Effektsammlungen sind in /15/, /16/ und /17/ enthalten.

EFFPRO1 sucht nun alle Effekte, die

- eine gegebene oder geforderte Eingangsgröße in eine geforderte Ausgangsgröße wandeln
- eine gegebene Eingangsgröße in eine beliebige Zwischengröße wandeln
- eine beliebige Zwischengröße in eine geforderte Ausgangsgröße wandeln.

Auf diese Weise kann der Erfinder mit EFFPRO1 beliebig lange Effektketten bilden.

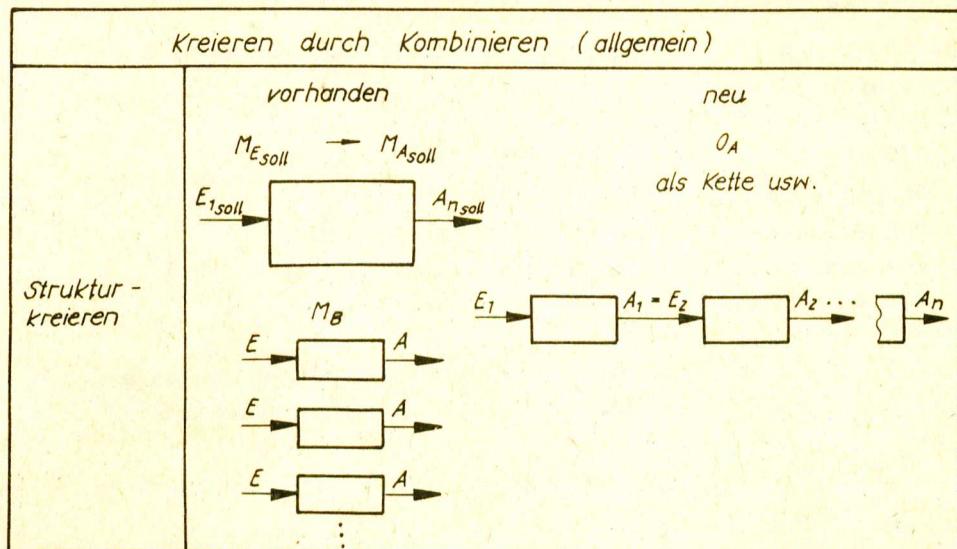


Bild 2.12.

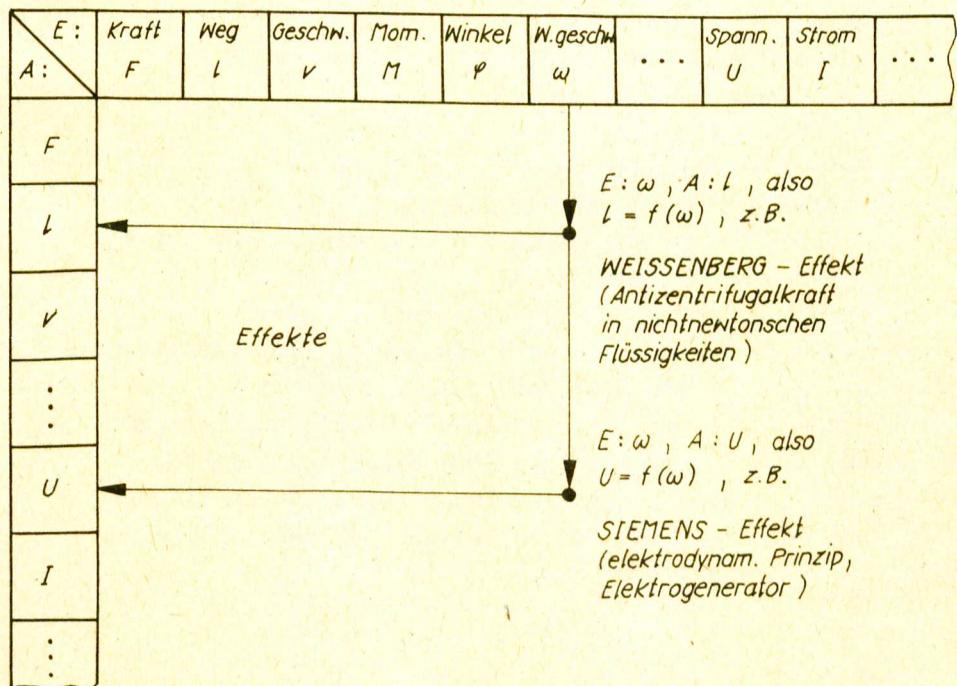


Bild 2.13.

b) Programmdaten

EFFDAT1 enthält die einfachste Beschreibung einiger physikalischer Effekte:

- Effektnummer (EFFNUM als Zahl (N))
- Effektname (EFFNAM als Zeichenkette (C))
- Eingangsgröße (EFFEIN als Zeichenkette)
- Formelzeichen (EFFZEE als Zeichenkette)
- Ausgangsgröße (EFFAUS als Zeichenkette)
- Formelzeichen (EFFZEA als Zeichenkette),

Bild 2.14 und Bild 2.15.

Diese normale Beschreibung - ein Effekt, einige Merkmale des Effekts - ist auf mindestens dreierlei Art verbesserrbar:

- das Merkmal "Mittel" wird mit erfaßt  
(beispielsweise beim WEISSENBERG-Effekt die nichtnewton-sche Flüssigkeit)
- das Merkmal "Anwendung" wird mit erfaßt  
(beispielsweise beim WEISSENBERG- Effekt  
"Drehzahlmessung",  
beispielsweise beim Zentrifugal- Effekt  
"Drehzahlmessung"  
"Krafterzeugung (Schleuderguß)"  
"Stofftrennung (Zentrifuge):
  - . Butterherstellung
  - . Metallkörnergewinnung (FR-, DE-Patente)
  - . Mädchenerzeugung (JP-Patente; Grundgedanke:  
X-Chromosomen-Spermienzellen sind etwas schwerer als  
X-Y-Spermienzellen /18/)
- die quantitativen Merkmale werden mit erfaßt.

Die inverse Beschreibung - eine Anwendung, alle zutreffenden Effekte - ist für den Erfinder besonders wichtig:

Für einen Aufgabentyp - beispielsweise "Drehzahlmessung" - findet er alle bisher als brauchbar erkannten Effekte.

Kurzgefaßte inverse Beschreibungen physikalischer Effekte sind in /19/, /17/ und /12/ enthalten und können für eine Datei INVDAT1 genutzt werden.

```
list struc
STRUCTURE FOR FILE: C:EFFDAT1 .DBF
NUMBER OF RECORDS: 00011
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD      NAME      TYPE WIDTH DEC
001      EFFNUM    N    003
002      EFFNAM    C    020
003      EFFEIN    C    012
004      EFFZEE    C    002
005      EFFAUS    C    012
006      EFFZEA    C    002
** TOTAL **           00052
```

Bild 2.14.

```
list
00001   1 Anisotropie-Effekt Kraft..... F Spannung.... U
00002   2 JOSEPHSON-Effekt Spannung;... U Winkelges... om
00003   3 Kondensator-Effekt Kraft..... F Spannung.... U
00004   4 Kondensator-Effekt Laenge..... l Spannung.... U
00005   5 MAGNUS-Effekt Winkelges... om Kraft..... F
00006   6 Piezoelektric-Effekt Kraft..... F Spannung.... U
00007   7 SIEMENS-Effekt Spannung.... U Winkelges... om
00008   8 SIEMENS-Effekt Winkelges... om Spannung.... U
00009   9 WEISSENBERG-Effekt Winkelges... om Laenge..... l
00010  10 Wirbelstrom-Effekt Winkelges... om Kraft..... F
00011  11 Zentrifugal-Effekt Winkelges... om Kraft..... F
```

Bild 2.15.

### c) Programmbeschreibung

EFFPRO1 ist ein einfaches Suchprogramm.

EFFPRO1 ist ein dBASE-Programm, s.a. Anlage 5.2.

EFFPRO1 besteht aus nur einer Szene:

- Systemnachrichten ausschalten (set ...)
- Hauptspeicherbereiche löschen (clear)
- Bildschirm löschen (erase)
- Datei eröffnen (use)
- Schleife: solange durchlaufen, bis die Antwort auf die Frage nach der Weiterarbeit verneint wird  
(do ... enddo am Programmende)
- Variable auf leer oder 0 stellen;  
für Eingabewerte zwei Stellen vorsehen:  
 $\omega$  = omega = om , Winkelgeschwindigkeit = Wi usw. (store)
- Aufgabe anfordern, Aufgabenwerte speichern  
(say ... get ... read)
- Ausgabe auf Drucker legen (set ...)
- Datei auf Anfang stellen (go top)
- Effekte, die die Eingangsgröße in die Ausgangsgröße wandeln,  
bis zum Dateiende suchen  
(locate ... do ... eof ... continue ... enddo)
- wenn keine Effekte gefunden werden, dann "keine" ausgeben  
(if ... endif)
- Datei auf Anfang stellen (go top)
- Effekte, die die Eingangsgröße in beliebige Zwischengrößen  
wandeln, bis zum Dateiende suchen und links als Druck ausgeben (locate ...)
- Datei auf Anfang stellen (go top)
- Effekte, die beliebige Zwischengrößen in die Ausgangsgröße  
wandeln, bis zum Dateiende suchen und rechts als Druck ausgeben (locate ...)
- Ausgabe auf Bildschirm legen (set ...)
- Weiterarbeit erfragen (say, get, read)
- Schleifenabschluß (enddo)
- Programm- und Dateiabschluß (return),

Bild 2.16.

```
* Programm EFFPRO1
set talk off
clear
use effdat1
store ' ' to antwort
do while antwort<>'N'
    store ' ' to antwort
    store ' ' to eingro
    rem usw. fuer einfor, ausgro, ausfor
    store 0 to z
    erase
    @ 1,0 say 'Effekte wandeln Eingangs- in Ausgangsgroeszen !'
    @ 3,0 say 'Die Eingangsgroesze heiszt...' get eingro
    @ 4,0 say 'Das Formelzeichen ist.....' get einfor
    @ 5,0 say 'Die Ausgangsgroesze heiszt...' get ausgro
    @ 6,0 say 'Das Formelzeichen ist.....' get ausfor
    read
    set format to print
    @ 8,0 say 'Fuer diese Aufgabe gibt es folgende Effekte:'
    @ 9,0 say '(Loesungen mit einem Effekt)'
    go top
    locate for (effein=eingro .and. effzee=einfor .and. ;
                effaus=ausgro .and. effzea=ausfor)
    do while .not. eof
        @ 10+z, 0 say effein
        @ 10+z,12 say effzee
        @ 10+z,20 say effnum
        @ 10+z,24 say effnam
        @ 10+z,50 say effaus
        @ 10+z,62 say effzea
        store z+1 to z
        continue
    enddo
    go top
    if z=0
        @ 10, 0 say 'keine'
    endif
    store z+1 to z
    @ 10+z, 0 say '(Loesungen mit mehreren Effekten)'
    store z+1 to z
    go top
    locate for (effein=eingro .and. effzee=einfor)
    do while .not. eof
        @ 10+z, 0 say effein
        @ 10+z,12 say effzee
        @ 10+z,18 say effnum
        @ 10+z,24 say effaus
        @ 10+z,36 say effzea
        store z+1 to z
        continue
    enddo
    go top
    locate for (effaus=ausgro .and. effzea=ausfor)
    do while .not. eof
        @ 10+z, 0 say 'und...'
        rem wie oben, aber mit Spalten 24, 36, 43, 50, 62
        store z+1 to z
        continue
    enddo
    set format to screen
    @ 15, 0 say 'weiter (j oder n) ? '
    get antwort picture '!'
    read
enddo
return
```

#### d) Programmmnutzung

Nach dem Aufruf

do effpro1

meldet sich EFFPRO1 mit dem Satz

"Effekte wandeln Eingangs- in Ausgangsgrößen !"

und erwartet diese Größen und ihre Formelzeichen als jeweils zweistellige Abkürzungen.

Die Ergebnisse der Suche werden ausgedruckt.

Beispiel: Drehzahlmessung

Fuer diese Aufgabe gibt es folgende Effekte:

(Loesungen mit einem Effekt)

Winkelges...om	2	JOSEPHSON-Effekt	Spannung....U
Winkelges...om	8	SIEMENS-Effekt	Spannung....U

(Loesungen mit mehreren Effekten)

Winkelges...om	2	Spannung....U	
Winkelges...om	5	Kraft.....F	
Winkelges...om	8	Spannung....U	
Winkelges...om	9	Laenge.....l	
Winkelges...om	10	Kraft.....F	
Winkelges...om	11	Kraft.....F	
und...		Kraft.....F	1 Spannung....U
und...		Winkelges...om	2 Spannung....U
und...		Kraft.....F	3 Spannung....U
und...		Laenge.....l	4 Spannung....U
und...		Kraft.....F	6 Spannung....U
und...		Winkelges...om	8 Spannung....U

s.a. Bild 2.17.

Weiterarbeit ist sinnvoll, wenn die Zwischengrößen als Eingangsgrößen bzw. als Ausgangsgrößen aufgefaßt werden. Auf diese Weise entstehen - von "vorn" und von "hinten" gleichzeitig gebildet - beliebig lange Effektketten.

#### e) Programmverbesserungen

Das Programm EFFPRO1 ist vielfältig verbesserbar, z.B.:

- Programminhalt: Verträglichkeitstest, auch quantitativ
- Programmform : Komfort, z.B. Menüangebot der Aufgaben
- Programmumfang: (entsprechend Inhalt)
- Dateiinhalt : Mittel, Anwendungen, Werte als Merkmale
- Dateiform : mehrstufig mit Beispielen, auch invers
- Dateiumfang : mehr Effekte.

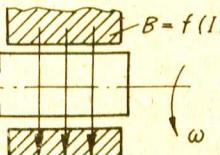
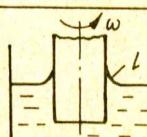
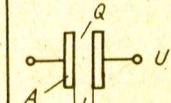
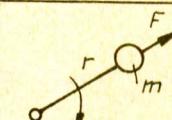
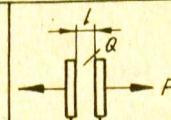
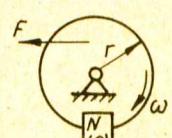
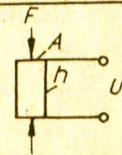
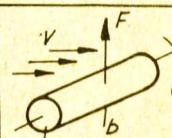
Aufgabe: Drehzahlen ( $E : \omega$ ) sind elektrisch ( $A : U$ ) zu messen.					
Lösungen mit einem Effekt:		$\omega \rightarrow U$			
Effekt	Gesetz	Erläuterung			
SIEMENS - Effekt (BIOT - SAVART - Effekt, el.-dyn. Prinzip)	$U = c \cdot f(I) \cdot \omega$				
JOSEPHSON - Effekt	$U = c \cdot I \cdot \omega$	bei supraleitenden Materialien			
Lösungen mit zwei Effekten:		$\omega \rightarrow I \rightarrow U$			
Effekt	Gesetz	Erläuterung	Effekt	Gesetz	Erläuterung
$\omega \rightarrow I$		$I \rightarrow U$			
WEISSEN - BERG - Effekt	$I = f(\omega)$		Konden - sator - effekt	$U = \frac{Q}{\epsilon_0 \cdot \epsilon_r \cdot A} \cdot L$ $\epsilon_r$ - Dielektr. - konst.	
$\omega \rightarrow F$		$F \rightarrow U$			
Zentri - fugal - effekt	$F = m \cdot r \cdot \omega^2$		Konden - sator - effekt	$U = \frac{L}{Q} \cdot F$	
Wirbel - strom - effekt	$F = c \cdot B^2 \cdot r \cdot \omega$		Piezo - elektr. Effekt	$U = \frac{c \cdot h}{A} \cdot F$	
MAGNUS - Effekt (FLETTNER - Rotor)	$F = 2\pi r^2 \cdot g \cdot v \cdot b \cdot \omega$		Aniso - tropie - Effekt	$U = f(F)$	z.B. als Druck - oder Zug - effekt

Bild 2.17.

#### 2.2.4. WIDPRO1 - ein Widerspruchsmerkmalprogramm

##### a) Programmidee

WIDPRO1 unterstützt das Kreieren von Merkmalen durch Bereitstellen von Regeln und Vorschriften zum Aufheben von Widersprüchen:

Es gibt fast 40 000 000 Patentschriften, aber vermutlich nur 40 bis 100 Regeln und Vorschriften zur erforderlichen Abwandlung und Verwandlung von Objekten und ihren Umgebungen.

ALTSCHULLER (Al'tšuller) hat 40 000 Patente untersucht und dabei

- etwa 40 Merkmale technischer Objekte gefunden, die - vorwiegend paarweise - miteinander im Widerspruch stehen können
- bisher 40 Regeln, Vorschriften oder Verhaltens-Prinzipien zum Beheben oder Aufheben der Widersprüche

gefunden /7/, /19/.

Es gelang ihm sogar, jedem Widerspruchspaar bis zu 4 Regeln zuzuordnen.

Beispiel:

Ist die bewegliche Masse (1) eines Objekts zu ändern, wegen der Materialökonomie vorzugsweise zu verringern, so steht dem die Leistung oder Leistungsfähigkeit (21) des Objekts entgegen, die im allgemeinen zu vergrößern ist.

In diesem Falle schlägt ALTSCHULLER vor:

- 12 Nutze Äquipotentialniveau !
- 18 Nutze Schwingungen (und/oder Reibungsminderer) !
- 36 Nutze Phasenübergänge !
- 31 Nutze Porosität !

WIDPRO1 sucht nun für alle möglichen Widerspruchspaare diese Regeln,

Bild 2.18.

Bemerkenswert ist, daß die Relation Merkmal - Widerspruchsmerkmal nicht symmetrisch ist,

s. Bild 2.18.

Merkmal, das entgegensteht	1 Masse, beweglich	2 Masse, unbeweglich	... 	21 Leistung, Kapazität	22 Energie - verluste
Merkmal, das zu ändern ist					
1 Masse, beweglich					12: Äquipotentialniveau 18: Schwingungen 36: Phasenübergänge 31: Porosität
2 Masse, unbeweglich				Regeln, Vorschriften	
:					
21 Leistung, Kapazität				8: Gegen-, Zusatzkräfte 36: Phasenübergänge 38: Oxydationsmittel 31: Porosität	
22 Energie - verluste					
:					

Bild 2.18.

b) Programmdateien

WIDPRO1 nutzt zwei Dateien:

- WIDDAT1 mit Sätzen aus
  - . Widerspruchsnummer (WIDNUM als Zahl, vierstellig:
    - .. zwei Stellen für das Merkmal
    - .. zwei Stellen für das Widerspruchsmerkmal)
  - . Lösungsnummern 1 ... 4 (LOENUM1...4 als Zahlen):

PRIMARY USE DATABASE					
FLD	NAME	TYPE	WIDTH	DEC	
001	WIDNUM	N	004		
002	LOENUM1	N	002		
003	LOENUM2	N	002		
004	LOENUM3	N	002		
005	LOENUM4	N	002		
** TOTAL **			00013		

list					
00001	102	0	0	0	0
00002	121	12	18	36	31
00003	122	6	2	34	19
00004	201	0	0	0	0
00005	221	15	19	18	22
00006	222	18	19	28	15
00007	2101	8	36	38	31
00008	2102	19	26	17	27
00009	2122	10	35	38	0
00010	2201	15	6	19	28
00011	2202	19	6	18	9
00012	2221	3	38	0	0

(Auszug)

- LOEDAT1 mit Sätzen aus
  - . Lösungsnummer (LOENUM als Zahl)
  - . Lösungstext (LOETEX als Zeichenkette),

Bild 2.19. und Bild 2.20.

Eine dritte Datei mit Beispielen

- aus dem Maschinenbau
  - aus der Elektrotechnik
  - aus der Verfahrenstechnik /20/
- ist in Vorbereitung (BEIDAT1).

```
list struc
STRUCTURE FOR FILE: C:LOEDAT1 .DBF
NUMBER OF RECORDS: 00040
DATE OF LAST UPDATE: 00/00/00
SECONDARY USE DATABASE
FLD.      NAME      TYPE WIDTH DEC
001.    LOENUM     N    002
002.    LOETEX     C    060
** TOTAL **           00063
```

Bild 2.19.

```
list
00001  1 Zerlege das Objekt in Teilobjekte!
00002  2 Trenne den stoerenden Teil ab!
00003  3 Passe die Struktur oertlich/zeitlich an!
00004  4 Fuehre das Objekt asymmetrisch aus!
00005  5 Kopple Objekte raeumlich, Operationen zeitlich!
00006  6 Gib dem Objekt mehrere Funktionen (Funktionsintegration)!
00007  7 Verschachtele mehrere Objekte!
00008  8 Nutze Gegen- oder Zusatzkraefte (mech./hydr./pneu./magn.)!
00009  9 Erzeuge vorherige Gegenwirkung (z.B. Vorspannung)!
00010 10 Erzeuge vorherige Wirkung!
00011 11 Beuge dem Schaden vor!
00012 12 Erzeuge Aequipotentialniveau (z.B. Gleichgewicht)!
00013 13 Kehre die Funktion um!
00014 14 Nutze krummlinige Formen (Kreis, Spirale, Kugel usw.)!
00015 15 Gestalte das Objekt dynamisch (anpaszbar, steuerbar)!
00016 16 Erzeuge kleinere oder groeszere Wirkung (z.B. Ueberschusz)!
00017 17 Nutze hoehere Dimensionen (z.B. Flaeche, Raum)!
00018 18 Nutze Schwingungen! Nutze Reibungsminderer!
00019 19 Nutze Impulse! Nutze Impulspausen!
00020 20 Nutze Kontinuitaet! Vermeide Leerlaeufe!
00021 21 Durcheile Gefaehrdungen!
00022 22 Wandle Schaedliches in Nuetzliches um!
00023 23 Fuehre Rueckkopplung ein!
00024 24 Nutze (zeitweilig) Adapter oder Katalysatoren!
00025 25 Nutze Selbsterzeugung! Erzeuge Selbsterzeugung!
00026 26 Nutze Objektkopien/Objektmodelle!
00027 27 Gestalte Objekt billig und kurzlebig!
00028 28 Ersetze Mechanik durch Optik, Akustik, Elektronik usw.!
00029 29 Nutze Hydro- oder Pneumoobjekte (z.B. Luftkissen)!
00030 30 Nutze biegsame Huellen und duenne Folien!
00031 31 Verwende poroese Stoffe! Fuelle (zeitweilig) Poren!
00032 32 Veraendere die Farbe (Farbstoffe, Leuchtstoffe)!
00033 33 Verwende gleiche Teile oder gleiche Stoffe!
00034 34 Beseitige oder regeneriere verbrauchte Teile!
00035 35 Aendere Aggregatzustand! Nutze Elaste!
00036 36 Nutze Phasenuebergaenge (z.B. Curie-Punkt)!
00037 37 Nutze Waermedehnung!
00038 38 Verwende starke Oxydationsmittel!
00039 39 Nutze Vakuum oder inerte Stoffe!
00040 40 Nutze zusammengesetzte Stoffe (Verbund, Gemisch)!
```

Bild 2.20.

c) Programmbeschreibung

WIDPRO1 ist ein Suchprogramm mit zwei Dateien.

WIDPRO1 ist ein dBASE-Programm, s.a. Anlage 5.2.

WIDPRO1 besteht aus nur einer Szene:

- Systemnachrichten ausschalten (set ...)
- Bildschirm löschen (erase)
- erste Datei als Primärdatei auswählen (select)
- zweite Datei als Sekundärdatei bezeichnen (select)
- Schleife: solange durchlaufen, bis die Antwort auf die Frage nach der Weiterarbeit verneint wird  
(do ... enddo am Programmende)
- (Widerspruch-) Merkmale anzeigen (say)
- Änderungsmerkmal erfragen (say, Zeile 13)
- Widerspruchsmerkmal erfragen (say, Zeile 14)
- Suchfrage aus Eingabewerten bilden (store)  
(Summe ist vierstellige Widerspruchsnummer)
- Primärdatei ansprechen (select)
- Datei auf Anfang stellen (go top)
- Satz mit Widerspruchsnummer suchen (locate)
- Bildschirm teilweise löschen (store, do ... enddo)
- Lösungsbüberschrift ausgeben (say)
- Sekundärdatei ansprechen (select)
- Datei auf Anfang stellen (go top)
- für vorhandene erste Lösungsnummer den Lösungstext ausgeben,  
sonst auf Programm ASSPRO hinweisen (if ... endif)
- für vorhandene weitere Lösungsnummern den Lösungstext aus-  
geben (if ... endif usw.)  
(aus Übersichtlichkeits- und Platzgründen nicht anders pro-  
grammiert)
- Weiterarbeit erfragen (say)
- Bildschirm löschen (erase)
- Schleifenabschluß (enddo)
- Programm- und Dateienabschluß (return),  
Bild 2.21.

```
type widpro1.cmd
* Programm WIDPRO1
set talk off
erase
select primary
use widdat1
select secondary
use loedat1
store '' to antwort
set format to print
do while antwort<>'N'
    store '' to antwort
    @ 1, 0 say 'Die wichtigsten Merkmale technischer Objekte ;
    sind'
    @ 2, 0 say '- Masse, beweglich.....(01)'
    @ 2,32 say '- Leistung, Kapazitaet.(21)'
    @ 3, 0 say '- Masse, unbeweglich...(02)'
    @ 3,32 say '- Energieverluste.....(22)'
* usw.
    @ 12, 0 say -----
    store 0 to widnum1
    store 0 to widnum2
    @ 13, 0 say 'Welches Merkmal soll verbessert werden ?';
    get widnum1 picture '99'
    @ 14, 0 say 'Welches Merkmal steht dem entgegen ?      ';
    get widnum2 picture '99'
    read
    store 100*widnum1+widnum2 to widnums
    select primary
    go top
    locate for widnum=widnums
    store -1 to zeile
    do while zeile<12
        store zeile+1 to zeile
        @ zeile,0
    enddo
    @ 1, 0 say 'Versuchen Sie, das Problem so zu loesen:'
    select secondary
    go top
    if loenum1<>0
        locate for loenum=loenum1
        @ 3, 0 say loenum
        @ 3, 4 say loetex
    else
        @ 3, 0 say 'Nutzen Sie das Programm ASSPRO !'
    endif
    if loenum2<>0
        locate for loenum=loenum2
        @ 4, 0 say 'oder'
        @ 5, 0 say loenum
        @ 5, 4 say loetex
    endif
    * usw. oder als Schleife
    @ 15, 0 say 'Weitere Probleme ? (J/N) ';
    get antwort picture '!!'
    read
    erase
enddo
return
```

Bild 2.21.

d) Programmnutzung

Nach dem Aufruf

do widpro1

meldet sich WIDPRO1 mit dem Satzstück

"Wichtige Merkmale technischer Objekte sind:" ,

liefert das Menüangebot der Widerspruchsmerkmale (hier auszugsweise), erwartet die Eingabe und sucht die zutreffenden Regeln und Vorschriften.

Beispiel: Schiffswippkran

Wichtige Merkmale technischer Objekte sind:

- |                             |                             |
|-----------------------------|-----------------------------|
| - Masse, beweglich....(01)  | - Leistung, Kapazitaet.(21) |
| - Masse, unbeweglich...(02) | - Energieverluste.....(22)  |

---

Welches Merkmal soll verbessert werden ? 01

Welches Merkmal steht dem entgegen ? 21

Für die Verbesserung eines Schiffswippkranes,

- dessen Auslegermasse (01!) klein werden soll
- dessen Leistungsfähigkeit (21!) aber dann sinkt, weil der Ausleger bei Rückwärtsneigung des Schiffes und Vorwärtswind auf den Kran kinematisch instabil wird, also nach hinten kippen kann,

sind die Programmergebnisse anregend:

Äquipotentialniveau, hier also gerader Lastweg,

- darf nicht - wie bei Landkranen - horizontaler Lastweg bedeuten
- muß vielmehr - eben wegen der Schiffsneigung - ansteigender Lastweg bedeuten.

Das wird mit einer Zusatzrolle (16) im Hubseilsystem erreicht: sie wird bei großer Ausladung nicht oder kaum und bei mittlerer Ausladung seilverkürzend und seilhebelverkürzend umschlungen, s. /21/,

Bild 2.22.

e) Programmverbesserungen

Das Programm WIDPRO1 ist vielfältig besserbar, z.B.:

- Programminhalt: strukturierte Regeln, s. /20/
- Dateinhalt : Beispiele für Anwendungen.

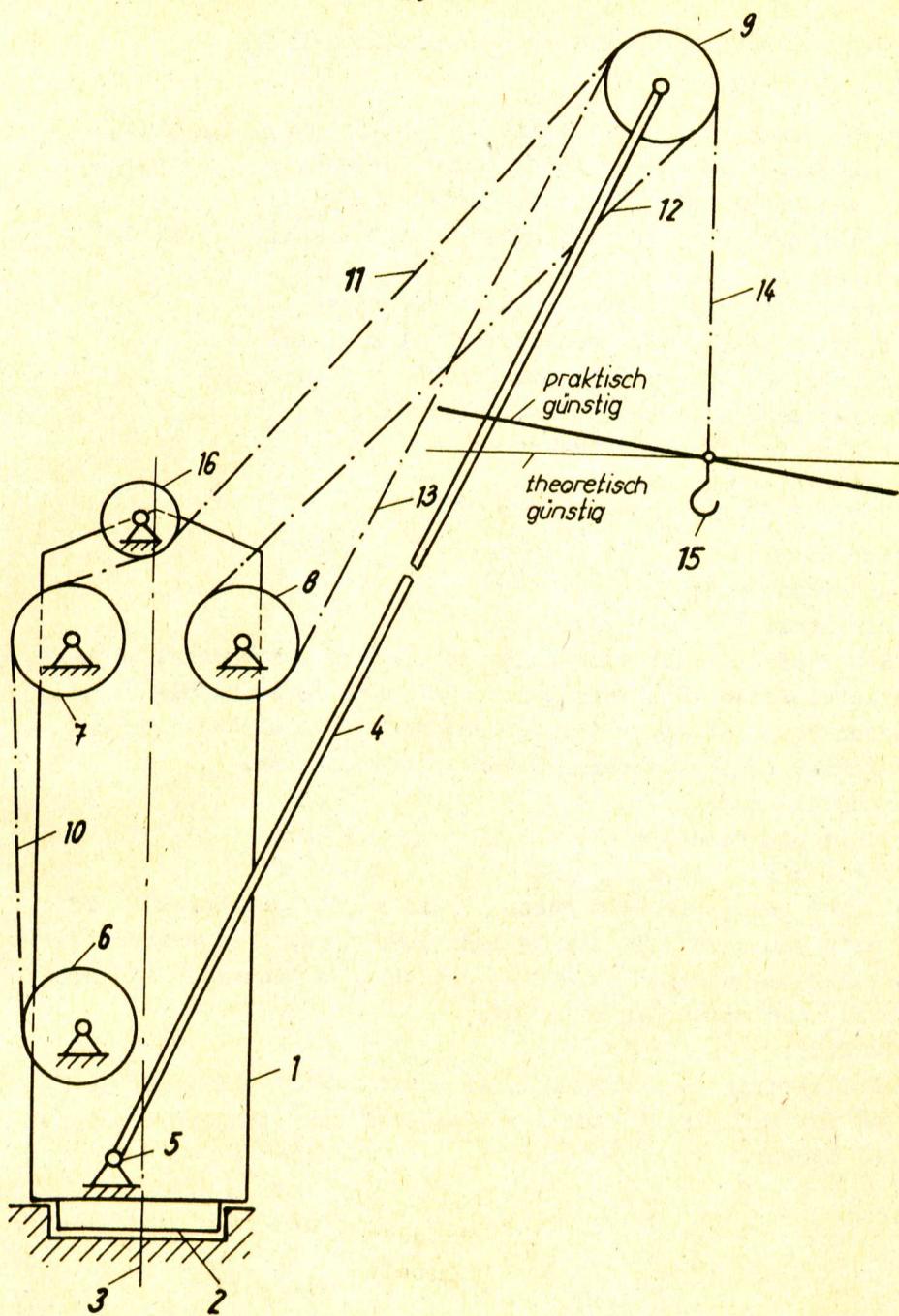


Bild 2.22.

### 2.2.5. STAPR01 - ein Standardsituationsprogramm

#### a) Programmidee

STAPR01 unterstützt das Variieren und Kreieren von Objekten durch Bereitstellen von Regeln und Vorschriften zum Umformen und zum Entwickeln von Stoff-Feld-Systemen:

Es gibt nur zwei abstrakte Bestandteile der Materie, nämlich

- Stoff (mit dem Eigenschaftsmerkmal Masse)
- Feld (mit dem Eigenschaftsmerkmal Energie).

ALTSCHULLER abstrahierte nun Erfindungssituationen bis zu Stoff-Feld-Systemen und fand dabei

- bisher 18 Standardsituationen
- etwa 30 Standardlösungen,

s. /19/.

Beispiel:

Standardsituation:

- ein Stoff S1 liegt vor
- der Stoff S1 lässt sich schlecht steuern (verändern, messen); beispielsweise liegt ein landtechnisches Versuchsgelände vor, dessen Boden S1 nicht für verschiedene Versuchsbedingungen (leichter Boden, schwerer Boden) steuerbar ist.

Standardlösung:

- führe ein Feld F ein
  - führe einen Stoff S2 ein;
- beispielsweise wird dem Boden S1 ein magnetisierbarer Stoff S2 untergemischt, und das Gemisch ändert seine Eigenschaften im elektromagnetischen Feld F, das von den Landmaschinen aus schalt- und steuerbar ist /19/,

Bild 2.23.

STAPR01 erfragt nun vom Erfinder schrittweise die Stoff-Feld-Situation und stellt dann die Standardlösung mit einem Beispiel bereit.

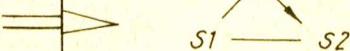
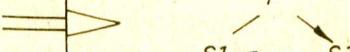
Nr.	Standardsituation : WENN ...	... DANN: Standardlösung	Nr.
7	ein Stoff $s_1$ vorliegt UND $s_1$ sich schlecht steuern lässt	füre ein Feld $F$ ein UND füre einen Stoff $s_2$ ein	1.1
	$s_1$		
11	ein Stoff $s_1$ vorliegt UND ein Feld $F$ vorliegt UND die Wechselwirkung auch Nachteile hat	füre einen Stoff $s_2$ ein UND Laß $F$ auf $s_2$ wirken UND verknüpfe $s_1$ mit $s_2$	2.8
	$F \approx s_1$		
z.Z. bis 18	...	...	

Bild 2.23.

b) Programmdaten

STAPRO1 enthält die Daten zur Beschreibung der Standardsituation; erfragt werden

- Anzahl der Elemente des Stoff-Feld-Systems
- Wenn nur ein Element vorliegt, wird weiter erfragt
  - . die Art des Elements (Stoff, Feld, Element beliebig)
  - . die Art künftiger Eigenschaften (steuerbar, verbessert), und die Standardlösungen 1.1. bis 1.3. (1 ... 3 in /19/) werden ausgegeben.
- Wenn zwei Elemente vorliegen, dann wird erfragt
  - . die Art der Elemente (Stoff/Stoff, Stoff/Feld)
  - . die Art der Wechselwirkung (keine, schädliche, ...)
  - . die Art künftiger Eigenschaften (wechselwirkend, steuerbar), und die Standardlösungen 2.1. bis 2.9. (4 ... 12 in /19/) werden ausgegeben.
- Wenn drei Elemente vorliegen, dann wird erfragt
  - . die Art der Aufgabe (Prüf-, Meß-, Wirkungsaufgabe)
  - . die Art der Veränderbarkeit (Ersatz möglich, unmöglich)
  - . die Art der Wechselwirkung (keine, schlechte, schädliche), und die Standardlösungen 3.1. bis 3.6. (14 ... 18 in /19/) werden ausgegeben.

STADAT1 enthält die Daten zur Beschreibung der Standardlösungen:

- Standardnummer (STANR als Zahl)
- Standardtext (STATEX1...7 als Zeichenketten)
  - . STATEX1...3 Allgemeinlösung
  - . STATEX4 Beispielsituation
  - . STATEX5...7 Beispieldlösung,

Bild 2.24. und Bild 2.25.

Die Standardnummer wird so gebildet:

- 1. Stelle aus der Anzahl der Elemente (1, 2, 3)
- 2. Stelle aus der Artbestimmung (1 ... 3, 1 ... 9, 1 ... 6).

```
list struc
STRUCTURE FOR FILE: C:STADAT1 .DBF
NUMBER OF RECORDS: 00002
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD      NAME      TYPE WIDTH DEC
001      STANR      N    002
002      STATEX1    C    060
003      STATEX2    C    060
004      STATEX3    C    060
005      STATEX4    C    060
006      STATEX5    C    060
007      STATEX6    C    060
008      STATEX7    C    060
** TOTAL **           00423
```

Bild 2.24.

- 00001 11 - Fuehre einen zweiten Stoff S2 ein  
- fuehre ein Feld F ein  
- lass das Feld auf S2 wirken.  
((Beispiel folgt))
- 00002 28 - Fuehre einen zweiten Stoff S2 ein  
- lass das Feld F auf S2 wirken  
- verknuepfe Stoff S1 mit S2.  
Spannbeton mit waermeempfindlichem Draht S1:  
- Fuehre einen waermeunempfindlichen Stab S2 ein  
- erwaerme mit F den Stab S2  
- verbinde S1 mit S2: S1 dehnt sich, wenn S2 schrumpft!

Bild 2.25. (Auszug)

c) Programmbeschreibung

STAPRO1 ist ein einfaches Suchprogramm für einen Fragebaum mit zwei (oder drei) Ebenen.

STAPRO1 ist ein dBASE-Programm, s.a. Anlage 5.2.

STAPRO1 besteht aus nur einer Szene:

- Systemnachrichten ausschalten (set ...)
- Bildschirm löschen (erase)
- Datei eröffnen (use)
- Schleife: solange durchlaufen, bis die Antwort auf die Frage nach der Weiterarbeit verneint wird  
(do ... enddo am Programmende)
- Menüangebot für Anzahl der Elemente (say)
- Variable auf 0 stellen (store)
- Menüauswahl (say, get, read)
- Fallunterscheidung (do case ... endcase)
  - . Fall 1: ein Element  
(nächstes Menüangebot, hier nicht dargestellt)
  - . Fall 2: zwei Elemente  
(nächstes Menüangebot, hier auszugsweise geliefert)
  - . Fall 3: drei Elemente  
(nächstes Menüangebot, hier nicht dargestellt)
- Variable auf 0 stellen (store)
- Menüauswahl (say, get, read)
- Suchfrage aus beiden Menüantworten bilden  
(store, zweistellige Standardnummer)
- Datei auf Anfang stellen (go top)
- Standardlösung suchen (locate)
- Bildschirm löschen (erase)
- Standardlösung ausgeben (say)
- Weiterarbeit erfragen (say, get, read)
- Bildschirm löschen (erase)
- Schleifenabschluß (enddo)
- Programm- und Dateiabschluß (return),  
Bild 2.26.

```
type stapro1.cmd

* Programm STAPRO1
set talk off
erase
use stadat1
store '' to antwort
do while antwort<>'N'
    store '' to antwort
    @ 1, 0 say 'Analysieren Sie Ihr Stoff-Feld-System !'
    @ 2, 0 say 'Ein Stoff-Feld-System hat zunaechst'
    @ 3, 0 say '- ein Element (Stoff, Feld).....(1)'
    @ 4, 0 say '- zwei Elemente (Stoff, Feld).....(2)'
    @ 5, 0 say '- drei Elemente (Stoff, Feld).....(3)'
    store 0 to stanr1
    @ 6, 0 say 'Wieviele Elemente hat das Stoff-Feld-System ?';
        get stanr1 picture '9'

    read
    * usw.
    do case
        case stanr1=1
        * usw.
        case stanr1=2
            @ 8, 0 say 'Die beiden Elemente koennen sein:'
            @ 9, 0 say '- zwei Stoffe,' 
            @ 10, 0 say ' die wechselwirken sollen und '
            @ 11, 0 say ' von denen einer veraenderbar ist...(1)'
        * usw.
            @ 12, 0 say '-- ein Stoff und ein Feld,' 
            @ 13, 0 say ' die bereits wechselwirken, aber '
            @ 14, 0 say ' die teilweise schaedlich wirken....(8)'
        * usw.
        case stanr1=3
        * usw.
    endcase
    store 0 to stanr2
    @ 16, 0 say 'Welche Wechselwirkungen liegen vor ? ';
        get stanr2 picture '9'

    read
    store 10*stanr1+stanr2 to stanrs
    go top
    locate for stanr=stanrs
    erase
    @ 1, 0 say 'Versuchen Sie, das Problem so zu loesen:'
    @ 3, 0 say statex1
    @ 4, 0 say statex2
    @ 5, 0 say statex3
    @ 6, 0 say 'Beispiel:'
    @ 7, 0 say statex4
    @ 8, 0 say statex5
    @ 9, 0 say statex6
    @ 10, 0 say statex7
    @ 15, 0 say 'Weitere Probleme ? (J/N) ';
        get antwort picture '!!'

    read
    erase
enddo
return
```

C>

Bild 2.26.

d) Programmnutzung

Nach dem Aufruf

do stapro1

meldet sich STAPRO1 mit dem Satz

"Analysieren Sie Ihr Stoff-Feld-System !"

und erwartet

- zunächst die Anzahl der Elemente
- danach die Art ihres gegebenen und geforderten Wirkens.

Beispiel: zwei Elemente (2), schädliche Wechselwirkung (8)

Analysieren Sie Ihr Stoff-Feld-System !

Ein Stoff-Feld-System hat zunaechst

- ein Element (Stoff, Feld).....(1)
- zwei Elemente (Stoff, Feld).....(2)
- drei Elemente (Stoff, Feld).....(3)

Wieviele Elemente hat das Stoff-Feld-System ? 2

Die beiden Elemente koennen sein:

- zwei Stoffe,  
die wechselwirken sollen und  
von denen einer veraenderbar ist...(1)
- ein Stoff und ein Feld,  
die bereits wechselwirken, aber  
die teilweise schaedlich wirken....(8)

Welche Wechselwirkungen liegen vor ? 8

Für diese Standardsituation wird dann die Standardlösung  
ausgegeben,

Bild 2.27 und Bild 2.28 nach SU 120 909, s.a. /19/.

e) Programmverbesserungen

Das Programm STAPRO1 ist vielfältig verbesserbar, z.B.:

- Programminhalt: Vervollständigung, größere Modellierungshilfe für den Erfinder
- Programmform : Komfort, Kontrollen, Korrekturmöglichkeit
- Programmumfang: (entsprechend Inhalt)
- Dateiinhalt : Beispiele nach Fachgebieten, Patentbeispiele
- Dateiform : mehrstufig
- Dateiumfang : (entsprechend Inhalt und Form).

Versuchen Sie, das Problem so zu loesen:

- Fuehre einen zweiten Stoff S2 ein
- lass das Feld F auf S2 wirken
- verknuepfe Stoff S1 mit S2.

Beispiel:

Spannbeton mit waermeempfindlichem Draht S1:

- Fuehre einen waermeunempfindlichen Stab S2 ein
- erwaerme mit F den Stab S2
- verbinde S1 mit S2: S1 dehnt sich, wenn S2 schrumpft !

Weitere Probleme ? (J/N)

Bild 2.27.

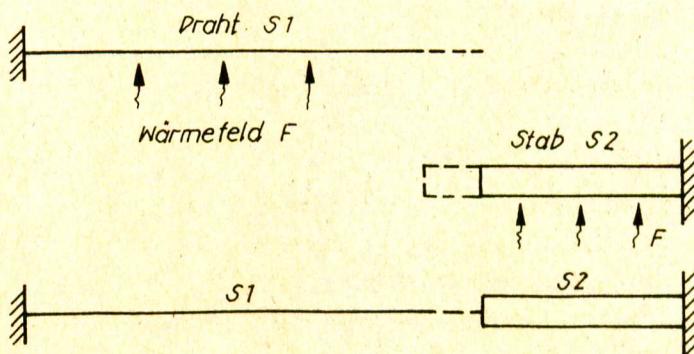


Bild 2.28.

### 2.2.6. TREPRO1 - ein Trenddateiprogramm

#### a) Programmidee

TREPRO1 unterstützt das Variieren und Kreieren von Objekten durch Bereitstellen von Regeln und Vorschriften auf der Grundlage technischer Trends:

Es gibt ungefähr 10 bis 20 technische Trends, die für die Entwicklung technischer Objekte kennzeichnend sind.

ALTSCHÜLLER /19/ und KAHMANN (unveröffentlicht) fanden Trends

- bei der Analyse von Patenten (Längsschnitte über Jahrzehnte)
- bei der Anwendung gesellschaftswissenschaftlicher Erkenntnisse
- beim Studium der Biografien bedeutender Wissenschaftler, Techniker, Politiker und Künstler.

Beispiel:

Technischer Trend: Dimensionalität

- Regel: Nutze alle Dimensionen !

- Merkmale:

- . Übergang zu Fläche und Raum
- . Übergang zu Kugel, Spirale, Schraube, Pseudosphäre
- . Übergang zu Verschachtelung und Vernetzung

- Musterbeispiele:

- . Mehrebenen-Leiterplatten, dreidimensionale Chips
- . Nahrung aus dem Meer
- . Hyper-Schalen-Bauweise
- . Schachtelbauten.

TREPRO1 stellt diese Trends mit Merkmalen, Beispielen und Patenten bereit.

Die inhaltliche Kopplung der Trends wird vorbereitet.

#### b) Programmdaten

TREPRO1 enthält ein Menüangebot der Trends.

TREDAT1 enthält die Daten zur Beschreibung der Trends, Bild 2.29 und 2.30.

```
list struc
STRUCTURE FOR FILE: C:TREDDAT1 .DBF
NUMBER OF RECORDS: 00001
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD      NAME      TYPE WIDTH DEC
001      TRENUM     N    002
002      TRENAM     C    060
003      TREMER1    C    060
004      TREMER2    C    060
005      TREMER3    C    060
006      TREBEI1    C    060
007      TREBEI2    C    060
008      TREBEI3    C    060
009      TREPAT1    C    060
010      TREPAT2    C    060
011      TREPAT3    C    060
** TOTAL **          00603
```

Bild 2.29.

Technischer Trend: Dimensionalitaet

Der Trend hat folgende Merkmale:

- Uebergang von Punkt/Linie zu Flaeche/Raum
- Uebergang zu doppelt gekruemmten Formen
- Uebergang zu Verschachtelungen/Vernetzungen

Den Trend zeigt folgendes Beispiel:

- bis 1950 Venturi-Duese (eindimensional)
- bis 1975 Venturi-Teller (zweidimensional)
- seit 1975 Venturi-Schraube (dreidimensional)

Den Trend zeigen die Patente:

- SU 287 806 / SU 486 768 (Teller)
- US 3 599 398 (Schraube)
- SU 502 645 (Verschachtelung)

Bild 2.30. (Auszug)

c) Programmbeschreibung

TREPRO1 ist ein einfaches Suchprogramm.

TREPRO1 ist ein dBASE-Programm, s.a. Anlage 5.2.

TREPRO1 besteht aus nur einer Szene:

- Systemnachrichten ausschalten (set ...)
  - Bildschirm löschen (erase)
  - Datei öffnen (use)
  - Schleife: solange durchlaufen, bis die Antwort auf die Frage nach der Weiterarbeit verneint wird  
(do ... enddo am Programmende)
  - Menüangebot für Trends (say)
  - Menüauswahl (say, get, read)  
(picture '99' heißt: numerisch, zweistellig)
  - Bildschirm löschen (erase)
  - Datei auf Anfang stellen (go top)
  - Trendbeschreibung suchen (locate)
  - Trendbeschreibung auf Bildschirm geben:
    - . Trendname
    - . Trendmerkmale 1 ... 3
    - . Trendbeispiele 1 ... 3
    - . Trendpatente 1 ... 3
  - Weiterarbeit erfragen (say, get, read)  
(picture '!' heißt: Kleinbuchstaben werden in Großbuchstaben gewandelt)
  - Bildschirm löschen (erase)
  - Schleifenabschluß (enddo)
  - Programm- und Dateiabschluß (return),  
Bild 2.31.
- TREPRO2 - in der Entwicklung - liefert Trendbeispiele für auswählbare Industriezweige, Kombinate, Erzeugnisgruppen bzw. Erzeugnisse. Es wird versucht, Trendfunktionen zu finden und zu nutzen.

```
* Programm TREPR01
set talk off
erase
use tredat1
store '' to antwort
do while antwort<>'N'
    store '' to antwort
    store 0 to trenum1
    @ 1, 0 say 'In der Technik herrschen folgende Trends:'
    @ 3, 0 say 'Raum: Dimensionen nutzen! (1)'
    @ 4, 0 say 'Zeit: Rhythmik abstimmen! (2)'
    @ 5, 0 say 'Steuerbarkeit erhöhen! (3)'
    @ 6, 0 say 'Stoff: Masse minimieren! (4)'
    @ 7, 0 say 'Werkstoffe konstruieren! (5)'
    @ 8, 0 say 'Feld: Energieaufwand minimieren! (6)'
    @ 9, 0 say 'Mikroeffekte nutzen! (7)'
    @ 10, 0 say 'Informat.: Anteil erhöhen! (8)'
    @ 11, 0 say 'berührungslos gewinnen! (9)'
    @ 12, 0 say 'System: Hierarchie nutzen! (10)'
    @ 13, 0 say 'Funktionen integrieren! (11)'
    @ 14, 0 say 'Struktur optimieren! (12)'
    @ 15, 0 say 'Zu welchem Trend wünschen Sie Einzelheiten ?';
        get trenum1 picture '99'
read
erase
go top
locate for trenum=trenum1
@ 1, 0 say 'Technischer Trend:'
@ 1, 20 say trenam
@ 2, 0 say '-----',
@ 3, 0 say 'Der Trend hat folgende Merkmale:'
@ 4, 0 say tremer1
@ 5, 0 say tremer2
@ 6, 0 say tremer3
@ 7, 0 say 'Den Trend zeigt folgendes Beispiel:'
@ 8, 0 say trebei1
@ 9, 0 say trebei2
@ 10, 0 say trebei3
@ 11, 0 say 'Den Trend zeigen die Patente:'
@ 12, 0 say trepat1
@ 13, 0 say trepat2
@ 14, 0 say trepat3
@ 15, 0 say 'weiter (j oder n) ? ';
        get antwort picture '!!'
read
erase
enddo
return
```

C>

Bild 2.31.

#### d) Programmnutzung

Nach dem Aufruf

do trepro1

meldet sich TREPRO1 mit dem Satz

"In der Technik herrschen folgende Trends: "

und liefert dann ein Menüangebot mit 12 Trends.

Der Nutzer wählt jeweils einen Trend aus, über den er mehr wissen möchte, der ihn in seiner Erfindersituation anregen soll.

Beispiel: Dimensionalität (1)

Die Trendmerkmale - beispielsweise Übergang von Punkt und Linie zu Fläche und Raum - werden angezeigt.

Das Beispiel bezieht sich auf Abscheider für verunreinigte strömende Gase (Venturi-Abscheider) /22/:

Anhand von Patenten aus über 30 Jahren und aus 6 Ländern kann der Übergang

- von der Eindimensionalität
- über die Zweidimensionalität
- bis zur Dreidimensionalität und Verschachtelung gezeigt werden,

Bild 2.32.

Aber die Entwicklung des Venturi-Abscheidens geht weiter, denn es wirken noch weitere Trends, etwa Trend 7: Mikroeffekte nutzen, beispielsweise magnetische Felder /22/.

#### e) Programmverbesserungen

Das Programm TREPRO1 ist vielfältig besserbar, z.B.:

- Programminhalt: Trendverknüpfung, Trendfunktion
- Programmform : Komfort, Kontrollen
- Programmumfang: (entsprechend Inhalt und Form)
- Dateiinhalt : Beispiele aus Zweigen usw.
- Dateiform : mehrstufig, evtl. auch invers:  
ein Objekt - mehrere Trends
- Dateiumfang : (entsprechend Inhalt und Form).

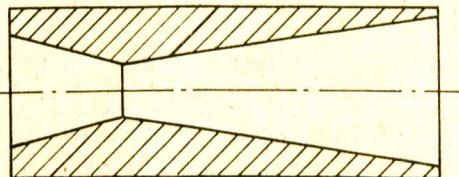
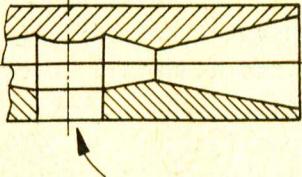
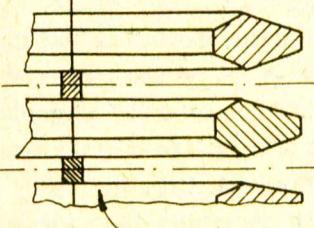
<p>Trend : Nutze alle Dimensionen!</p>		
<p>Beispiel : Venturi - Düse als Abscheider</p>		
Linie (eindimensional)	<p>Venturi - Düse</p> 	<p>G. B. Venturi um 1800</p>
	<p>Verbesserung : Hintereinanderschalten von Düsenabschnitten</p>	<p>als Abscheider um 1946</p>
Ebene (zweidimensional)	<p>Venturi - Teller</p> 	<p>I. Rechenberg nach 1970</p>
	<p>Verbesserung : vertikal bewegliche Teller (Steuerbarkeit!)</p>	<p>SU 237 806</p>
Raum (dreidimensional)	<p>Venturi - Ringe</p> 	<p>Ovotron Corp. (CA, DE, GB, FR)</p>
	<p>Verbesserung : Anordnung auf einer Schraubenlinie („Spirale“); Verschachtelung in „Spirale“</p>	<p>US 3 599 398 SU 502 645</p>

Bild 2.32.

### 2.2.7. ASSPRO1 - ein Assoziationsprogramm

#### a) Programmidee

ASSPRO1 unterstützt das Kreieren von Objekten und Merkmalen durch Assoziieren:

Es gibt unendlich viele Objekte mit jeweils unendlich vielen Merkmalen. Also gibt es auch

- "viele" Objekte  $O_B$  mit Merkmalen  $M_B \approx M_A = M_{\text{soll}}$ , aus denen neue technische Objekte  $O_A$  mit Merkmalen  $M_A$  entwickelbar sind
- "viele" Objekte  $O_B$  mit Merkmalen  $M_B$ , mit denen vorhandene technische Objekte  $O_A$  um Merkmale  $M_B = M_A$  bereicherbar sind,

Bild 2.33.

Beispiel:

Eine Vorrichtung zum Entfernen von Harnleitersteinen ist zu erfinden /23/:

- Merkmal  $M_A = M_{\text{soll}}$  ist
  - sehr unscharf: Harnsteine entfernend
  - noch unscharf: in Harnleiter einföhrbar, dort vergrößerbar und den Harnstein umschließend
- Objekte  $O_B$  sind beispielsweise
  - Regenschirme (aufspannbar)
  - Luftballons (aufblasbar)
- Objekt  $O_A$  wird dann ein Harnsteinentferner, der
  - in den Harnleiter einföhrbar ist
  - vor und hinter dem Harnstein aufblasbar ist
  - samt Stein aus dem Harnleiter ziehbar ist,

Bild 2.34.

ASSPRO1 findet die Objekte  $O_B$  zufällig:

- das Programm erzeugt eine n-stellige Zufallszahl
- die Datei enthält  $10^{n-1}$  Objektbeschreibungen
  - (mit den Merkmalen: Bestandteile, Eigenschaften, Beziehungen)
- das Programm liefert diejenige Objektbeschreibung, deren Satznummer die Zufallszahl ist.

Der Erfinder entwickelt dann aus  $O_B$  neue oder verbesserte  $O_A$ .

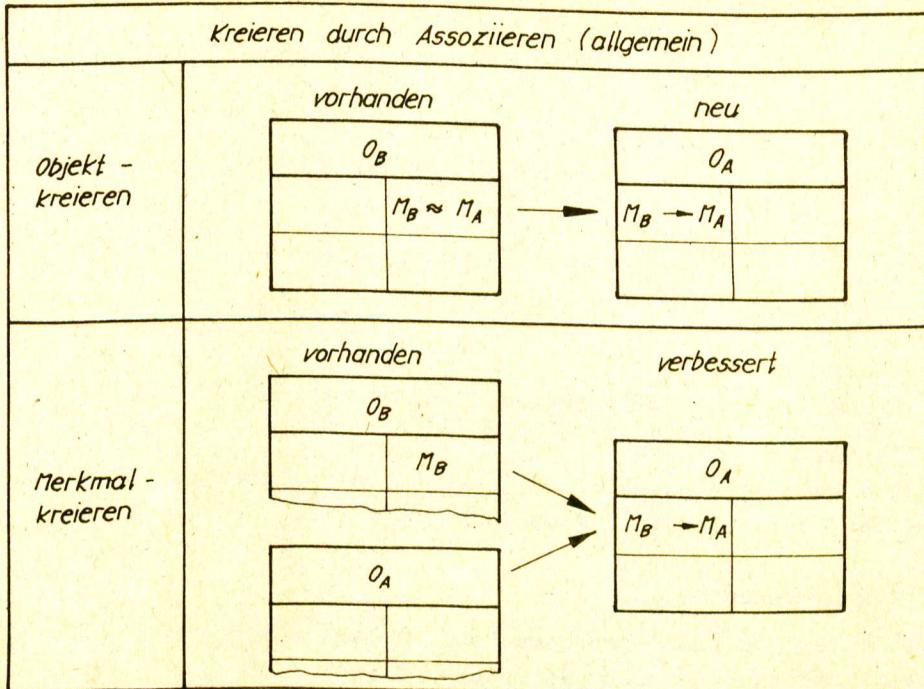


Bild 2.33.

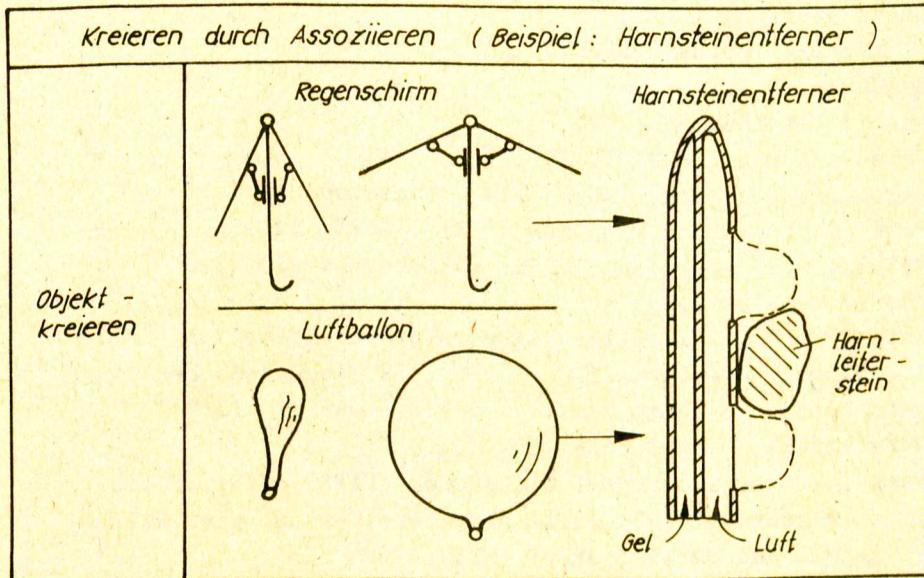


Bild 2.34.

b) Programmdaten

ASSDAT1 enthält einfache Beschreibungen einfacher Objekte mit ihren Merkmalen:

- Objektnummer (0Z% als ganze Zahl)
- Objektname (NA<sup>X</sup> als Zeichenkette)
- Bestandteile (T<sub>1</sub><sup>X</sup> ... T<sub>3</sub><sup>X</sup>)
- Eigenschaften
  - . Form (FO<sup>X</sup>)
  - . Struktur (ST<sup>X</sup>)
  - . Material (MA<sup>X</sup>)
- Beziehungen
  - . Funktion (FU<sup>X</sup>) {genauer: Nutzfunktion}
  - . Anordnung (AN<sup>X</sup>) {bei Verfahren: Ablauf}
  - . Handhabung (HA<sup>X</sup>) .

ASSDAT1 enthält vorerst nur 9 Objekte, beispielsweise 1 Haus, 2 Koffer, 3 Bleistift, 7 Wasserglas, 9 Motor.

c) Programmbeschreibung

ASSPR01 ist ein BASIC-Programm, s.a. Anlage 5.1.

ASSPR01 besteht aus einem Vorspann und zwei Szenen:

Vorspann:

- Konstante für Zeilenlöschen definieren (LET)
- Datei eröffnen (OPEN)
- Bildschirm löschen (PRINT)
- Menüauswahl (IF ...)

Szene 1: Dateiaufbau

- Daten anfordern (PRINT, INPUT)
- Satz aufbauen (LET) (Komma als Trennzeichen)
- Satz einordnen und ausgeben (FIELD, LSET, PUT)
- Weiterarbeit erfragen (INPUT, weiterer Dateiaufbau)

Szene 2: Dateinutzung

- Anfangswert für Zufallszahlgenerator bereitstellen (LET)
- Zufallszahl erzeugen (LET ZZ=RND) (ZZ liegt zwischen 0 und 1)
- Zufallszahl in ganze Zahl von 0 bis 9 wandeln (LET)
- Zufallszahl 0 ausblenden (IF ...)
- Satz (Assoziationsobjekt O<sub>B</sub>) suchen (FIELD, GET, IF ...)
- Satz ausgeben (PRINT) (hier unaufbereitet, s. aber PATPR01)
- Programm- und Dateiabschluß (END),

Bild 2.35.

```
1000 REM Programm ASSPRO1
1010 LET ZV¤=STRING¤(24,10)
1020 OPEN "R",#1,"ASSDAT1",128
1030 PRINT ZV¤
1040 INPUT "Kennwort (Thema) : ",KW¤
1050 PRINT
1060 PRINT "A - Aufbau der Datei"
1070 PRINT "N - Nutzung der Datei"
1080 INPUT "Eingabe (A oder N): ",EG¤
1090 IF EG¤="n" OR EG¤="N" THEN 1360
1100 REM: Szene 1: Dateiaufbau
1110 INPUT "Objektnummer : ",OZ%
1120 INPUT "Objektname : ",NA¤
1130 PRINT "Objektmerkmale"
1140 PRINT "- Bestandteile:"
1150 INPUT ". Teil 1 : ",T1¤
1160 INPUT ". Teil 2 : ",T2¤
1170 INPUT ". Teil 3 : ",T3¤
1180 PRINT "- Eigenschaften:"
1190 INPUT ". Form : ",FO¤
1200 INPUT ". Struktur : ",ST¤
1210 INPUT ". Material : ",MA¤
1220 PRINT "- Beziehungen:"
1230 INPUT ". Funktion : ",FU¤
1240 INPUT ". Anordnung : ",AN¤
1250 INPUT ". Handhabung : ",HA¤
1260 LET K1¤=STR¤(OZ%)+", "+NA¤+", "+T1¤+", "+T2¤+", "+T3¤+", "
1270 LET K2¤=FO¤+", "+ST¤+", "+MA¤+", "+FU¤+", "+AN¤+", "+HA¤
1280 LET KA¤=K1¤+K2¤
1290 FIELD #1,128 AS K¤
1300 LSET K¤=KA¤+","
1310 PUT #1,OZ%
1320 INPUT "weiter ? (j): ",EG¤
1330 PRINT
1340 IF EG¤="j" OR EG¤="J" THEN 1100
1350 GOTO 1520
1360 REM Szene 2: Dateinutzung
1370 REM Datei ist bereits eroeffnet!
1380 LET ZA=ASC(KW¤)
1390 RANDOMIZE (ZA)
1400 LET ZZ=RND
1410 LET OZ%=INT(10*ZZ)
1420 IF OZ%=0 THEN 1400
1430 FIELD #1, 128 AS K¤
1440 GET #1,OZ%
1450 IF EOF(1)=-1 THEN 1400
1460 PRINT ZV¤
1470 PRINT "Versuchen Sie, mit folgendem Objekt zu assoziieren:"
1480 PRINT
1490 PRINT K¤
1500 INPUT "weiter ? (j): ",EG¤
1510 IF EG¤="j" OR EG¤="J" THEN 1400
1520 END
```

Bild 2.35.

#### d) Programmnutzung

Nach dem Aufruf

LOAD "ASSPRO1" RUN

fordert ASSPRO1 ein Kennwort an und zeigt das Menü

- A - Aufbau der Datei
- N - Nutzung der Datei.

Auf die Eingabe a oder A (oder beliebiges Zeichen außer n oder N) werden Objektnummer, Objektname und Objektmerkmale angefordert.

Die Sätze werden auf Diskette ausgegeben; anschließend wird Weiterarbeit erfragt.

Auf die Eingabe n oder N erscheint eine zufällig aus der Datei ausgewählte Objektbeschreibung  $O_B$ , beispielsweise  
2, Koffer, Boden, Rahmen, Deckel, kubisch, aufklappbar, Pappe/Leder/  
Textilien, huellen, beim Menschen, tragbar,

oder

3, Bleistift, Huelle, Mine, Spitze, zylindrisch, verbunden, Holz/  
Plaste/Graphit, schreiben, in Hand, beruehren, .

Der Erfinder assoziert nun:

- Was hat ein Koffer mit Bungalows zu tun ?
- Was haben Koifer und Bleistift mit Skibern zu tun ?

Beispiele für Assoziationen zeigt Bild 2.36.

#### e) Programmverbesserungen

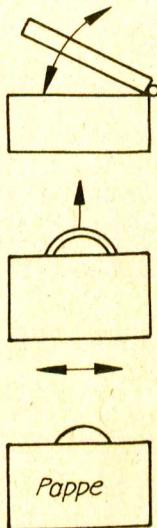
Das Programm ASSPRO1 ist vielfältig verbesserbar, z.B.:

- Programminhalt: Assoziationsketten nach BUŠ, s. /24/
- Programmform : Komfort, Kontrollen, Korrekturen
- Programmumfang: Ausgabeauffbereitung
- Dateiinhalt : Objekte aus Alltag, Natur und Technik,  
aus Märchen- und Phantastik-Welt
- Dateiform : strukturierte Dateien
- Dateiumfang : mehr Objekte, mehr Merkmale.

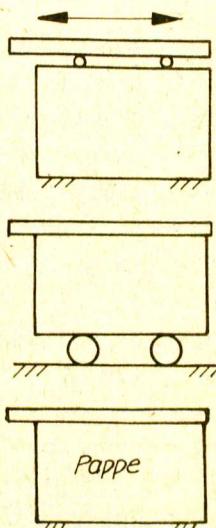
Aufgabe: Bungalows sind zu verbessern.

Lösungsansätze:

Koffer  $O_B$



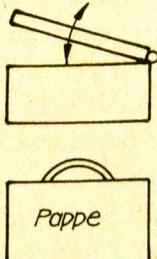
Bungalow  $O_A$



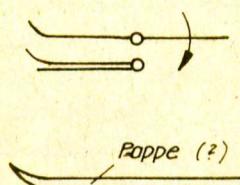
Aufgabe: Skier sind zu verbessern.

Lösungsansätze:

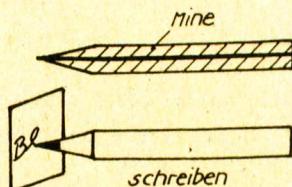
Koffer  $O_B$



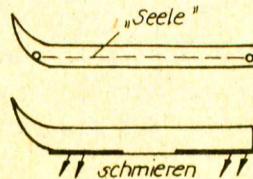
Skier  $O_A$



Bleistift  $O_B$



Skier  $O_A$



### 3. Beispiele

#### 3.0. Übersicht

Die vorgestellten Programme sind nur eine kleine Auswahl aus Hilfsmitteln, die HEUREKA enthalten soll.

Beispielsweise fehlen

- EVO PRO1 - ein Evolutions simulierungsprogramm
- STUPRO1 - ein Struktur konkretisierungsprogramm zum Kombinieren und Variieren  
(s. dazu Abschnitt 4.1.)
- WIRPRO1 - ein Wirkprinzip suchprogramm für technologisch nutzbare naturwissenschaftliche Wirkprinzipien.

Diese Wirkprinzipien sind bereits mit folgenden Merkmalen beschrieben worden:

- Wirkung (bezogen auf das zu fertigende Objekt)
- Wirkvorgang 2 (im Objekt)
- Wirkenergie, innen
- Wirkvorgang 1 (von Umgebung)
- Wirkmittel
- Wirkenergie, außen,  
s. /25/ und Bild 3.1.

Außerdem hat MECK für jedes Prinzip ein Katalogblatt mit Bild-, Zahlen- und Quellenangaben aufgebaut.

Für die Suche wird z.Z. die Sichtlochkartei WIESEL (Wiederhol-objekt-Selektion) genutzt. WIESEL ist eine maschinell herstellbare Sichtlochkartei für inverse Speicherung und flexible Suche:

- eine Maschinenlochkarte - ein Merkmal - alle Nummern von Objekten, die dieses Merkmal haben:
  - . Spalten 1...50 enthalten die Zehner und Hunderter
  - . Zeilen 0...9 enthalten die Einer der Objektnummern
  - . Spalten 51...80 dienen der Merkmalbenennung
- ein ausgewählter Maschinenlochkartenstapel - mehrere Merkmale - Durchblick bei nur denjenigen Nummern, deren Objekte alle Merkmale des Stapels haben,

Bild 3.2.

An zwei neuen Erfindungen, nämlich

- Verfahren für Feststofftrennung (Gestein, Beton)
- Vorrichtung für Rotationsenergiemumformung (Anlaufen, Wenden), wird nun gezeigt, daß die Programme für Erfinder Anregungen geben können.

naturwissenschaftliche Kurzbeschreibung: induktives Härteln		
Merkmale		
Merkmalkategorie	Merkmalausprägung	Kurzwort für Merkmal
Wg (Wirkung)	Stoffzusammenhalt beibehält	WG : BEIBEHALTEN
WV <sub>2</sub> (Wirkvorgang 2)	Gefüge umlagern	WV <sub>2</sub> : UMLAGERN
Wi (Wirkenergie, innen)	thermische Energie	WI : THERMEN
WV <sub>1</sub> (Wirkvorgang 1)	induktive Erwärmung	WV <sub>1</sub> : INDUKTERW
WM (Wirkmittel)	elektromagnetisches Feld	WM : FELD
Wa (Wirkenergie, außen)	elektromagnetische Energie	WA : ELMAG

Bild 3.1.

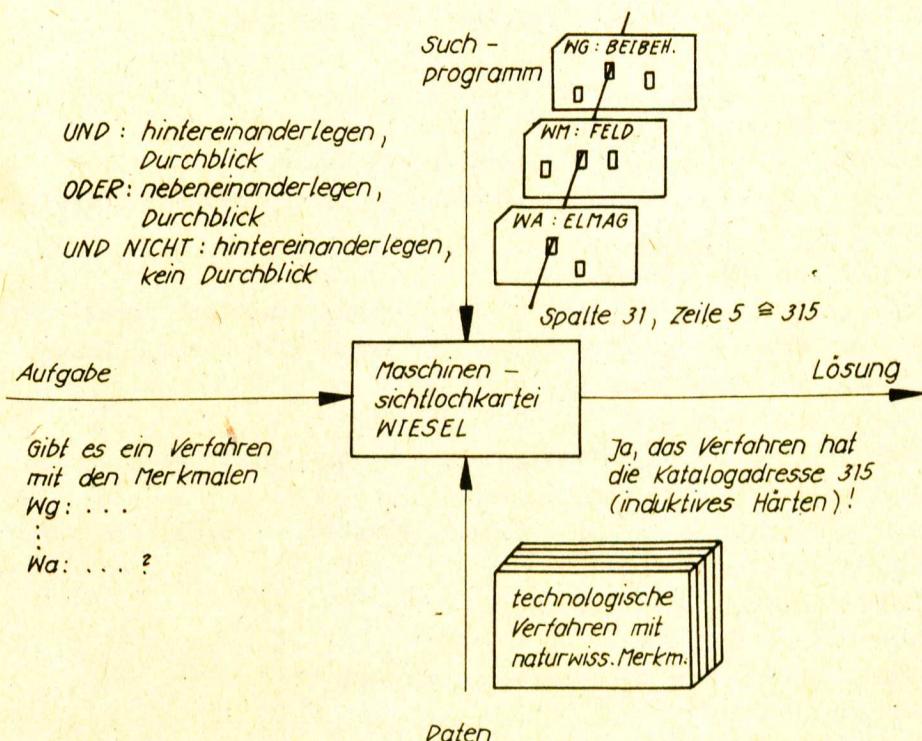


Bild 3.2.

### 3.1. Feststofftrennung

Situation:

Trennen von Feststoffen, insbesondere Zerkleinern von Baustoffen (Gestein, Beton) in definierte Formen und Größen, ist technologisch unausgereift. Der Erhöhung der Produktivität steht die geringer werdende Zuverlässigkeit der Werkzeuge (Verschleiß der Meißel und Sägen) entgegen.

Aufgabe:

Es ist ein Verfahren zum Trennen von Gestein und Beton so zu entwickeln, daß die Arbeitsmittel nicht verschleifen.

Lösungsansätze:

- ASSPRO1 liefert beispielsweise  
"1 Haus, ... , auf Erde, ..." und assoziiert "Verwitterung"
- TREPRO1 liefert neben "6 Energie" mit Merkmal "Impulstechnik" "7 Mikroeffekte nutzen" mit dem Merkmal "Übergang von Mechanik zu Hydraulik, Elektrik, Akustik, Optik" (Laser!)
- WIRPRO1 liefert rund 40 Trennverfahren, darunter  
"Laser-Strahl"  
"Sauerstoff-Lanze" (Verwitterung!)  
"Wasser-Strahl" (Verwitterung!)  
"Explosion" (Mikroexplosion von Wasser!)
- WIDPRO1 liefert für Produktivität (39) kontra Zuverlässigkeit (27)  
"1. Zerlegung" (Aufgabe)  
"35 Aggregatzustandsänderung" (Schmelzen! künstlich Verwittern)  
"10 vorherige Wirkung" (Wechsel: Erwärmen-Abkühlen?! Spannungen erzeugen, dann nur auslösen?!)  
"28 Mechanik ersetzen" (s. TREPRO1)
- EFFPRO1 liefert für beliebigen Eingang und für Ausgang von Objektteilen 17 Verfahren, darunter  
"Laser-Strahl-", "Wasser-Strahl-", "Elektronen-Strahl-Schneiden"  
"Ätzen" (Verwitterung!)  
"Explodieren" (Verwitterung!).

Lösung: DD-WP 230 813 /26/

Laser-Ablations-Trennverfahren, gekennzeichnet dadurch, daß

- ein Impuls-Laser
- mit dunklem Flüssigkeitsfilm

Mikroexplosionen und Mikroablationen erzeugt.

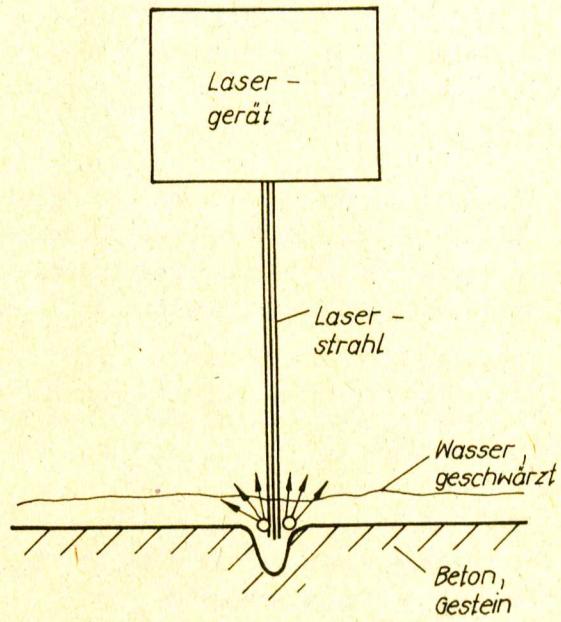


Bild 3.3.

### 3.2. Rotationsenergieumformung

Situation:

Umformen von Rotationsenergie geschieht mit Getrieben, auch mit Motoren und Kupplungen.

Für gleichzeitig

- große Übersetzungen
- stufenloses Verstellen der Übersetzung
- stufenloses Ändern der Abtriebsdrehrichtung,

also für Verhältnisse, die in Arbeitsmaschinen, Fahrzeugen, Industrierobotern, Geräten und Spielzeugen vorkommen können, fehlt eine kompakte Umformeinrichtung.

Aufgabe:

Es ist ein Getriebe so zu entwickeln, daß bei großen Übersetzungen die Abtriebsdrehrichtung geändert werden kann, ohne daß sich Art und Anzahl der Wirkpaarungen ändern und obwohl die Antriebswinkelgeschwindigkeit in Größe und Richtung gleich bleibt.

Lösungsansätze:

- EFFPR01 liefert für Eingang "Winkelgeschwindigkeit" und für Ausgang "Winkelgeschwindigkeit"  
"Hebel-Effekt"  
"Momentanpol-Effekt"  
und für Ausgang "große/kleine Winkelgeschwindigkeit"  
"Umlaufgetriebe" (Planetengetriebe)  
"Wellgetriebe" (Harmonic Drive)
- WIDPRO1 liefert für Geschwindigkeit (9) kontra Wirkdauer (15)  
"3 örtliche/zeitliche Anpassung" (Wirkpaarung wächst/schrumpft ein?! Wirkpaarung wirkt/wirkt nicht?)  
"19 Impulse, Impulspausen" (Schrittmotor!)  
"35 Aggregatzustand/Elaste" (Pneumopaarung? lastabhängig!)  
"5 Kopplung" (Doppelgetriebe!).

vorläufige Lösung:

- Hebel hat alle Geschwindigkeiten: positiv - null - negativ.
- Bei Rollbewegung ist der Berührungsrand der Momentanpol.
- Doppelplanetengetriebe haben am Abtrieb keine Drehzahl, wenn das Planetenradpaar 3-4 durchmessergleich ist (denn der Berührungsrand 3-6 ist der Momentanpol), Bild 3.4., Bild 3.5. und Bild 3.6.

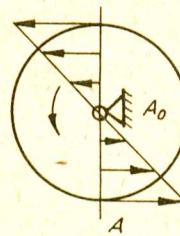


Bild 3.4.

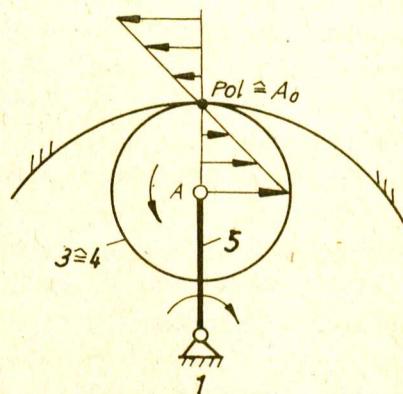


Bild 3.5.

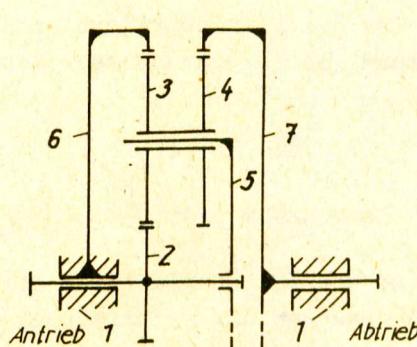
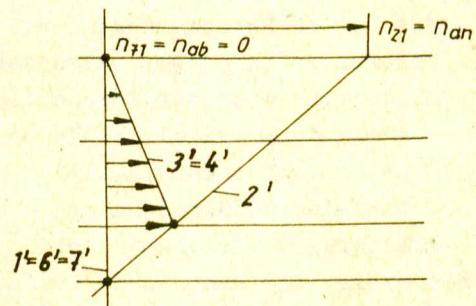


Bild 3.6.



Ist das Planetenrad 4 etwas kleiner als das Planetenrad 3, so wird die Abtriebsdrehzahl  $n_{71} = n_{ab}$  klein und zur Antriebsdrehzahl  $n_{21} = n_{an}$  gleichsinnig gerichtet,  
Bild 3.7.

Ist das Planetenrad 4 etwas größer als das Planetenrad 3, so wird die Abtriebsdrehzahl  $n_{71} = n_{ab}$  klein und zur Antriebsdrehzahl  $n_{21} = n_{an}$  gegensinnig gerichtet,  
Bild 3.8.

Eine geringfügige Durchmesserdifferenz zwischen den Paarungen 3-6 und 4-7 führt also entweder zum Gleichlauf oder zum Gegenlauf des Abtriebs gegenüber dem Antrieb.

Diese mindestens geringfügige Durchmesserdifferenz ist beispielsweise erreichbar durch

- elastische Verformung der Paarung (Kegelpaarung, Axialverschiebung)
- pneumatische oder hydraulische Verformung (Schlauchpaarung)
- thermische Verformung (Festkörperpaarung 3-6 und 4-7 mit unterschiedlichen Wärmedehnzahlen),

Bild 3.9.

EFFPR01 lieferte hierzu etwa 20 Effekte, von denen die oben angeführten besonders brauchbar waren.

Lösung: DD-WP-Anm. 283 700 ff. /10/

Reibradplanetengetriebe als Anlauf- und Wendegetriebe, gekennzeichnet dadurch, daß

- die beiden Reibpaarungen zwischen den Planetenrädern und den Reibrädern in einem Bezugszustand gleiche Durchmesser haben
- mindestens eine der Reibpaarungen
  - . mechanisch radial (Schläuche)
  - . mechanisch axial (Kegel)
  - . thermisch (Wärmedehnung)geringfügig verformbar ist.

Beispielsweise kann Temperaturerhöhung bei Überlast das Stillstehen des Abtriebs oder gar das Umkehren der Drehrichtung bewirken (Bohrer entspannt Bohrung usw.).

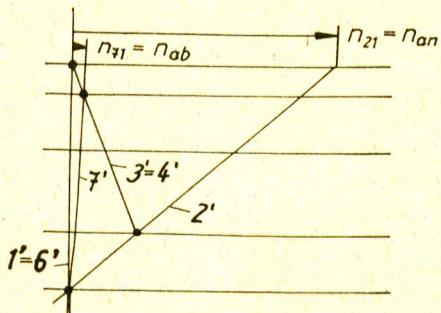
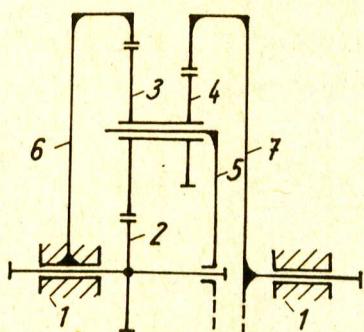


Bild 3.7.

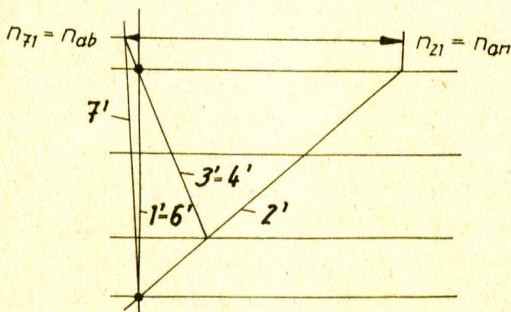
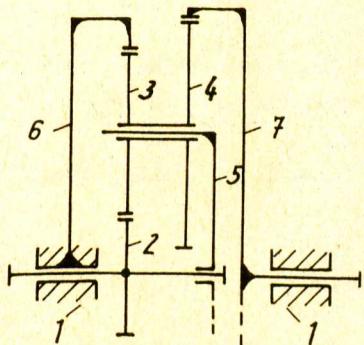


Bild 3.8.

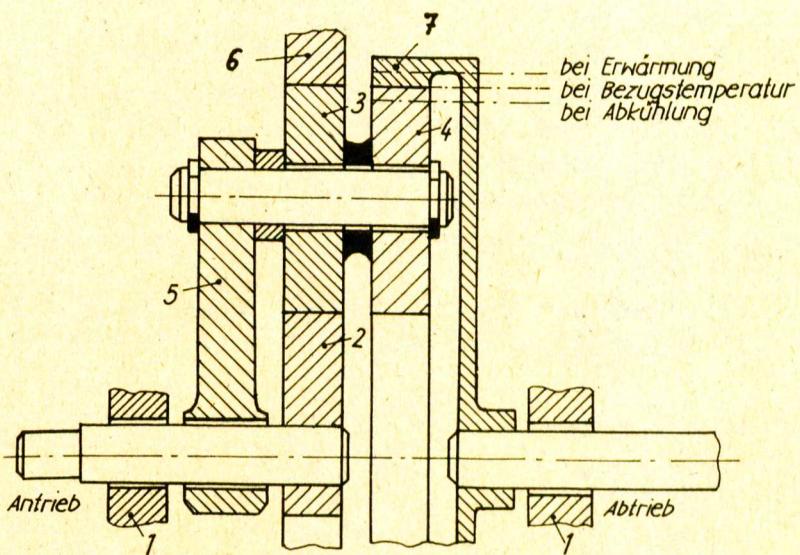


Bild 3.9.

#### 4. Ausblick

##### 4.0. Übersicht

Neben vielen kleinen Verbesserungen an den Programmen,  
beispielsweise

- Vervollkommnung der Menütechnik
  - Einführung von Kontrollen und Korrekturmöglichkeiten
  - Vergrößerung der Dateien,
- sind einige große Verbesserungen denkbar und machbar,  
beispielsweise
- die Lösung des Umfangsproblems (oder auch des Komplexionsproblems) beim Kombinieren
  - die Lösung des Formproblems (oder auch des Kommunikationsproblems) beim Nutzen von Grafik usw.
  - die Lösung des Inhaltsproblems (oder auch des Kognitionsproblems) beim Nutzen von Wissen, nicht nur von Daten,  
Bild 4.1.

Der kybernetische Wissenschaftszweig "Künstliche Intelligenz"  
stellt dafür die Verfahren bereit:

- das Umfangsproblem wird mit  
Dekomponieren, Reduzieren und Sortieren gelöst (KOLBE /29/)
- das Formproblem wird mit  
ähnlichen Verfahren - angewandt auf Bild- und Situationsanalyse - und mit Transformieren gelöst
- das Inhaltsproblem wird mit  
Deduzieren (BIBEL /27/) und Analogisieren (KLIX /28/) gelöst.

Im allgemeinen sind die derzeit verfügbaren Personalcomputer  
zu klein für diese Verfahren. Zwei Programmideen, nämlich

- STUPRO1 - eine Strukturkonkretisierungsidee
- EXPPRO1 - eine Expertensystemidee,

sollen aber zeigen, was abgerüstet derzeit und aufgerüstet künftig realisierbar ist.

Der Weg von der "Künstlichen Intelligenz" zur "Künstlichen  
Kreativität" ist unaufhaltsam !

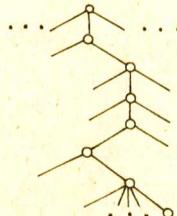
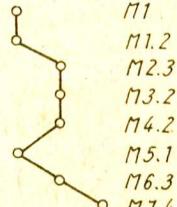
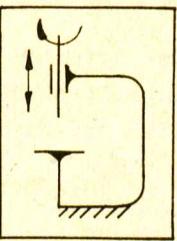
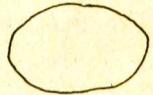
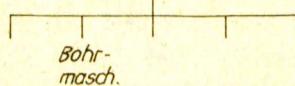
Komplexionsproblem (Kombination)	
<p>Umfang groß</p> <p>Objekt aus Merkmalen: Bohrmaschine</p>  <p>Beispiel: 10 Ebenen (Bestandteile) 10 Verzweigungen je Ebene</p>	<p>Welche Bohrmaschine aus <math>10^{10}</math> Möglichkeiten ist die beste?</p>  <p>bestes Objekt</p>
Kommunikationsproblem (Transformation)	
<p>Form verschieden</p> <p>Text über Objekt in Sprache 1</p>  <p>Grafiktext</p>	<p>Sprache 2</p> <p>„Maschine zum Herstellen relativ kleiner Rund - Löcher“</p> <p>Schrifttext</p>
Kognitionsproblem (Deduktion, Assoziation)	
<p>Inhalt unklar</p> <p>Menge der Objekte 1</p> <p>Werkzeugmasch.</p>  <p>Erodiermaschinen</p> 	<p>Objekte 2</p> <p>Bohrmaschinen</p>  <p>Bohrmaschinen</p>  <p>Welche Beziehungen haben die Objekte bzw. Begriffe?</p> <p>Werkzeugmaschinen</p>  <p>Ober - begriff Unter - begriff</p> <p>Ahnlichkeit</p>

Bild 4.1.

#### 4.1. STUPRO1 - eine Strukturkonkretisierungsidee

STUPRO1 unterstützt das Strukturkonkretisieren von Objekten, also das Kombinieren mit massenhaft Komplexionen:

Jedes technische Objekt hat einige Teifunktionen zu erfüllen, beispielsweise 10.

Jede Teifunktion kann von verschiedenartigen Bestandteilen realisiert werden, beispielsweise von 20 je Teifunktion.

Es entstehen  $10^{20}$  Kombinationsmöglichkeiten,  $10^{20}$  Komplexionen.

Allein für die Erzeugung dieser Komplexionen würde ein schneller Rechner Millionen Jahre benötigen!

Mit Hilfe der Verfahren der "Künstlichen Intelligenz" ist diese Aufgabe aber in Minuten bis Tagen lösbar!

Aufgabe:

- Gegeben ist die Beschreibung eines aufzubauenden technischen Objekts, und zwar als Graph oder Matrix der notwendigen abstrakten Beziehungen seiner abstrakten Bestandteile,  
Bild 4.2.

Beispiel: Automobil

- . Beschreibung des Objekts: Blockschaltbild des Automobils
- . Beschreibung der Bestandteile: Blöcke für Antrieb (Motor), Übertragung (Getriebe), Abtrieb (Räder), Hülle und Stützen (Karosserie) usw.
- Gegeben ist weiterhin die Beschreibung der möglichen konkreten Bestandteile des Objekts, und zwar mindestens als Matrix ihrer möglichen Beziehungen zueinander (Kopplungsmatrix),  
Bild 4.3.

Beispiel: Automobil

- . konkrete Bestandteile des Antriebsorgans sind Ottomotor oder Dieselmotor oder Elektromotor oder ...
- . konkrete Beziehungen können beispielsweise haben Otto- und Dieselmotoren mit dem Kraftstofftank und Elektromotoren mit einer Batterie als Energiespeicher (Markierung mit "1"); keine Beziehungen zueinander haben jedoch Elektromotoren mit dem Kraftstofftank (Markierung mit "0", s. aber Brennstoffbatterie der Zukunft).
- Gegeben ist möglichst noch eine Bewertung jedes konkreten Bestandteils, beispielsweise seine Masse, die Kosten, die Beschaffbarkeit u.ä.

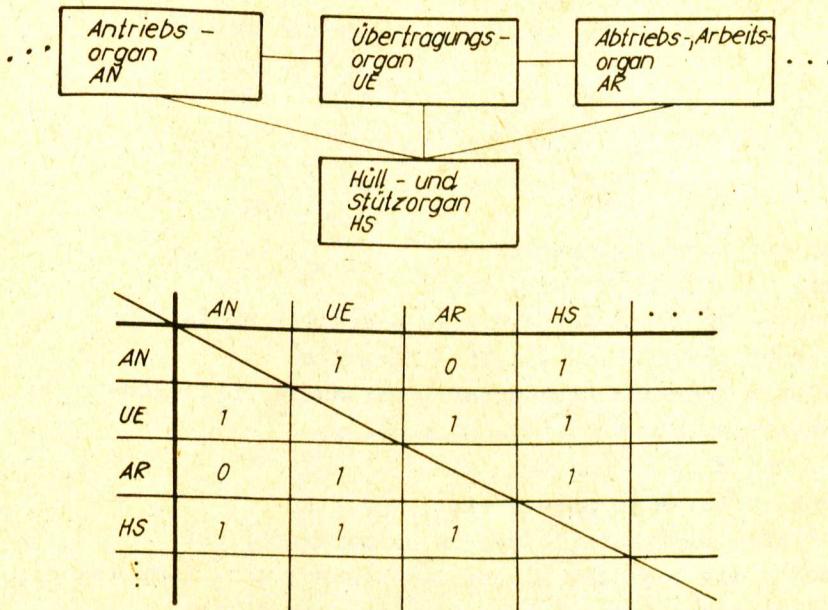


Bild 4.2.

Tabelle zur Struktur eines technischen Systems:

	AN	UE	AR	HS	EIN
OM	1	0	1		KT BA
PM					1 0
AN					1 0
EM					0 1
...					
UE					
AR					
HS					
EIN					

Bild 4.3.

Lösung:

- Dekomposition des Strukturgraphen:

Der Graph wird vom Rechner automatisch in Teilgraphen zerlegt, die

- . starke oder viele innere Bindungen
- . schwache oder wenige äußere Bindungen

haben,

Bild 4.4.

- Reduktion der Kopplungsmatrix:

Die Matrix wird so reduziert, daß

- . irrelevante konkrete Bestandteile endgültig entfallen  
(nicht koppelbare, also nutzlose Elemente)
- . äquivalente konkrete Bestandteile vorläufig entfallen  
(gleichartige, gleichwertige Elemente),

Bild 4.5. und Bild 4.6.

- Sortierung in der Kopplungsmatrix:

Die abstrakten Bestandteile werden so sortiert, daß

- . einerseits die schlecht koppelbaren (komplexen, komplizierten) Bestandteile möglichst frühzeitig getestet werden
- . andererseits die schlecht bewerteten (teuren, schweren) Bestandteile möglichst wenig Einfluß auf die Lösungsgüte bekommen,

s. /29/.

- Suche nach hochbewerteten Lösungen

- . zunächst als vollständige Suche  
(Breitensuche)
- . danach als unvollständige Suche nach hochbewerteten Lösungen  
(Tiefensuche).

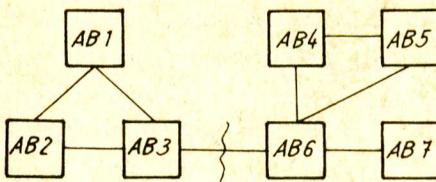
Beispiel:

Eine Elektroanlage mit

- 19 abstrakten Bestandteilen (Teilfunktionen)
  - 127 konkreten Bestandteilen (Bauelemente zur Auswahl)
- ergibt

- $1,45 \cdot 10^{15}$  Lösungen nach der Dekomposition in 5 Teilgraphen
- $2,79 \cdot 10^5$  Lösungen nach der Irrelevanzreduktion
- $2,88 \cdot 10^2$  Lösungen nach der Äquivalenzreduktion
- $1,00 \cdot 10^1$  Lösungen, die hochbewertet sind.

(Ergebnisse mit STRUKO /29/ auf Großrechner in 90 Sekunden).



Der Strukturgraph  $S$  wird in die Teilgraphen  $T_1$  und  $T_2$  zerlegt

Bild 4.4.

	AB1	AB2	AB3	
AB1	KB11 KB12 KB13	KB21 KB22	KB31 KB32 KB33	
AB2	KB21 KB22	0 1 0 1 0 1	0 0 1 1 0 0	
AB3	KB31 KB32 KB33	0 0 0 1 0 1 0 0 1	0 1 0 0 0 0 1 0 0	

Der konkrete Bestandteil  $KB_{12}$  ist irrelevant; die Zeilen und Spalten für  $KB_{12}$  werden gestrichen.

Bild 4.5.

	AB1	AB2	AB3	
AB1	KB11 KB13	KB21 KB22	KB31 KB32 KB33	
AB2	KB21 KB22	0 0 1 1	0 0 1 1 0 0	
AB3	KB31 KB32 KB33	0 0 1 1 0 0	0 1 0 0 1 0	

Die konkreten Bestandteile  $KB_{11}$  und  $KB_{13}$  sind äquivalent; die Zeilen und Spalten für  $KB_{13}$  werden gestrichen.

Bild 4.6.

#### 4.2. EXPPRO1 - eine Expertensystemidee

EXPPRO1 unterstützt das Finden neuer Ideen aus altem Wissen: Jede (?) neue Idee ist aus zwei oder mehreren wohlbekannten Ideen oder Tatsachen "zusammengedacht" (Bisoziation) oder "zusammendenkbar".

Es kommt nun darauf an, das vorhandene Wissen brauchbar zu formulieren und zu speichern. Bei den ersten verbreitetsten Praxisergebnissen der "Künstlichen Intelligenz", bei den Experten- oder Beratungssystemen, wird Wissen in Form von Regeln präsentiert:

- FF-Regeln:

WENN Fakt 1 gilt, DANN gilt Fakt 2.

oder:

IF f1 THEN f2

- FH-Regeln:

WENN Fakt 1 UND Fakt 2 gelten, DANN gilt in 80% aller Fälle auch die Hypothese 1.

oder:

IF f1 AND f2 THEN h1 ( $p=0,8$ )

- HH-Regeln:

IF h1 ( $p>0,4$ ) AND h2 ( $p>0,8$ ) THEN h3 ( $p=0,99$ ).

Beispiel:

Ein Stoff S1 (Krebszellen) ist gegeben, ist schädlich und muß vernichtet werden.

Ein Stoff S2 (Eisen) kann von S1 aufgenommen werden und/oder erwärmt S1 bis zu seiner Vernichtung.

Ein Feld F1 (elektromagnetisches Feld) kann S2 erwärmen.

Das ist der Stand der Technik: lokale Hyperthermie, s. /30/.

Gesucht ist nun ein Verfahren, das die sehr aufwendige Temperaturmessung und -regelung ( $43,5^{\circ}\text{C}$ ) entbehrlich macht.

EXPPRO1 prüft eventuell einige hundert Regeln, insbesondere dann aber diese Folge:

S1.1 - F1.2 - F1.1 (elektromagnetisches Wechselfeld)

F1.2 - S2.1 - S2.2 - S2.3 (Spray)

S2.1 - S2.5 (HEUSLER-Legierung mit  $43,5^{\circ}\text{C}$  CURIE-P.).

Das könnte ein neues Verfahren zur Krebsbehandlung sein,  
Bild 4.7.

Nr.	WENN ...	... DANN ...
S1	<b>Stoff 1</b> (stoffbezogene biologische Effekte / Fakten)	
S1.1	Krebszellen	sterben ab bei Erwärmung $> 43^{\circ}\text{C}$
S1.2	Krebszellen	absorbieren Eisen aus Blut
S2	<b>Stoff 2</b> (stoffbezogene physikalische Effekte / Fakten)	
S2.1	Magnetstoff	Eisen, HEUSLER - Legierung
S2.2	Magnetstoff	Stück, Korn, Pulver, Puder
S2.3	Puder	Nebel, Spray, Suspension
S2.4	Eisen	CURIE - Punkt $770^{\circ}\text{C}$
S2.5	HEUSLER - Legierung	CURIE - Punkt beliebig
F1	<b>Feld 1</b> (feldbezogene physikalische Effekte / Fakten)	
F1.1	Elektromagn. Wechselfeld	Induktion
F1.2	Induktion	Erwärmung im Magnetstoff bis CURIE - Punkt

Bild 4.7.

## 5. Anhang

## 5.1. Programmiersprache BASIC (Anlage 5.1)

### a) Programmbefehle

BASIC ist eine einfache Programmiersprache

- mit englischen Schlüsselwörtern für die Befehle
  - mit Sprungbefehl GOTO
  - mit Zeilennummern als Sprungadressen.

BASIC eignet sich besonders für das Programmieren von Rechenaufgaben (und zur Zeichenkettenverarbeitung).

BASIC wird akzeptiert vom

- BASIC-Interpreter (sofortige Programmausführung, langsam)
  - BASIC-Compiler (zunächst Übersetzung, danach Ausführung, mindestens doppelt so schnell).

Die wichtigsten Schlüsselwörter für Programmbefehle sind:

- REM - Bemerkung (von remark - Bemerkung)
  - INPUT - Eingabe
  - PRINT - Ausgabe (von to print - drucken, hier aber:  
Ausgabe auf Bildschirm)
  - LPRINT - Ausgabe auf Drucker (von line - Zeile)
  - LET - es sei (zu definieren, zu errechnen)
  - GOTO - Sprung (von go to - gehe zu ...)
  - IF ... THEN ... ELSE - wenn ... dann ... sonst
  - AND ... OR - und ... oder
  - FOR ... NEXT - für ... nun nächster Schleifenzyklus
  - END - Ende des Programms,

s. auch Zusammenstellung S. 87.

Die Konstanten und Variablen können deutschsprachige mnemonische Abkürzungen sein, und zwar

- für Ganzzahlen (integer, INT) mit %-Zeichen als Anhang
  - für Dezimalzahlen (real) ohne Zeichen
  - für Zeichenketten (string, STR) mit &-Zeichen als Anhang.

Die Programmentwicklung geschieht am besten in vier Etappen:

- 1. Etappe: Formulierung in Deutsch
  - 2. Etappe: Formulierung in strukturiertem Deutsch  
(deutsche Schlüsselwörter)
  - 3. Etappe: Formulierung in strukturiertem Englisch  
(englische Schlüsselwörter, deutsche Namen (Abk.))
  - 4. Etappe: Formulierung in BASIC und Test.

## Wichtige BASIC-Befehle

### 1. Rechenbefehle

LET	es sei ... (kann auch weggelassen werden)
+ - * /	wie in Mathematik: Punkt- vor Strichrechnung
^	Potenz; größere Bindung als Punktrechnung
( )	wie in Mathematik, sonst von links nach rechts
= > <	wie in Mathematik
abs bzw. ABS	wie in Mathematik
INT, SIN usw.	wie in Mathematik
AND, OR	wie in Logik
RND	Erzeugung von Zufallszahlen 0...1

### 2. Eingabe-Ausgabe-Befehle

INPUT	Eingabe
PRINT	Ausgabe (Bildschirm), nur PRINT ist Leerzeile
LPRINT	Ausgabe (Drucker)
TAB(position);	Tabulator, z.B. PRINT TAB(5); "ab 5. Position"

### 3. Steuerbefehle

IF...THEN...ELSE - wenn...dann...(sonst)  
FOR i ... NEXT i - für i ... nun weiter mit nächstem i  
GOTO zeilennr. - gehe zum Befehl mit der Zeilennummer

### 4. Zeichenkettenbefehle

VAL(K <sup>X</sup> )	Zahl aus den Ziffern in der Zeichenkette K <sup>X</sup>
STR <sup>X</sup> (Z)	Zeichenkette aus der Zahl Z
LEN(K <sup>X</sup> )	Länge der Zeichenkette K <sup>X</sup>
LEFT <sup>X</sup> (K <sup>X</sup> ,N)	Linker Teil der Zeichenkette K <sup>X</sup> , n lang
RIGHT <sup>X</sup> (K <sup>X</sup> ,N)	rechter Teil der Zeichenkette K <sup>X</sup> , n lang
MID <sup>X</sup> (K <sup>X</sup> ,M,N)	mittlerer Teil der Zeichenkette K <sup>X</sup> , ab m-ter Stelle n lang

### 5. Dateibefehle

OPEN	Datei eröffnen; "I" sequentiell, "R" wahlfrei
CLOSE	Datei schließen
PUT	Satz auf Datei ausgeben (Diskette)
GET	Satz von Datei eingeben (Diskette)

b) Programmentwicklung

1. Etappe: Formulierung in Deutsch

Zur weiteren Entwicklung des Erfindungswesens im Kombinat faßt die Leitung folgenden Beschuß:

- das Jahr 1987 wird "Jahr des Neuerers und Erfinders"
- für jede Neuerung und für jede Erfindung wird eine Sonderprämie bereitgestellt, deren Höhe vom Nutzen abhängt:
  - . für 1000.- Mark Nutzen beträgt die Prämie bei NV 25.- Mark
  - . für 100 000.- Mark Nutzen beträgt sie bei NV 1000.- Mark
  - . für diese Stützwerte ist eine Formel zu entwickeln
  - . für WP beträgt die Sonderprämie jeweils das Zweieinhalfache.

Für jede Neuerung und jede Erfindung ist die Prämie auszurechnen und auszudrucken.

2. Etappe: Formulierung in strukturiertem Deutsch

Bemerkung: Rechenprogramm 1

Löschen der Hauptspeicherbereiche

Löschen des Bildschirms

Eingabe: Name (Titel) des Neuerers/Erfinders (Neuerung/Erfindung)  
Nutzen in Mark

Neuerervorschlag oder Wirtschaftspatent

Frage im Rechner: wenn NV, dann Konstante gleich 10 oder 1  
wenn WP, dann Konstante gleich 25 oder 2,5

Rechnung: Sonderprämie = (Konstante/Nutzen<sup>0,2</sup>) · (Nutzen/100)  
(das ergibt für NV bei 1000.- M 25.- M Prämie  
und für NV bei 100 000.- M 1000.- M Prämie (K=10))

Druck: Name

Sonderprämie in Mark

Frage an Nutzer: wenn Weiterarbeit, dann Sprung nach Eingabe  
(oder nach Bildschirmlöschen)

Ende.

3. Etappe: Formulierung in strukturiertem Englisch

REMARK Rechenprogramm 1

CLEAR

LET (16x Leerzeile) PRINT (16x Leerzeile)

INPUT nam, nutz, art

IF art="NV" or art="nv" THEN k=10 : GOTO (übernächst)

IF art="WP" or art="wp" THEN k=25 ELSE (erneut) INPUT

LET son=k/nutz<sup>0,2</sup>\* nutz/100

LPRINT nam

LPRINT integer(son)

usw.

4. Etappe: Formulierung in BASIC, s. Bild 5.1.

```
100 REM Programm RECPR01
110 CLEAR
120 LET ZV¤=STRING¤(16,10)
130 PRINT ZV¤
140 INPUT "Name : ",NAM¤
150 INPUT "Nutzen in Mark: ",NUTZ
160 INPUT "NV oder WP ? : ",ART¤
170 IF ART¤="NV" OR ART¤="nv" THEN K=10 : GOTO 190
180 IF ART¤="WP" OR ART¤="wp" THEN K=25 ELSE 160
190 LET SON=K/NUTZ^.2*NUTZ/100
200 LPRINT NAM¤
210 LPRINT "Sonderpraemie in Mark: ";INT(SON)
220 LPRINT
230 PRINT
240 INPUT "weiter (j oder n) ? ",EG¤
250 PRINT
260 IF EG¤="J" OR EG¤="j" THEN 140
270 END
```

Bild 5.1.

## 5.2. Programmiersprache dBASE (Anlage 5.2)

### a) Programmbefehle

dBASE ist eine einfache Programmiersprache

- mit englischen Schlüsselwörtern

(s. aber DDR-Produkt REDABAS)

- ohne Sprungbefehl GOTO

- mit Blockorientierung

(strukturiertes Programmieren verlangt abgeschlossene Blöcke,

z.B. DO WHILE ... ENDDO

oder IF ... ELSE ... ENDIF).

dBASE eignet sich besonders für das Programmieren von Suchaufgaben (im Zusammenhang mit Datenbankverwaltung).

dBASE wird vom dBASE-Interpreter akzeptiert.

Die wichtigsten Schlüsselwörter für Programmbefehle sind:

- # - Bemerkung

- GET - Eingabe (von to get - erhalten)

- SAY - Ausgabe (von to say - sagen)

- STORE TO - speichere (von to store - aufspeichern)

- SET - setze (von to set - setzen)

- USE datei- Datei eröffnen (von to use - benutzen)

- GO TOP - Dateianfang

- PICTURE - Schablone (von picture - Bild

z.B. '!' - Wandlung in Großbuchstaben

oder '9' - einstellige Zahl)

- LOCATE FOR ... - Suche für ... (von to l. - ausfindig machen)

- DO WHILE ... ENDDO - tue solange gilt ... Ende des Blocks

- IF ... ELSE ... ENDIF - wenn ... sonst ... Ende des Blocks

- .AND. ... .OR. - und ... oder

- RETURN - Programmende (von to return - zurückgehen),

s. auch Zusammenstellung S. 91.

Die Konstanten und Variablen können deutschsprachige mnemonische Abkürzungen sein.

Die Art der Konstanten und Variablen (Zahl, Zeichenkette) wird mit STORE festgelegt.

Die Programmentwicklung geschieht am besten wieder in vier Etappen.

## Wichtige dBASE-Befehle

### 1. Rechenbefehle

+ - * /	wie in Mathematik: Punkt- vor Strichrechnung
( )	wie in Mathematik
= > <	wie in Mathematik
INT	wie in Mathematik: ganze Zahl
.AND., .OR.	wie in Logik
STORE ... TO	Speicherbefehl, z.B. STORE 0 TO NULL

### 2. Eingabe-Ausgabe-Befehle

INPUT TO	Eingabe
SAY...GET...READ	Eingabe mit Position und Maske
SAY	Ausgabe
	. SET FORMAT TO SCREEN auf Bildschirm (Standardausgabe)
	. SET FORMAT TO PRINT auf Drucker
Q Z,S	Tabulator für Zeile Z und Spalte S
PICTURE '...'	Maske, z.B. PICT '999' für dreistellige Zahl

### 3. Steuerbefehle

DO WHILE ... ENDDO	- tue solange gilt ... Ende
IF ... ELSE ... ENDIF	- wenn ... (sonst) ... Ende
DO CASE ... CASE ... ENDCASE	- tue im Falle ... Fall ... Ende

### 4. Zeichenkettenbefehle

VAL(...)	Zahl aus den Ziffern einer Zeichenkette
STR(...)	Zeichenkette aus einer Zahl
LEN(...)	Länge einer Zeichenkette
M (K,M,N)	mittlerer Teil einer Kette K ab m-ter Stelle n lang (s. LEFT, RIGHT, MID in BASIC)

### 5. Dateibefehle

USE dateiname	Datei öffnen
SELECT PRIMARY	erste Datei markieren und auswählen
SELECT SECONDARY	zweite Datei markieren und auswählen

### 6. Suchbefehle

GO TOP	Dateianfang
LOCATE FOR ...	Satzsuche von Dateizeigerposition aus

b) Programmentwicklung

1. Etappe: Formulierung in Deutsch

In der Fachliteratur und im Schriftverkehr sind Patente oft nur mit Angaben über Land und Patentnummer zitiert.

In einer Datei, die Patentdaten zu

- Land

- Patentnummer

- Patentbeschreibung (mindestens Klasse und Titel)

enthält, ist jeweils diejenige Patentbeschreibung zu suchen, für die Land und Patentnummer in den Rechner eingetastet werden. Die Daten sind auf Bildschirm auszugeben.

2. Etappe: Formulierung in strukturiertem Deutsch

Bemerkung: Suchprogramm 1

Systemnachrichten ausschalten

Bildschirm löschen

Datei eröffnen

Speicherplatz für Antwort benennen und löschen

Schleife: solange durchlaufen, bis die Frage nach der Weiterarbeit verneint wird

Speicherplatz für Antwort (einstellig, z.B. N), Land (zweistellig, z.B. DD), Patentnummer benennen

Eingabe: Land und Patentnummer

Datei auf Anfang stellen

Suche für Land(Datei)=Land(Eingabe) und Patentnummer(Datei)=Patentnummer(Eingabe)

Ausgabe: Land, Patentnummer, Patentbeschreibung (aus Datei)

Frage nach Weiterarbeit: wenn N oder n, dann ...

Schleifenende

Ende.

3. Etappe: Formulierung in strukturiertem Englisch

\* Suchprogramm 1

SET TALK OFF

ERASE

USE suchdatei

STORE (leer) TO antwort

DO WHILE antwort ungleich N

STORE (...) TO antwort, land1, patn1

SAY ...

usw.

4. Etappe: Formulierung in dBASE, s. Bild 5.2.

```
* Programm SUCPR01
set talk off
erase
use sucdat1
store '' to antwort
do while antwort<>'N'
  store '' to antwort
  store '' to land1
  store 0 to patn1
  @ 1, 0 say 'Welches Land (zweistellig) ? ';
    get land1 picture '!!!'
  @ 2, 0 say 'Welche Patentnummer ? ';
    get patn1 picture '9999999'
  read
  go top
  locate for land=land1 .and. patn=patn1 -----
  @ 3, 0 say '-----'
  @ 4, 0 say land
  @ 5, 0 say patn
  @ 6, 0 say patb
  @ 15, 0 say 'weiter (j oder n) ? ';
    get antwort picture '!!'
  read
enddo
return
```

Bild 5.2.

## 6. Quellen

- /1/ Stelzer, F.  
beispielsweise:  
Zweitakt-Brennkraftmaschine  
F o2 b 71/oo DE-OS 30 29 287
- /2/ Muslin, J.  
Maschinen des 20. Jahrhunderts  
(a. d. Russ.)  
Berlin: Verlag Technik 1974
- /3/ ...  
Dosiereinrichtung spart Kraftstoff.  
Der erste Schritt zum Wassermotor?  
der neuerer (1981) 7, S. 230-231
- /4/ Zwicky, F.  
Entdecken, Erfinden, Forschen im morphologischen Weltbild  
Zürich: Knaur 1966
- /5/ Thiel, R.  
Aufgabenstellungen in der Ingenieurarbeit und der dialektische Widerspruch  
Dt. Zeitschrift für Philosophie 29 (1981) 6, S. 651-662
- /6/ Herrig, D.; Müller, H.; Thiel, R.  
Technische Probleme - dialektische Widersprüche -  
erfinderische Widerspruchslösung  
Maschinenbautechnik 34 (1985) 6, S. 277-279  
und:  
Technische Probleme - methodische Mittel -  
erfinderische Lösungen  
Maschinenbautechnik 34 (1985) 7, S. 297-300
- /7/ Altschuller, G.S. (Al'tšuller)  
Erfinden - (k)ein Problem?  
(a. d. Russ. unter der Leitung von K. Willimczik)  
Berlin: Verlag Tribüne 1973
- /8/ Herrig, D.  
Design Theory for CAD Systems and CAD Objects  
in:  
Encarnacao, J; Krause, F.-L. (ed.)  
File Structures and Data Bases for CAD  
Amsterdam, New York, Oxford: North-Holland 1982
- /9/ Herrlich, M.; Zadek, G.  
KDT-Erfinderschule.  
Berlin: Kammer der Technik 1982  
(Lehrmaterial)
- /10/ Möws, H.; Herrig, D.; Wziontek, G.  
beispielsweise:  
Reibradplanetengetriebe als Anlauf- und Wendegetriebe  
F 16 h DD-WP-Anm. 283 700 und 283 701

- /11/ Koller, R.  
Konstruktionslehre für den Maschinenbau  
(2. Auflage)  
Berlin (W), Heidelberg, New York, Tokyo: Springer 1985  
Erstveröffentlichung der Matrix:  
Ein Weg zur Konstruktionsmethodik  
Konstruktion 23 (1971) 10, S. 388-400
- /12/ Roth, K.  
Konstruieren mit Konstruktionskatalogen  
Berlin (W), Heidelberg, New York: Springer 1982
- /13/ Schubert, J.  
Physikalische Effekte.  
Anwendungen, Beschreibungen, Tabellen  
(2. Auflage)  
Weinheim: Physik-Verlag 1984
- /14/ Kurth, J.; Busch, K.; Busch, H. (Buš, G.)  
Rechnergestützte Prinziperarbeitung in der Gerätekonstruktion  
Feingerätetechnik 30 (1981) 11, S. 490-492
- /15/ ...  
(unter der Leitung von Röhrs)  
Katalog physikalischer Effekte  
Dresden: Technische Universität, Sektion 10 1976  
(Mikrofiche)
- /16/ v. Ardenne, M.; Reball, S. u.a. (Hrsg.)  
Effekte und Prinzipien. Wissensspeicher  
(Arbeitstitel)  
Berlin: Dt. Verlag der Wissenschaften 1986
- /17/ Borodastov, G.V.; Denisov, S.D.; Efimov, V.A.;  
Subarev, V.V.; Kustov, V.P.; Gončarov, A.H.  
Verzeichnis physikalischer Erscheinungen und Effekte  
zur Lösung von Erfindungsaufgaben  
(a. d. Russ. von B. Kahmann)  
Moskau: Zentralinstitut für Information und Forschung  
der Atomwissenschaft und Atomtechnik 1979
- /18/ ...  
Weibliche Babys auf Wunsch  
ADN - SVZ (1986) vom 8.8., Beilage S. 3
- /19/ Altschuller, G.S. (Al'tšuller)  
Erfinden.  
Wege zur Lösung technischer Probleme  
(a. d. Russ. unter der Leitung von R. Thiel/H. Patzwaldt)  
Berlin: Verlag Technik 1984
- /20/ Zobel, D.  
Erfinderfibel.  
Systematisches Erfinden für Praktiker  
Berlin: Dt. Verlag der Wissenschaften 1985

- /21/ Herrig, D.; Herrig, Ch.  
Hubseilführung für Wippkrane mit Wippseilsystem  
B 66 c DD-WP-Anm. 287 790
- /22/ Čerbakov, V.  
Das Schicksal der Dinosaurier  
(a. d. Russ.)  
Technika i nauka (1980) 9, S. 14-15
- /23/ Pahl, G.  
Intuitiv betonte Methoden zur Lösungsfindung  
Konstruktion 24 (1972) 9, S. 373-376
- /24/ Polovinkin, A.I. (Hrsg.)  
Methoden der Suche neuer technischer Lösungen  
(a. d. Russ. unter der Leitung von J. Müller/B. Schüttauf)  
Halle, Berlin: ZIS und ZKI der AdW 1976  
(Technisch-wissenschaftliche Abhandlungen des ZIS 121,  
Informationen aus dem ZKI der AdW 5/1976)
- /25/ Herrig, D.; Meck, W.; Müller, H.  
Suche naturwissenschaftlicher Wirkprinzipien für die  
Entwicklung und Fertigung von Maschinen und Geräten  
mit dem WIESEL-Verfahren  
Wiss. Z. der TU Dresden 32 (1983) 4, S. 80-85
- /26/ Kahmann, B.; Finsterbusch, W.  
Laser-Ablations-Trennverfahren  
B 23 k 26/oo DD-WP 230 813
- /27/ Bibel, W.; Siekmann, J.H. (Hrsg.)  
Künstliche Intelligenz  
Berlin (W), Heidelberg, New York: Springer 1982  
(Informatik-Fachberichte 59)
- /28/ Klix, F.  
Über die Nachbildung von Denkanforderungen ...  
Z. für Psychologie 193 (1985) 3, S. 175-211  
(mit BASIC-Programm)
- /29/ Herrig, D.; Kolbe, W.  
Optimierung bei der Strukturkonkretisierung technischer  
Gebilde  
Konstruktion 33 (1981) 1, S. 25-28  
(Algol-Programme in:  
Kolbe, W.  
Untersuchungen zur Strukturkonkretisierung im technischen  
Entwurfsprozeß auf der Grundlage von Problemlösungstech-  
niken  
Berlin: Akademie der Wissenschaften 1979  
(Dissertation A))
- /30/ ...  
Neue Möglichkeiten der Krebsbehandlung  
ADN - SVZ (1986) vom 23.5., Beilage S. 3

