

Kessler, Wolfgang; Rueck, Günter-Friedrich

**Working Paper — Digitized Version**

## Zwei Programmsysteme zur maschinellen Anfertigung von Organigrammen

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 74

**Provided in Cooperation with:**

Christian-Albrechts-University of Kiel, Institute of Business Administration

*Suggested Citation:* Kessler, Wolfgang; Rueck, Günter-Friedrich (1980) : Zwei Programmsysteme zur maschinellen Anfertigung von Organigrammen, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 74, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/193901>

**Standard-Nutzungsbedingungen:**

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

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

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

**Terms of use:**

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

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

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

Nr. 74

ZWEI PROGRAMMSYSTEME ZUR MASCHINELLEN  
ANFERTIGUNG VON ORGANIGRAMMEN

Wolfgang Kessler\*      Günter-Friedrich Rueck\*

Februar 1980

\* Dipl.-Ing. Wolfgang Kessler

\* cand.rer.nat. Günter-Friedrich Rueck

Institut für Betriebswirtschaftslehre  
der Christian-Albrechts-Universität Kiel  
Lehrstuhl für Finanzwirtschaft  
Professor Dr. Reinhart Schmidt

INHALTSVERZEICHNIS

Seite

1	Einleitung .....	1
2	Problemstellung .....	2
3	Programme zur maschinellen Erstellung von Organigrammen .....	3
3.1	ORGAN.F4 - Organigramme auf einem Lineprinter .....	3
3.1.1	Anforderungen an die einzugebenden Daten .....	4
3.1.2	Programmablauf .....	5
3.1.3	Bedienung des Programms ORGAN.F4 .....	9
3.2	ORGPOS, ORGPLT - Organigramme auf einem Plotter .....	9
3.2.1	Anforderungen an die einzugebenden Daten .....	10
3.2.2	Programmablauf ORGPOS .....	11
3.2.2.1	Einlesen der Daten .....	11
3.2.2.2	Grobpositionierung der Elemente .....	11
3.2.2.3	Feinpositionierung der Elemente .....	12
3.2.2.4	Koordinatenausgabe mit Berechnung der Verbindungen .....	13
3.2.2.5	ORGPOS-Fehlermeldungen .....	14
3.2.3	Programmablauf ORGPLT .....	15
3.2.3.1	Einlesen der Daten .....	15
3.2.3.2	Koordinatentransformation .....	15
3.2.3.3	Einlesen und Zwischenspeichern der Element- beschriftungen .....	15
3.2.3.4	Plot der Elemente .....	16
3.2.3.5	Einlesen und transformieren der Verbin- dungen .....	16
3.2.3.6	Plot der Verbindungen .....	16
3.2.3.7	Seitenmarkierungen .....	17
3.2.3.8	ORGPLT-Fehlermeldungen .....	17
3.2.4	Bedienung der Programme auf der PDP 11/45 .....	18
3.2.4.1	Bedienung des Programms ORGPOS .....	18
3.2.4.2	Bedienung des Programms ORGPLT .....	19
	Literaturangaben .....	22

Anhang

A	Kurzbeschreibung der benutzten Systempro- zeduren
B	Beispiel - Output
B 1	Lineprinter-Output erzeugt von ORGAN.F4
B 2	Plotter-Output erzeugt von ORGPOS, ORGPLT
C	Vollständige Haupt- und Unterprogramme
C 1	ORGAN.F4
C 2	ORGPOS, INPUT, RANGE
C 3	ORGPLT, DISTNC, KTRANS, LTRANS, NEXT

Verzeichnis der Abbildungen und Tabellen

Seite

Abbildung 1	Druckmuster .....	6
Abbildung 2	Beispiel.....	7
Abbildung 3	Flußdiagramm für die Druckmuster- erkennung .....	8
Abbildung 4	Verbindungsformen unter Elementen .....	10
Abbildung 5	Direkte Verbindungen .....	13
Abbildung 6	Rechtwinklige Verbindungen .....	13
Abbildung B1	Lineprinter-Output einer Beispiel- organisation erzeugt von ORGAN.F4 .....	B2
Abbildung B2	Plotter-Output einer Beispielorgani- sation erzeugt von ORGPOS, ORGPLT .....	B3
Tabelle 1	Druckmusterzuordnung .....	7

## 1 Einleitung

"Eine Skizze sagt mehr als 1000 Worte".<sup>1</sup> Diese bei Praktikern weit verbreitete Meinung tritt besonders deutlich bei der Darstellung von Organisationsplänen hervor. Solche Organisationspläne sind allerdings nur dann von Nutzen, wenn sie aktuell sind. Die manuelle Anfertigung und ständige Aktualisierung sind mit erheblichen Kosten verbunden, so daß die Möglichkeit zur maschinellen Darstellung von Organisationsplänen wünschenswert erscheint.

Die Darstellung von Baumstrukturen mit Hilfe einer EDV-Anlage ist dann sehr aufwendig, wenn viele verschiedene Erscheinungsformen von einem Programm bearbeitet werden sollen. Um eine grafische Darstellung von Aufbauorganisationen zu erstellen, könnte man auf bereits vorhandene Methoden und Programme, wie sie z.B. in der Netzplantechnik bekannt sind, zurückgreifen.<sup>2</sup> Bei näherer Untersuchung der Aufgabenstellung zeigt sich jedoch, daß die in einem Netzplan auftretende Vielfalt von möglichen Verbindungen zwischen den Knoten eines Netzes in einem Unterstellungsdiagramm<sup>3</sup> nicht vorkommt. Es erscheint daher sinnvoll, einfachere Verfahren zur Anfertigung von Organigrammen zu entwickeln. Dabei soll das Organigramm als ein Teil des Organisationshandbuches den von Jope<sup>4,5</sup> aufgezeigten Anforderungen genügen.

---

<sup>1</sup> Friedrich, F.J., G.Ploch, R.Pohlenz, Ein System zur grafischen Darstellung von Vorgangsknoten-Netzplänen mit Hilfe elektronisch gesteuerter Zeichenmaschinen, in: Zeitschrift für Operations-Research, Band 19 (1975), S. B120. nach Händler, W., Bericht Nr.2, Fachtagung Computer Graphics der Gesellschaft für Informatik, Berlin, 19. bis 21. 10. 1971, S. 421.

<sup>2</sup> Vgl. Theis, W., Trommer, W., Ein Verfahren zur übersichtlichen maschinellen Darstellung von Netzplänen, in: Zeitschrift für Operations-Research, Band 20 (1976), S. B95-B114.

<sup>3</sup> Vgl. Silber, G., Das EDV-Organigramm und seine Ergänzung durch Anweisungs-, Informations- und Beratungsflußpläne, in: Zeitschrift für Organisation, Band 42 (1973), S. 393-401.

<sup>4</sup> Jope, G.R., Das Organisations-Handbuch, in: Harzburger Hefte, Band 13 (1970), S. 195-208.

<sup>5</sup> Jope, G.R., Parallelität der vertikalen Linienführung - Eine Systemdarstellung zur Anfertigung von Organisationsplänen, in: Harzburger Hefte, Band 12 (1969), S. 82 ff.

## 2. Problemstellung

Joepeschreibt die Anforderungen an einen Organisationsplan: "Die grafische Darstellung der Organisationspläne erfolgt nach dem System der "Parallelität der vertikalen Linienführung"<sup>1</sup>. Die Organisationspläne enthalten:

- alle genehmigten Planstellen (lt. Budget) innerhalb des Unternehmens,
- zu den Planstellen die entsprechenden Stelleninhaber (unberücksichtigt bleiben Aushilfen, Lehrlinge, Praktikanten und Studenten),
- die Bezeichnung der Planstelle, die das Aufgabengebiet und die Funktion des Stelleninhabers umreißt,
- die Kenntlichmachung von Planstellen gegebenenfalls als Teilzeitbeschäftigte (T) Lohnempfänger (L)."

Eine programmgesteuerte Erstellung von Organisationsplänen muß somit folgenden Bedingungen genügen:

- 1) jede Stelle muß gemäß der hierarchischen Gliederung richtig positioniert gedruckt werden,
- 2) die Verbindungen zwischen den Stellen müssen den vorgegebenen Kriterien entsprechen, die sich aus dem Darstellungsgegenstand ergeben. Mögliche Darstellungsarten sind z.B.:
  - Unterstellungsdiagramm
  - Weisungsflußplan
  - Informationsflußplan
  - Dienstleistungsflußplan usw.,
- 3) eine Kennzeichnung der Stellen muß nach verschiedenen Gesichtspunkten möglich sein,
- 4) die Daten für ein Organigramm sollen unsortiert gelesen werden können, um einen einfachen Änderungsdienst zu gewährleisten.

---

<sup>1</sup> Joepeschreibt, G.R., Parallelität der vertikalen Linienführung ..., a.a.O., S. 82 ff.

Die von Friedrich et al.<sup>1</sup> für Netzpläne aufgestellten Forderungen gelten hier sinngemäß:

- 5) hohe zeichnerische Qualität der grafischen Darstellung,
- 6) hinsichtlich Übersichtlichkeit und Gliederung sollen die Möglichkeiten manuell gezeichneter Pläne angestrebt werden,
- 7) leichte Lesbarkeit und Eindeutigkeit der organisatorischen Abhängigkeiten,
- 8) Möglichkeit der Einflußnahme des Anwenders auf Gestaltung und Struktur des Organigramms,
- 9) Unveränderlichkeit der Struktur der nicht betroffenen Teile bei Korrekturen.

Von den Autoren sind zwei Verfahren zur maschinellen Darstellung von Organisationsplänen entwickelt worden, die sich hinsichtlich der Anforderungen an die Rechner-Peripherie stark unterscheiden.

Das Verfahren 1<sup>2</sup> ist auf die Darstellung mit Hilfe eines Line-Printers zugeschnitten, während sich das Verfahren 2<sup>3</sup> eines Plotters bedient, bei dem die grafischen Gestaltungsmöglichkeiten wesentlich vielfältiger sind.

### 3 Programme zur maschinellen Erstellung von Organigrammen

#### 3.1 ORGAN.F4 - Organigramme auf einem Line-Printer

Das Programm ORGAN.F4 ist in Standard-FORTRAN-IV geschrieben und stellt keinerlei Anforderungen an Library-Functions. Die Portabilität dieses Programmes ist damit gewährleistet.

---

<sup>1</sup> Vgl. Friedrich, F.J. et al., Ein System zur grafischen Darstellung ..., a.a.O., S. B121.

<sup>2</sup> bearbeitet von Wolfgang Kessler (im Rahmen des im WS 1976/77 veranstalteten Seminars über Organisation).

<sup>3</sup> bearbeitet von G.F. Rueck.

### 3.1.1 Anforderungen an die einzugebenden Daten

Für die mit Hilfe dieses Programms darstellbaren Organisationen bestehen folgende Anforderungen an die einzugebenden Daten:

- 1) die Organisation wird mit einem Koordinatensystem überzogen, bei dem für jedes Element die eine Koordinate die Hierarchieebene und die andere Koordinate die Spalte im Organisationsplan darstellt.  
Bei der Codierung ist darauf zu achten, daß zwischen den Stellen der untersten Hierarchieebene einige unbesetzt bleiben, damit bei einer späteren Erweiterung um zusätzliche Stellen nicht alle neu codiert werden müssen. Auf diesen Punkt weist auch Silber hin<sup>1</sup>. Die dort definierte Strukturnummer entspricht dem hier verwendeten Datensatz.
- 2) Um entsprechend dem Darstellungsgegenstand Verbindungslinien zwischen Elementen ziehen zu können, muß zu jedem Element die vorgelagerte Stelle (Vorgesetzter, Weisungsgeber, Informationsquelle) mit ihren Koordinaten angegeben werden.  
Es muß eine Eindeutigkeit in den Beziehungen zwischen Vorgesetzten und Untergebenen herrschen. Ein Vorgesetzter kann mehrere Untergebene haben, aber ein Untergebener kann nur einen Vorgesetzten haben.  
Wird aufgrund einer fehlerhaften Eingabe ein Koordinatenpaar doppelt vergeben, wird nur das zufällig zuerst gelesene Element berücksichtigt.
- 3) Das vorliegende Programm berücksichtigt drei verschiedene Kennzeichnungsmöglichkeiten einer Stelle:
  - a) die Umrandung eines Organisationselementes wird je nach der Funktion des Elementes mit einem "I" bei einer Linien- bzw. mit einem "S" bei einer Stabsfunktion ausgeführt. Stäbe werden ihrer Funktion entsprechend der vorgesetzten Stelle beigeordnet.
  - b) Über eine DATA-Anweisung, die über eine 0,1-Variable aufgerufen werden kann, ist eine besondere Stellenkennzeichnung, wie sie Joep fordert, möglich. Hier

---

<sup>1</sup> Vgl. Silber, G., Das EDV-Organigramm..., a.a.O..



ist die Abkürzung ++MU++ zur Kenntlichmachung einer Mehrfachunterstellung gewählt worden.

- c) Schließlich kann mit bis zu vier Textzeilen von je 8 Zeichen die Stelle beschrieben werden. Damit läßt sich in das umrandete Feld einer Stelle z.B. die Funktionsbezeichnung und der Name des Stelleninhabers eintragen.

Ein vollständiger Datensatz für ein Element hat dann folgende Gestalt:

3 003 2 001 0 1 PERSONALWESEN MEYER LEITUNG

Hierbei sind:

- 3 003 die Koordinaten des Organisations-  
elementes,
- 2 001 die Koordinaten seines Vorgesetzten,
- 0 Merkmalsvariable 1 0 bewirkt ++MU++ (Mehrfachunter-  
stellung),  
1 bewirkt Leerzeile,
- 1 Merkmalsvariable 2 0 bewirkt Umrandung des Elementes mit  
SSS.. (Stabsstelle),  
1 bewirkt Umrandung des Elementes mit  
III.. (Linienstelle).

Danach folgt der alphanumerische Text, der in die Umrandung gedruckt werden soll.

### 3.1.2 Programmablauf

#### Einlesen der Daten

Beim Einlesen der Daten, die z.B. auf Lochkarten gespeichert sein können, wird eine Sortierung vorgenommen, so daß die Eingabereihenfolge beliebig ist<sup>1</sup>.

#### Bestimmung einer Druckzeile

Da in der Regel die Zahl der Organisationsspalten größer ist, als die Zahl der hierarchischen Ebenen, läuft eine Druckzeile über alle Ebenen. Bevor eine Zeile gedruckt werden kann, müssen folgende Daten bestimmt werden:

- 1) welche Elemente gehören zu dieser Organisationsspalte,
- 2) welche Merkmalsvariablen haben diese Elemente,

<sup>1</sup> Vgl. Punkt 4) in Kap. 2 Problemstellung.

- 3) haben diese Elemente ihren Vorgesetzten in einer vorherigen, in derselben oder in einer nachfolgenden Spalte,
- 4) ist ein Koordinatenpaar in der zu druckenden Zeile nicht durch ein Element besetzt, muß geklärt werden, ob horizontale oder vertikale Verbindungslinien von vorangegangenen oder noch folgenden Elementen die betrachtete Spalte kreuzen.

Wenn genau identifiziert ist, welches Element oder welche Verbindungslinie gedruckt werden soll, dann wird dieses Druckmuster einer gesonderten Datei entnommen. Dieses Verfahren bietet den Vorteil, daß bei anderen Gestaltungswünschen nur diese Datei geändert werden muß. Die Abbildung 1 zeigt, daß nur neun verschiedene Druckmuster nötig sind, um alle hier geforderten Erscheinungsformen von Organisationsplänen drucken zu können.

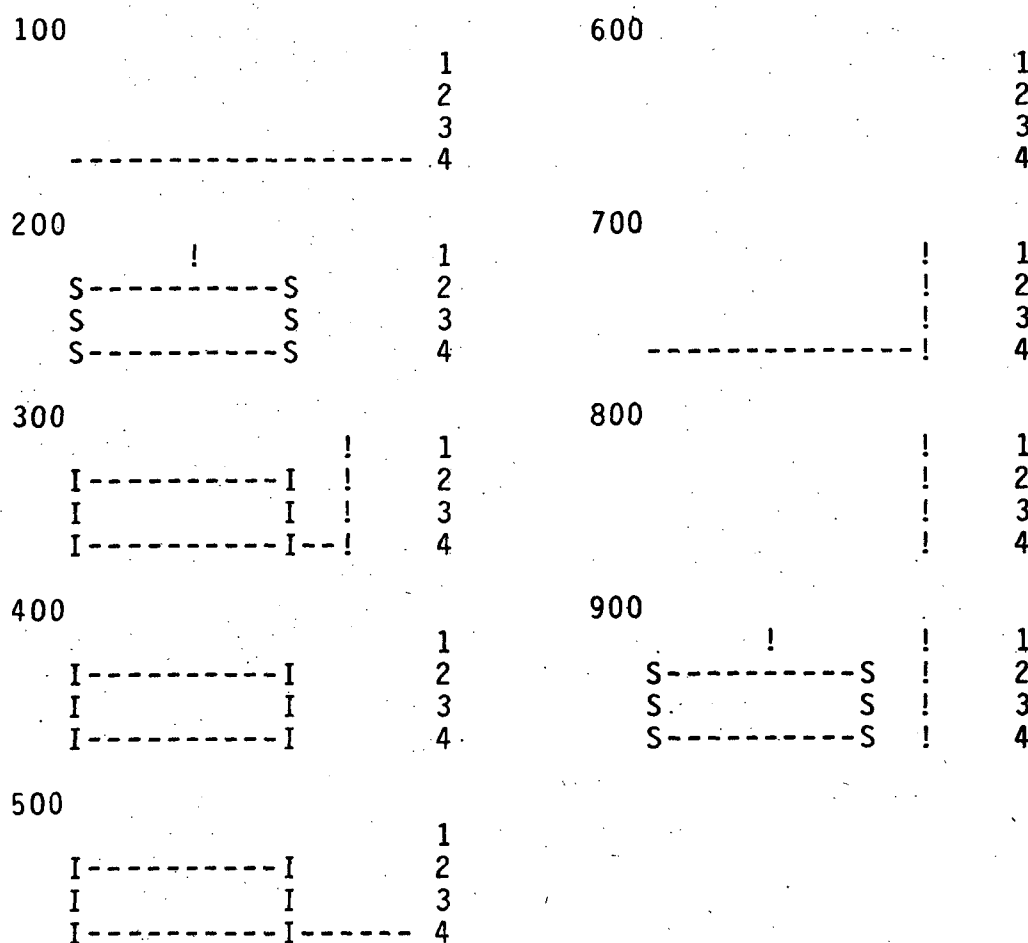


Abbildung 1 Druckmuster

Der Abbildung 1 kann man entnehmen, daß Verbindungslinien grundsätzlich nur nach rechts bzw. nach oben laufen. In einem kartesischen Koordinatensystem entspräche dies einer positiven X- bzw. Y-Richtung. Dies bedeutet, daß Verbindungslinien immer nur zu noch zu druckenden Elementen gezogen werden müssen. Die Formenvielfalt konnte dadurch erheblich reduziert werden, und es entfällt jeder Rückgriff auf bereits abgearbeitete Elemente. Die dadurch bedingte Einschränkung der grafischen Gestaltungsmöglichkeiten ist derart, daß eine Mittenausrichtung eines Vorgesetzten über allen ihm unterstellten Elementen nicht möglich ist. Die Abbildung 2 zeigt, daß eine linksbündige Ausrichtung erfolgt.

Spalte	3	2	1	Ebene
1	I-----I I C I I-----I	I-----I I B I I-----I	I-----I I A I I-----I	
2	I-----I I D I I-----I			
3	I-----I I F I I-----I	I-----I I E I I-----I		
4	I-----I I G I I-----I			

Abbildung 2 Beispiel

Die Tabelle 1 gibt an, welche Muster den einzelnen Druckfeldern in Abbildung 2 zugeordnet werden.

Ebene/Spalte	1/1	1/2	1/3	1/4	2/1	2/2	2/3	2/4	3/1	3/2	3/3	3/4
Muster Nr.	400	600	600	600	500	800	300	600	500	300	500	300

Tabelle 1 Druckmusterzuordnung

Die Muster 600 (Leerfeld) und 800 (senkrechte Linie) sind deshalb interessant, weil sie als "Dummy-Elemente" vom Programm eingefügt werden. Die Logik zur Auswahl der Druckmuster ist als Flußdiagramm in Abbildung 3 dargestellt.



### 3.1.3 Bedienung des Programms ORGAN.F4

Das Programm steht unter der Benutzernummer 6400,64111 des Rechenzentrums der Christian Albrecht Universität Kiel zur Verfügung.

Drei Parameter müssen im Programm vorgegeben werden:

- 1) Anzahl der Hierarchieebenen    KSMAX ; maximal    7
- 2) Anzahl der Spalten                IMAX    ; maximal 200
- 3) Anzahl der Textzeilen            ITZZ    ; maximal    4

Als Eingabe werden zwei Dateien benötigt:

- a) FORØ1.DAT enthält die Koordinaten der Elemente (Ebene,Spalte),  
die Koordinaten des Vorgesetzten (Ebene,Spalte),  
Mehrfachunterstellungen ; ja=Ø , nein=1 ,  
Stabsfunktion ; ja=Ø , nein=1 im  
FORMAT (2(1X,I1,1X,I3),2(1X,I1),1X,4A8),
- b) FORØ2.DAT Druckmusterdatei (vergleiche Abbildung 1).

Das fertige Organigramm steht nach Beendigung des Programm-  
laufes unter FORØ3.DAT zur Ausgabe auf einem Line-Printer  
zur Verfügung.

Das vollständige Programm befindet sich im Anhang C

### 3.2 ORGPOS , ORGPLT - Organigramme auf einem Plotter

Zur Erweiterung der Gestaltungsmöglichkeiten, die durch  
den beschränkten Zeichenvorrat eines Line-Printers begrenzt  
sind, wurde ein weiteres Verfahren entwickelt, das im Nach-  
folgenden beschrieben wird.

Zur Darstellung von Organisationsdiagrammen auf einem Plotter  
wurden auf der PDP-11/45 des Fachbereichs WiSo an der CAU  
Kiel zwei Programme erstellt. Beide sind in Standard-FORTRAN-IV  
abgefaßt und erlauben bis auf die Anpassung an die Plotter-  
Software weitgehend Portabilität. Eine Kurzbeschreibung der  
verwendeten Plotter-Routinen befindet sich im Anhang A.

### 3.2.1 Anforderungen an die einzugebenden Daten

Für die mit Hilfe dieser Programme darstellbaren Organisationen ergeben sich folgende Forderungen:

- 1) die Organisation wird in Ebenen (Levels) aufgeteilt und zwar so, daß der Vorgesetzte eines Elementes direkt in der über dem Element liegenden Ebene angesiedelt ist,
- 2) Stäbe liegen in derselben Hierarchieebene, in der sich auch deren Vorgesetzte befinden,
- 3) alle Elemente sind in der Darstellung im Diagramm gleichartig,
- 4) ein Stab kann Untergebene aber keine Stäbe haben, ein Element kann beliebig viele Untergebene und, soweit das Element seinerseits kein Stab ist, beliebig viele Stäbe haben,
- 5) mit Ausnahme der Elemente der obersten Hierarchieebene hat jedes Element genau einen Vorgesetzten.

Aus diesen Forderungen ergeben sich folgende Verbindungsformen unter Elementen:

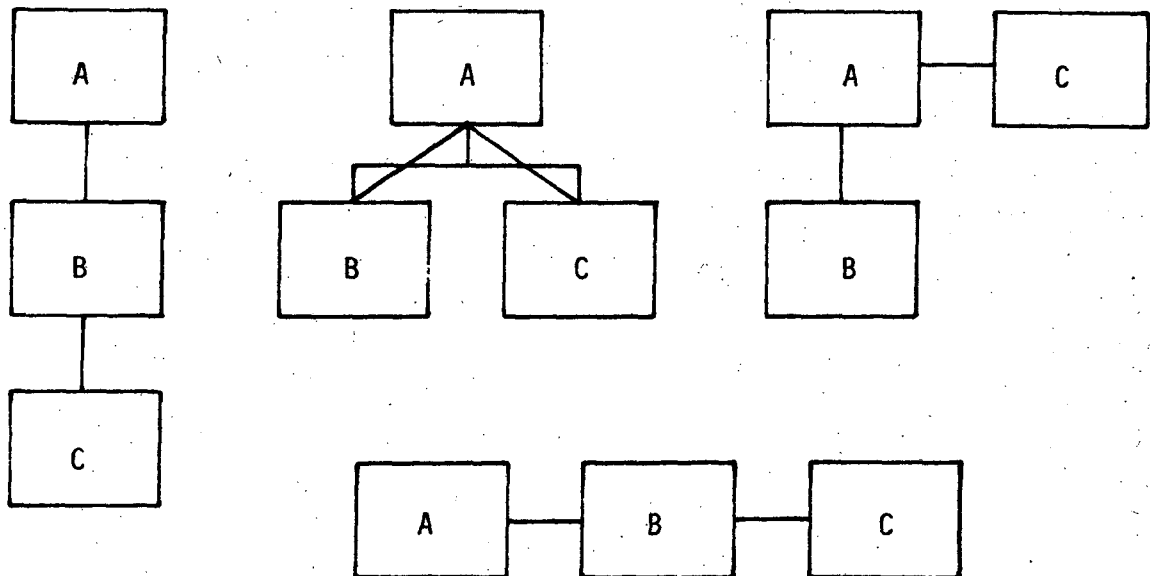


Abbildung 4 Verbindungsformen unter Elementen

Von dem Programmsystem müssen folgende Forderungen erfüllt werden:

- a) die Positionen der Elemente einer darzustellenden Organisation sollen aus den Beziehungsrelationen "A ist Vorgesetzter von B" bzw. "B ist Stab von A" bestimmt werden,
- b) die Eingabereihenfolge der Relationen soll beliebig sein,
- c) die Größe und die Beschriftung der Elemente sollen frei wählbar sein.

### 3.2.2 Programmablauf ORGPOS

Die Positionierung der Elemente wird mit Hilfe des Programms ORGPOS durchgeführt. Als Grundlage (Eingabe) hierzu stehen neben einigen Parametern zur Größe der Elemente im Diagramm und zur Darstellung der Verbindungslinien nur die Vorgesetztenrelationen zur Verfügung.

Die Positionierung wird in vier Phasen durchgeführt.<sup>1</sup>

#### 3.2.2.1 Einlesen der Daten

In der ersten Phase werden neben einigen Parametern die Beziehungsrelationen eingelesen. Dabei wird gleichzeitig eine Liste der 4-stelligen Elementbezeichnungen erzeugt (Vektor NAME) und zu jedem Element dessen Vorgesetzter oder Stabvorgesetzter vermerkt (Vektoren VORGES bzw. STABV). Die Reihenfolge des Auftretens bestimmt auch zum Teil die Position im Diagramm, wenn diese nicht durch die Relationen eindeutig festgelegt wird.

#### 3.2.2.2 Grobpositionierung der Elemente

In der zweiten Phase wird die Reihenfolge der Elemente innerhalb der einzelnen Hierarchiestufen festgelegt. Dazu werden zunächst alle Elemente, die weder einen Vorgesetzten haben, noch Stab sind, in der obersten Ebene angesiedelt. Zu jedem Element, das auf diese Weise gefunden wird, werden alle Stäbe gesucht und direkt hinter dem vorgesetzten Element eingetragen. Die Reihenfolge der Elemente wird im Vektor LEVEL festgehalten. Nach Abschluß der Bearbeitung

<sup>1</sup>Durch Verwendung von ALGOL, PL/1 oder PASCAL ließe sich dieses rekursive Problem wesentlich einfacher lösen. 3.2.2.2 und 3.2.2.3 wären in einem Schritt durchführbar.

der obersten Ebene werden die übrigen Elemente nach folgendem Verfahren placiert:

ausgehend von dem Vektor LEVEL werden zu jedem Element alle Untergebenen gesucht und deren Reihenfolge ebenfalls in LEVEL vermerkt, der auf diese Weise dynamisch wächst. Wird ein Untergebener eingetragen, so werden sofort dessen Stäbe gesucht und direkt dahinter eingetragen. Sind alle Elemente dieser Ebene gefunden, so wird deren Ende durch den Eintrag von  $\emptyset$  gekennzeichnet und die Positionierung der Elemente der nächsten Ebene begonnen.

Wird bei diesem Verfahren ein Element angetroffen, das nicht Vorgesetzter anderer Elemente ist, so wird, um die richtige Positionierung der untersten Ebene zu gewährleisten, ein Dummy-Element erzeugt und als Untergebener in LEVEL eingetragen. Das Verfahren bricht ab, wenn in einer Ebene nur Dummy-Elemente positioniert werden. Während des Verfahrens werden noch folgende Vektoren aufgebaut:

LEVADR enthält zu jeder Ebene den Index des ersten Elementes in LEVEL,

STATE gibt zu jedem Element an, ob es bereits positioniert wurde.

### 3.2.2.3 Feinpositionierung der Elemente

In der dritten Phase werden die Koordinaten der Elemente innerhalb des Diagramms festgelegt. Die Koordinaten werden in x und y angegeben, wobei x in Richtung der Ebenen läuft und y senkrecht auf den Ebenen steht (+x nach rechts; +y von oben nach unten).

Die y-Koordinate ergibt sich jeweils aus der Ebenennummer und den Parametern HOEHE (Höhe der Elemente) und ABSTAN (Abstand zweier Ebenen voneinander).

Die x-Koordinate für die Elemente der untersten Ebene ergibt sich aus den Parametern BREITE (Breite der Elemente) und RAND (Abstand der Elemente in einer Ebene voneinander). Aufbauend auf der untersten Ebene, werden die Elemente der darüberliegenden Ebene so positioniert, daß sie genau in der Mitte darüberliegen. Sind alle Elemente einer Ebene positioniert, so werden ausgehend von dieser die Elemente



der darüberliegenden Ebene positioniert. Die x- bzw. y-Koordinaten der Elemente werden in SPOS bzw. ZPOS festgehalten. Bei der Positionierung eines Elementes werden zu dessen Vorgesetzten die minimalen und die maximalen x-Koordinaten aller Unterbenen in MINPOS bzw. MAXPOS festgehalten, so daß der Wert für x durch einfaches Mitteln dieser Werte bestimmt werden kann.

#### 3.2.2.4 Koordinatenausgabe mit Berechnung der Verbindungen

Nach Abschluß der Feinpositionierung erfolgt in der 4. Phase die Ausgabe der Koordinaten der Elemente und der Verbindungslinien. Das Programm ermöglicht direkte (Abbildung 5) und rechtwinklig gezogene (Abbildung 6) Verbindungslinien.

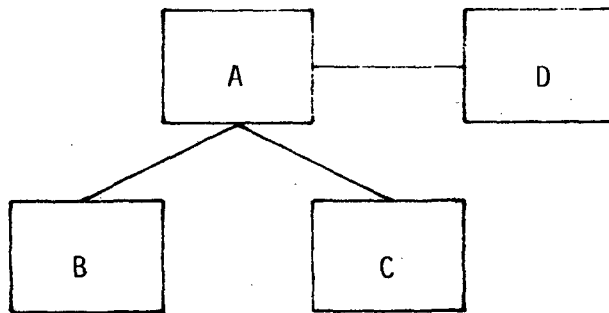


Abbildung 5 direkte Verbindungen

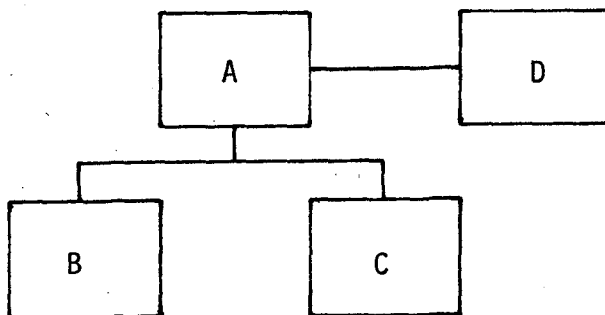


Abbildung 6 rechtwinklige Verbindungen

Die Verbindungen zu den Stäben (Element D in den Abb. 5 und 6) bleiben von der Wahl der Verbindungsart unberührt. Bei direkten Verbindungen werden Anfangs- und Endkoordinaten bestimmt und ausgegeben. Die rechtwinkligen Verbindungen werden in drei Teile aufgespalten:

- vom Vorgesetzten zum horizontalen Ast; die Berechnung erfolgt bei der Bearbeitung des Vorgesetzten,
- vom Unterbenen zum horizontalen Ast; die Berechnung erfolgt bei der Bearbeitung des Unterbenen,

- c) die Koordinatenberechnung des horizontalen Astes; dieser fehlt, wenn ein Vorgesetzter nur einen Untergebenen hat.

In der vorliegenden Version erfolgt die Ausgabe der Elementkoordinaten und die der Verbindungskoordinaten in eine gemeinsame Datei. Eine Trennung in zwei Dateien kann durch Veränderung der Kanalnummern IODSK und/oder IODSK2 erfolgen.

Durch den Parameter PAR1 kann ein Protokoll auf dem Zeilendrucker angefordert werden.

### 3.2.2.5 ORGPOS - Fehlermeldungen

#### ? UEBERLAUF DER LEVELLISTE

Die maximal mögliche Anzahl von Elementen in der Organisation wurde überschritten.

Abhilfe: Erweitern des Programms.

#### ? UEBERLAUF DER ELEMENTLISTE BEI EINFUEGEN DUMMY

Die maximal mögliche Anzahl von Elementen wurde überschritten. Dummy-Elemente werden für die logische Positionierung der Elemente benötigt, wenn diese keine Untergebenen haben und nicht auf der untersten Ebene liegen.

#### ? KEIN VORGESETZTER ZU ELEMENT name

Dieser Fehler darf in der vorliegenden Version nicht auftreten. Überprüfung der Relationen.

#### ? ELEMENT name NICHT POSITIONIERT

Dieser Fehler tritt auf, wenn ein Element, das nicht in der untersten Ebene liegt, keinen Dummy-Untergebenen erhalten hat. Im allgemeinen deutet dieser Fehler auf eine Falschpositionierung eines Untergebenen hin, die durch einen logischen Fehler in der Eingabe hervorgerufen wurde.

#### ? KEIN TOP-LEVEL ELEMENT

Für alle Elemente wurde ein Vorgesetzter definiert, d.h. die Organisation enthält mindestens eine Schleife.

#### ? UEBERLAUF DER ELEMENTLISTE

Beim Einlesen wurde die maximal mögliche Anzahl von Elementen überschritten.

#### ? ELEMENT name STEHT MIT SICH SELBST IN RELATION

Die eingelesene Karte enthält in ELM1 und ELM2 denselben

Wert. Die Karte wird ignoriert und das Einlesen fortgesetzt.

#### % UNZULAESSIGE RELATION "xx"

Die Karte beschreibt eine nicht definierte Relation. Sie wird ignoriert und das Einlesen fortgesetzt.

Das vollständige Programm ORGPOS befindet sich im Anhang C.

### 3.2.3 Programmablauf ORGPLT

Die Ausgabe des von ORGPOS berechneten Organisationsdiagramms erfolgt mit dem Programm ORGPLT, das sich in mehrere Abschnitte gliedert, die im folgenden einzeln beschrieben werden.

#### 3.2.3.1 Einlesen der Daten

Hier werden die Parameter für ORGPLT und die im Diagramm gewünschten Überschriften eingelesen und gespeichert. Dem Ergebnis von ORGPOS werden die dort spezifizierten Parameter und die Koordinaten der Elemente entnommen.

#### 3.2.3.2 Koordinatentransformation

Die in ORGPLT erzeugten Element- und Verbindungskordinaten liegen in einer unbeschränkten Zeichenebene. Um die Ausgabe großer Diagramme auch einem Plotter mit schmaler Papierbreite zu ermöglichen, wird das Diagramm, falls eine y-Koordinate die Papierbreite (Variable PSIZE) überschreitet, segmentiert. Durch die Koordinatentransformation wird das Diagramm in "Streifen" getrennt, die nacheinander ausgegeben werden<sup>1</sup>.

#### 3.2.3.3 Einlesen und Zwischenspeichern der Elementbeschriftungen

Die Elemente des Diagramms können beliebig beschriftet werden, wobei die Anzahl der Zeilen, die Schrifthöhe jeder Zeile und die Zeilenabstände durch Parameter vorgegeben werden. Die Zeilenabstände können auch vom Programm bestimmt werden. Die Zuordnung von Element und Beschriftung erfolgt mit Hilfe der vierstelligen Elementbezeichnung. Die Beschriftungen werden eingelesen und in einer Direktzugriffsdatei zwischengespeichert,

<sup>1</sup>An der verwendeten Rechananlage steht ein CALCOMP-563 Trommelplotter zur Verfügung, so daß eine Segmentierung in x-Richtung nicht betrachtet wurde.

wobei in IREC zu jedem Element vermerkt wird, wo sich der dazugehörige Text befindet.

#### 3.2.3.4 Plot der Elemente

Vor dem Plotten der Elemente werden die Überschriften ausgegeben. Beim Zeichnen der Elemente werden die Leerwege teilweise minimiert. Die Prozedur NEXT bestimmt dabei ausgehend von der Position des Zeichenstiftes das nächste anzufahrende Element und dessen Koordinaten. Nach dem Anfahren der Position wird der Rand gezeichnet und das Element beschriftet. Zu beachten ist, daß die positive y-Richtung im Diagramm der negativen y-Richtung auf dem Plotter entspricht. Aus diesem Grunde sind alle y-Koordinaten in den Plot-Aufrufen mit einem negativen Vorzeichen versehen.

#### 3.2.3.5 Einlesen und transformieren der Verbindungen

Nach dem Zeichnen der Elemente werden die Koordinaten der Verbindungslinien eingelesen und sofort entsprechend 3.2.3.2 transformiert. Dabei ist zu beachten, daß die Grenze zweier Segmente immer zwischen den Ebenen der Elemente liegt und daher die diese Grenze überschreitenden Verbindungen geteilt werden müssen. Für die Speicherung der Koordinaten werden die nicht mehr benötigten Vektoren der Elementkoordinaten benutzt. Da durch starke Segmentierung des Diagramms und/oder durch rechtwinklige Verbindungen die Anzahl der zu zeichnenden Linien sehr groß werden kann, werden bei einem Überlauf des Speichers alle bis dahin gespeicherten Linien gezeichnet und dann das Einlesen und Transformieren fortgesetzt.

#### 3.2.3.6 Plotten der Verbindungen

Auch hier erfolgt, wie in 3.2.3.4 beschrieben, eine teilweise Minimierung der Leerwege des Zeichenstiftes. Dabei wird der dem derzeitigen Standort des Zeichenstiftes am nächsten liegende Endpunkt der zu zeichnenden Linie angefahren und diese dann gezeichnet. Nachdem alle eingelesenen und transformierten Verbindungen gezeichnet sind, wird mit 3.2.3.5 fortgefahren.

### 3.2.3.7 Seitenmarkierungen

Nach der Ausgabe aller Verbindungen erhält das Diagramm Marken und Segmentbezeichnungen, die das Zusammensetzen der einzelnen Segmente einer Zeichnung erleichtern sollen.

### 3.2.3.8 ORGPLT - Fehlermeldungen

#### ? EOF BEIM LESEN POSITIONIERUNGSERGEBNIS

Das Positionierungsergebnis wurde von ORGPOS nicht vollständig erzeugt, i.a. nach Fehlern in ORGPOS.

#### ? EOF BEIM LESEN DER BESCHRIFTUNGEN

siehe Text 2 unten

#### ? UNZULAESSIGE ZEILENZAHL

siehe Text 1 unten

#### ? SCHRIFTGROESSE ZU KLEIN

Auf der Parameterkarte wurde SHOEHE kleiner als 0.07 angegeben.

#### ? SUMME SCHRIFTGROESSEN GROESSER ALS ELEMENTHOEHE

siehe Text 1

#### ? NEGATIVER ZEILENABSTAND

siehe Text 1

#### ? ELEMENT abcd NICHT POSITIONIERT

Folge eines Fehlers in ORGPOS

#### ? EOF BEIM LESEN PARAMETER FUER UEBERSCHRIFTEN

siehe Text 2

#### ? NICHT AUSREICHENDER RAUM FUER UEBERSCHRIFTEN

siehe Text 1

#### ? EOF BEIM LESEN UEBERSCHRIFTEN

siehe Text 2

#### ? SCHRIFT IN UEBERSCHRIFT ZU KLEIN

siehe Text 1

#### ? ZUVIELE ZEILEN IN UEBERSCHRIFT

siehe Text 1

#### ? MEHR ALS EINE SEITE FUER UEBERSCHRIFT

siehe Text 1

#### ? UEBERSCHRIFT BREITER ALS SEITE

siehe Text 1

-----Text 1 : Auf der Parameterkarte wurde ein unzulässiger Wert angegeben.

-----Text 2 : Folgefehler durch falschen Wert auf Parameterkarte.

Das vollständige Programm ORGPOS befindet sich im Anhang C.

### 3.2.4 Bedienung der Programme auf der PDP 11/45

Die Programme können unter DOS/BATCH betrieben werden. Sie stehen auf der Platte BWL Nr.5 unter der Benutzernummer {10,10}. Folgende Handlungen sind auszuführen:

- 1) Löschen der Files DK:INTER.FCE und DK:SCRATC.H

```
*R PIP
PIP VERSION .....
D DK:INTER.FCE/DE
D DK:SCRATC.H/DE
U↑C
.KI
*
```

- 2) Einlegen der Parameter- und Relationenkarten in den Kartenleser
- 3) Kartenleser: START drücken
- 4) Drucker: ON-LINE
- 5) Starten von ORGPOS

```
*RU ORGPOS
I350000000
*
```

- 6) Einlegen der Parameterkarten für ORGPLT in den Leser
- 7) Kartenleser: START
- 8) Vorbereiten Plotter
- 9) Starten ORGPLT

```
*RU ORGPLT
I350000000
*
```

- 10) Ende des Laufes; Drucker, Plotter und Kartenleser ausschalten.

#### 3.2.4.1 Bedienung des Programms ORGPOS

Das Programm steht unter der Nummer{10,10} auf der Platte BWL Nr.5.

- 1) Eingabe

Die Eingabe der Daten für ORGPOS erfolgt mit Hilfe von Lochkarten. Die Parameterkarten steuern die graphische Gestaltung des Organigramms, während die Relationenkarten die Unterstellungsverhältnisse der Elemente untereinander festlegen.

Parameterkarte:

Spalte	Parameter	Bemerk.	Bedeutung
01	PAR1	+	Ausgabe des Positionierungsprotokolls auf dem Drucker
02	PAR2	+	Ausgabe des Positionierungsergebnisses für ORGPLT (File DK:INTER.FCE)
03	PAR3		Art der Verbindungen: 0=direkt,1=rechtwinkl.
09-13	BREITE	*	Breite eines Elementes x-Richtung
14-18	HOEHE	*	Höhe eines Elementes y-Richtung
19-23	RAND	*	Abstand der Elemente x-Richtung
24-28	ABSTAN	*	Abstand der Ebenen y-Richtung
29-33	SRAND	*	Abstand zweier Segmente x-Richtung

Bemerkungen:

- + durch Angabe von 0 unterbleibt die Ausgabe, 1 bewirkt die Ausgabe.
- \* die Angabe erfolgt in cm im Format(F5.2).

Relationenkarte:

Spalte	Parameter	Bedeutung
01-04	ELM1	4-stelliger Bezeichner eines Elementes
06-07	R	Relation der Elemente zueinander
09-12	ELM2	4-stelliger Bezeichner eines Elementes

Folgende Relationen können definiert werden:

- R = "V " ELM1 ist Vorgesetzter von ELM2
- R = "U " ELM1 ist Untergebener von ELM2
- R = "VS" ELM1 ist Vorgesetzter des Stabes ELM2
- R = "US" ELM1 ist Stab von ELM2

2) Ausgabe

Die Ausgabe umfaßt je nach Wahl der Parameter ein Positionierungsprotokoll auf dem Drucker und/oder das Positionierungsergebnis für ORGPLT und gegebenenfalls Fehlermeldungen.

3.2.4.2 Bedienung des Programms ORGPLT

Die Parameter und die Beschriftungen der Elemente werden vom Kartenleser (CR:) eingelesen. Das Ergebnis der Positionierung aus dem Programm ORGPOS wird dem File DK:INTER.FCE entnommen.

Analog zu ORGPOS wird die Gestaltung der Ausgabe durch Parameter- und Datenkarten gesteuert.

Parameterkarte:

Spalte	Parameter	Bemerk.	Bedeutung
01-02	NLINES		Anzahl der Zeilen in der Beschriftung der Elemente, max. 16.
03	ISUPC		Ø : Die Zeilenabstände in den Elementbeschriftungen werden berechnet 1 : Die Zeilenabstände werden von Karten eingelesen
04	FARB1	*	Farbcode für Elemente
05	FARB2	*	Farbcode für Verbindungen
06	FARB3	*	Farbcode für Informationsverbindungen
07	FARB4	*	Farbcode für Seitenmarkierungen
08	SMT	*	Beschriftung der Seitenmarkierungen
09	SMP	*	Seitenmarkierungen
10-11	NZUE		Anzahl Zeilen in Überschrift, max. 6.

Bemerkungen:

\* noch nicht implementiert; Ø oder blank eingeben. Die vorliegende Version erzeugt immer Seitenmarkierungen und -beschriftungen. SMT=1 ; SMP=1

Überschrift:

Für jede der NZUE Zeilen in der Überschrift ist eine Karte mit der Anzahl der Zeichen, der Schrifthöhe, dem Abstand zur vorhergehenden Zeile bzw. zum Seitenanfang und dem Zeilenparameter im FORMAT(I3,2F5.2,I1) anzugeben. Danach folgt eine Karte mit dem Text der Zeile. Enthält die Überschrift mehr als 80 Zeichen, so sind entsprechend weitere Karten anzugeben.

Als Zeilenparameter können folgende Werte angegeben werden:

- Ø Zeile wird linksbündig ausgegeben
- 1 Zeile erscheint rechtsbündig
- 2 Zeile wird über der Diagrammitte ausgerichtet

Schrifthöhen in Elementen:

Für jede der NLINES Zeilen der Elementbeschriftung ist die gewünschte Schrifthöhe im FORMAT(16F5.2) vorzugeben. Die Schrifthöhen sollten dabei ganzzahlige Vielfache von 0.07 sein.



Zeilenabstände (nur, wenn ISUPC=1):

Für jede Zeile der Elementbeschriftung ist der Abstand zur darüberliegenden Zeile im FORMAT(16F5.2) und bei der ersten Zeile der Abstand zum Rand anzugeben.

Enthält die Parameterkarte den Wert ISUPC=0, so berechnet das Programm die Zeilenabstände in Abhängigkeit von den Schrifthöhen und dem freien Platz.

Beschriftung der Elemente:

Für jedes Element im Organisationsdiagramm ist ein Satz von NLINES+1 Karten zu erstellen, wobei die erste Karte den Bezeichner des Elementes in den Spalten 01-04 und die restlichen NLINES Karten den Text für jeweils eine Zeile der Beschriftung enthalten. Die Anzahl der Zeichen je Zeile ist auf 80 Zeichen beschränkt.

Die Anzahl  $N_i$  der in Zeile  $i$  ausgegebenen Zeichen berechnet sich aus

$$N_i = \min\{NK_i, (0.8 * BREITE) / ((SH_i * 6.0) / 7.0)\},$$

wobei  $NK_i$  = Anzahl der Zeichen in Zeile  $i$  (auf der Karte)  
 $BREITE$  = Breite des Elementes in cm  
 $SH_i$  = Schrifthöhe in Zeile  $i$  sind.

- Friedrich, F.J.,  
G.Ploch,  
R.Pohlenz,  
Ein System zur grafischen Darstellung  
von Vorgangsknoten-Netzplänen mit Hilfe  
elektronisch gesteuerter Zeichenmaschi-  
nen, in: Zeitschrift für Operations-  
Research, Band 19(1975), S.B120, nach  
Händler, W., Bericht Nr.2, Fachtagung  
Computer Graphics der Gesellschaft für  
Informatik, Berlin, 19. bis 21. 10. 1971,  
S. 421.
- Joep, G.R.  
Das Organisations-Handbuch, in: Harz-  
burger Hefte, Band 13(1970), S. 195-208.
- Joep, G.R.  
Parallelität der vertikalen Linienführung-  
Eine Systemdarstellung zur Anfertigung  
von Organisationsplänen, in: Harzburger  
Hefte, Band 12(1969), S. 82 ff.
- Theis, W.,  
Trommer, W.,  
Ein Verfahren zur übersichtlichen maschi-  
nellen Darstellung von Netzplänen, in:  
Zeitschrift für Operations-Research,  
Band 20(1976), S. B95-B114.
- Silber, G.  
Das EDV-Organigramm und seine Ergänzung  
durch Anweisungs-, Informations- und  
Beratungsflußpläne, in: Zeitschrift für  
Organisation, Band 42(1973), S. 393-401.

## Anhang A

### Kurzbeschreibung der benutzten Systemprozeduren

#### 1) ASSIGN

Durch einen Anruf ASSIGN(i,spec,l) wird ein Kanal i mit einem Ein-Ausgabegerät (KB:=Konsole, LP:=Drucker, CR:=Kartenleser, DK:=Platte) oder einer Datei verbunden. l ist die Länge des Textes.

#### 2) DATE

DATE liefert das Tagesdatum als Text im Format tt-mmm-jj. (Tag-Monat-Jahr)

#### 3) TIME

TIME liefert die Tageszeit als Text im Format hh:mm:ss. (Stunden:Minuten:Sekunden)

#### 4) ABS

ABS liefert den Absolutbetrag einer Variablen.

#### 5) INT

Konvertierung REAL in INTEGER durch Abschneiden (ohne Rundung).

#### 6) FLOAT

Konvertierung INTEGER in REAL.

#### 7) PLOTST

PLOTST initialisiert die Plotterausgabe. Parameter sind die Zeicheneinheit und die Bezeichnung der Einheit.

#### 8) PLOTND

PLOTND schließt die Plotterausgabe ab.

#### 9) PLOT(x,y,j)

Bewegung des Zeichenstiftes vom derzeitigen Standort zum Punkt mit den Koordinaten (x,y).

j =  $\pm 2$  Bewegung mit gesenkten Stift

j =  $\pm 3$  Bewegung mit gehobenem Stift

Bei negativem j wird der angefahrene Punkt (x,y) zum neuen Ursprung der Zeichnung, d.h. (x,y) = (0,0).

10) SYMBOL

Ausgabe eines alphanumerischen Textes SYMBOL(X,Y,H,T,A,J)

X,Y Punkt an dem der Text beginnen soll, d.h. Koordinaten der linken unteren Ecke des ersten Zeichens. Der Punkt (X,Y) wird vor Ausgabe des Textes angefahren. Werden als Koordinaten des Punktes (999.0,999.0) angegeben, so beginnt die Ausgabe am Standort des Zeichenstiftes.

H Schrifthöhe

T Text, der ausgegeben werden soll

A Winkel zur x-Achse in Grad, in dem der Text geplottet werden soll.

J Anzahl der auszugebenden Zeichen von T.

11) NUMBER

Ausgabe einer Zahl NUMBER(X,Y,H,V,A,J)

X,Y,H,A siehe SYMBOL

V auszugebender Wert

J Anzahl der Stellen hinter dem Dezimalpunkt.

Bei negativem J wird auch die Ausgabe des Punktes unterdrückt.

12) DEFINE FILE statement

Definition eines Files mit Direktzugriff durch

DEFINE FILE IU(N,M,U,IVAR).

IU FORTRAN-Kanalnummer

N Anzahl der Records

M Länge des Records in Worten

U Der Bezeichner U zeigt an, daß der File binär geschrieben und gelesen wird.

IVAR "assoziierte Variable" - findet hier keine Verwendung

Anhang B

- B1 Lineprinter-Output einer Beispielorganisation  
erzeugt von ORGAN.F4
- B2 Plotter-Output einer Beispielorganisation  
erzeugt von ORGPOS und ORGPLT

```

I-----I      I-----I      I-----I
I      I      I      I      I      I
I LAGER 1 I      I SPARTE 1 I      I LEITUNG I
I      I      I      I      I      I
I      I      I      I      I      I
I-----I-----I-----I

I-----I      !
I      I      !
I LAGER 2 I      !
I      I      !
I      I      !
I-----I-----!

I-----I      I-----I      !
I      I      I      I      !
I FERTI- I      I SPARTE 2 I      !
I GUNG 1 I      I      I      !
I      I      I      I      !
I      I      I      I      !
I-----I-----I-----!

I-----I      !      S-----S
I      I      !      S      S
I FERTI- I      !      S STAR S
I GUNG 2 I      !      S SPARTE 2 S
I      I      !      S      S
I      I      !      S      S
I-----I-----!      S-----S

I-----I
I      I
I HILFS- I
I LAGER I
I      I
I **MU** I
I-----I

```

Abbildung B1 Lineprinter-Output einer Beispielorganisation  
erzeugt von ORGAN.F4

# AUFBAUORGANISATION DER DATENVERARBEITUNG IN EINEM KLEINEN UNTERNEHMEN

UMSATZ 50-100 MIO. DM/JAHR

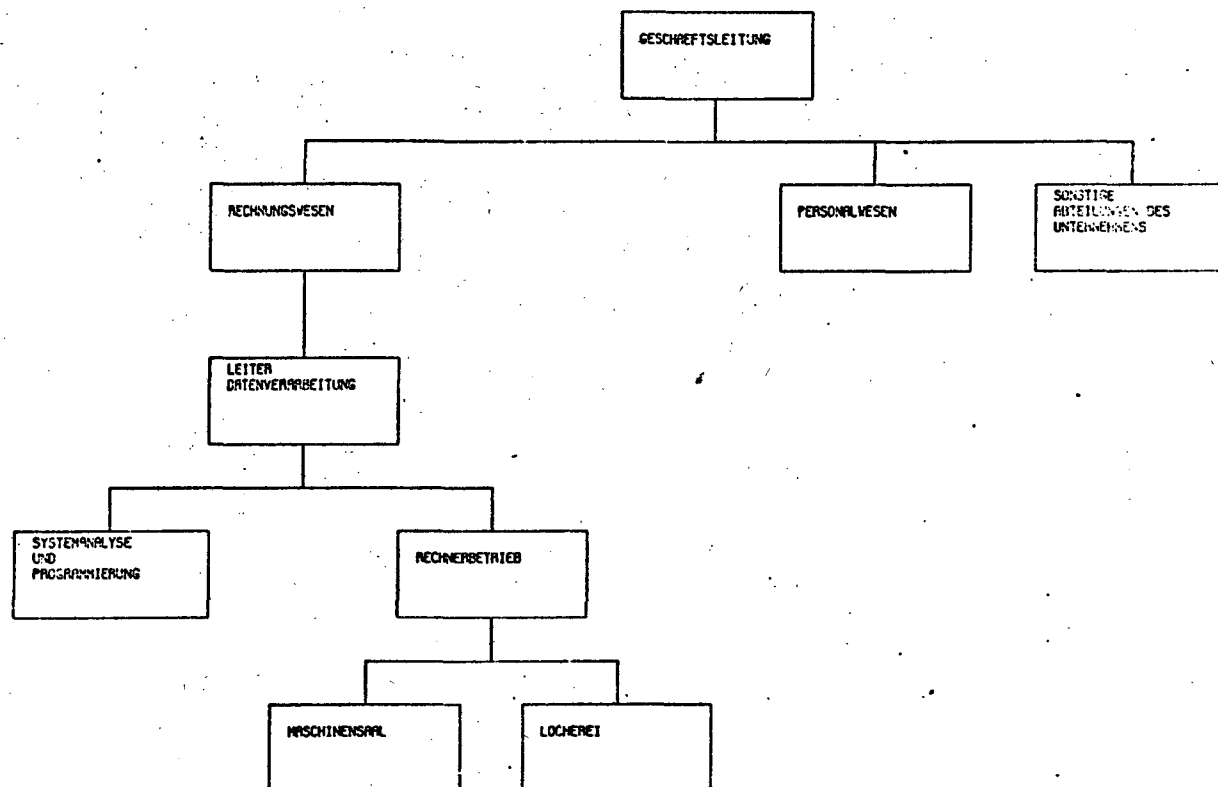


Abbildung B2 Plotter Output einer Beispielorganisation  
erzeugt von ORGPOS, ORGPLT

C            Vollständige Haupt- und Unterprogramme

C 1                    ORGAN.F4



```

CC
CC PROGRAMM ORGAN.F4
CC ZUR ERSTELLUNG VON BAUMSTRUKTUREN UND ORGANIGRAMMEN.
CC DIE ELEMENTE EINES HIERARCHISCHEN SYSTEMS WERDEN GEMAEß
CC DEN VORGEgebenEN UNTERSTELLUNGSVERHAELTNISSEN UNTEREIN-
CC ANDER VERBUNDEN.
CC
CC VORZUGEBENDE PARAMETER:
CC ANZAHL DER HIERARCHIEEBENEN      KSMAX;  MAXIMALE ANZAHL= 7
CC ANZAHL DER SPALTEN                IMAX ;   MAXIMALE ANZAHL=200
CC ANZAHL DER TEXTZEILEN            ITZZ ;   MAXIMALE ANZAHL= 4
CC
CC INPUT-FILES:
CC FOR01.DAT:  KOORDINATEN DER ELEMENTE, UNTERSTELLUNGEN,
CC              MEKMALE, TEXTE
CC      VARIABLEN: KOORDINATEN DES ELEMENTES      (EBENE,SPALTE)
CC                  KOORDINATEN DES VORGESETZTEN (EBENE,SPALTE)
CC                  MEHRFACHUNTERSTELLUNGEN      JA=0 ; NEIN=1
CC                  STABSFUNKTION, BEIORDNUNG JA=0 ; NEIN=1
CC      FORMAT(2(1X,I1,1X,I3),2(1X,I1),1X,4A8)
CC
CC FOR02.DAT:  DRUCKMUSTERDATEI
CC
CC OUTPUT-FILE:
CC FOR03.DAT
CC
CC AUTOR :    W.  KESSLER
CC
CC -----
CC
CC
CC      DOUBLE PRECISION WORT,AWORT
CC      DIMENSION KSPALT(7,200),KZEIL(7,200),KSPAVID(7,200)
CC      1,KZEIV0(7,200)
CC      DIMENSION KMERFA(7,200),KSTAB(7,200),AWORT(4,7,200)
CC      1,WORT(4)
CC      DIMENSION KSTEU(7,200),KSUCH(7),PRINT(5,9,7),AWORT3(7)
CC      DIMENSION AWORT4(7)
CC      DATA DA1/'      '//,DA2/' **M'//,DA3/'U** '//
CC
CC VORGABE DER PARAMETER:
CC
CC      IMAX=20
CC      KSMAX=4
CC      ITZZ=4
CC.....
CC      ILZZ=5+ITZZ
CC
CC EINLESEN DER DATEN:
CC
CC      REWIND 1
CC      DO 40 J1=1,1000

```

```
      READ(1,430,END=50)ISPALT,IZEIL,ISPAVO,IZEIVO,IMERFA,ISTAB,
1(WORT(IW),IW=1,ITZZ)
      DO 30 I1=1,IMAX
      IF(IZEIL.NE.I1)      GOTO 30
      DO 20 K=KSMAX,1,-1
      IF(ISPALT.NE.K)      GOTO 20
      KSPALT(K,I1)=ISPALT
      KZEIL(K,I1)=IZEIL
      KSPAVO(K,I1)=ISPAVO
      KZEIVO(K,I1)=IZEIVO
      KMERFA(K,I1)=IMERFA
      KSTAB(K,I1)=ISTAB
      KSTEU(K,I1)=1
      DO 10 M=1,ITZZ
      MW=M+3
10      AWORT(MW,K,I1)=WORT(M)
      GOTO 40
20      CONTINUE
30      CONTINUE
40      CONTINUE
CC
CC      BESTIMMUNG DES DRUCKMUSTERS:
CC
50      I=0
      DO 420 I1=1,IMAX
      I=I+1
60      KSUMM=0
      DO 70 K=KSMAX,1,-1
      KSUMM=KSUMM+KSTEU(K,I1)
70      IF(KSUMM.EQ.0)      GOTO 420
      DO 290 K=KSMAX,1,-1
      KMAX=K
      IF(KSTEU(K,I1).EQ.0)      GOTO 130
      IF(KSPAVO(K,I1).NE.9)      GOTO 80
      KSUCH(K)=400
      GOTO 290
80      IF(KZEIVO(K,I1).EQ.KZEIL(K,I1))      GOTO 110
      IF(KSTAB(K,I1).EQ.1)      GOTO 90
      GOTO 260
90      IF(KSPAVO(K,I1).NE.(K-1))      GOTO 100
      KSUCH(K)=300
      GOTO 290
100     KSUCH(K)=500
      GOTO 290
110     IF(KSTAB(K,I1).EQ.1)      GOTO 120
      GOTO 260
120     KSUCH(K)=500
      GOTO 290
      IFIX=0
130     DO 140 I30=KMAX,KSMAX,1
      IF(KSTEU(I30,I1).NE.1)      GOTO 140
      IFIX=I30
      GOTO 150
```

```
140  CONTINUE
      GOTO 190
150  IF(KZEIVO(IFIX,I1).NE.I1)      GOTO 160
      KSUCH(K)=100
      GOTO 290
160  IF(KSPAVID(IFIX,I1).NE.K)      GOTO 170
      GOTO 190
170  IF(KSPAVID(IFIX,I1).GE.K)      GOTO 180
      GOTO 190
180  IF(KSPAVID(IFIX,I1).NE.(K-1))  GOTO 190
      KSUCH(K)=700
      GOTO 290
190  IF(K.EQ.1)      GOTO 220
      DO 210 I5=I1,IMAX
      DO 210 MK=KSMAX,K,-1
      IF((KSPAVID(MK,I5).EQ.(K-1)).AND.(KZEIVO(MK,I5).LT.I1)) GOTO 230
200  IF((KSPAVID(MK,I5).LE.(K-2)).AND.(KZEIL(MK,I5).EQ.I1)
      1.AND.(KZEIVO(MK,I5).LT.I1))  GOTO 250
210  CONTINUE
220  KSUCH(K)=600
      GOTO 290
230  IF(KZEIL(MK,I5).NE.I1)      GOTO 240
      KSUCH(K)=700
      GOTO 290
240  KSUCH(K)=800
      GOTO 290
250  KSUCH(K)=100
      GOTO 290
260  DO 280 I5=I1,IMAX
      DO 280 MK=KSMAX,K,-1
      IF(KSPAVID(MK,I5).EQ.(K-1))  GOTO 270
      GOTO 280
270  KSUCH(K)=900
      GOTO 290
280  CONTINUE
      KSUCH(K)=200
290  CONTINUE
CC
CC  EINLESEN DES DRUCKMUSTERS:
CC
      DO 350 K=KSMAX,1,-1
      REWIND 2
300  READ(2,440) KSUCH2
      IF(KSUCH2.EQ.KSUCH(K))      GOTO 320
      DO 310 IZ=1,4
310  READ(2,450) ALEER
      GOTO 300
320  DO 340 IZ=1,ILZZ
      IF((IZ.GE.4).AND.(IZ.LT.ILZZ)) GOTO 330
      READ(2,460)(PRINT(N,IZ,K),N=1,5)
      GOTO 340
330  REREAD 470,(PRINT(N,IZ,K),N=1,4)
340  CONTINUE
```

```
350  CONTINUE
      DO 380 K=KSMAX,1,-1
      IF(KSTEU(K,I1).EQ.0)      GOTO 360
      IF(KMERFA(K,I1).NE.1)    GOTO 370
360  AWORT3(K)=DA1
      AWORT4(K)=DA1
      GOTO 380
370  AWORT3(K)=DA2
      AWORT4(K)=DA3
380  CONTINUE
CC-
CC  ZUSAMMENSTELLUNG EINER DRUCKZEILE:
CC
      DO 410 IZ=1,ILZZ
      IF((IZ.GE.4).AND.(IZ.LT.(ILZZ-1))) GOTO 390
      IF(IZ.EQ.(ILZZ-1)) GOTO 400
      WRITE(3,480)((PRINT(N,IZ,K),N=1,5),K=KSMAX,1,-1)
      GOTO 410
390  WRITE(3,490)(PRINT(1,IZ,K),AWORT(IZ,K,I1),(PRINT(N,IZ,K),
      1N=2,4),K=KSMAX,1,-1)
      GOTO 410
400  WRITE(3,500)I,(PRINT(1,IZ,K),AWORT3(K),AWORT4(K),(PRINT(N,
      1IZ,K),N=2,4),K=KSMAX,1,-1)
410  CONTINUE
420  CONTINUE
      STOP
430  FORMAT(2(1X,I1,1X,I3),2(1X,I1),1X,4A8)
440  FORMAT(1X,I4)
450  FORMAT(A4)
460  FORMAT(5X,4A4,A2)
470  FORMAT(5X,A2,8X,A2,A4,A2)
480  FORMAT(1H*,3X,7(4A4,A2))
490  FORMAT(1H*,3X,7(A2,A8,A2,A4,A2))
500  FORMAT(1H*,I3,7(A2,2A4,A2,A4,A2))
      END
```

C 2

ORGPOS

INPUT

RANGE

PROGRAMM

ORGPOS:

BERECHNUNG DER POSITIONEN VON ELEMENTEN EINES ORGANISATIONS DIAGRAMMS  
AUS DEN UNTERSTELLUNGSRELATIONEN

G.F. RUECK

14-JUN-78

```

0001      IMPLICIT INTEGER*2 (A-Z)
0002      REAL FL,POS,FAKT,ADDX,ADDY,HGHPOS,HG2POS,SRAND
0003      REAL BREITE,RAND,HOEHE,ABSTAN,DUMMY,BLANK,BR2,H2
0004      REAL XO,YO,X1,Y1
0005      REAL DATUM(3),ZEIT(3)

0006      DIMENSION VORGES(500),STABV(500),STATE(500)
0007      REAL NAME,SPOS(500),ZPOS(500),MINPOS(500),MAXPOS(500)
0008      DIMENSION LEVADR(100),LEVEL(600)

0009      COMMON IOCDR,IOLPT,IODSK,MAXIDX,MAXLEV,MAXELM,NELM,NELM2,NAME(500)

0010      DATA DUMMY/'DMY.'/, BLANK/' ',ITOP/-1/

      LOESCHEN DER FELDER

0011      CALL DATE(DATUM)
0012      CALL TIME(ZEIT)
0013      MAXIDX=600
0014      MAXLEV=100
0015      MAXELM=500
0016      IOCDR=1
0017      IOLPT=2
0018      IODSK=3
0019      IODSK2=3
0020      IDX=0
0021      NELM=0
0022      NELM2=0
0023      DO 10 I=1,MAXELM
0024          VORGES(I)=0
0025          STABV(I)=0
0026          STATE(I)=0
0027          NAME(I)=BLANK
0028          SPOS(I)=0.0
0029          ZPOS(I)=0.0
0030          MINPOS(I)=-1.0
0031          MAXPOS(I)=-1.0
0032 10      CONTINUE
0033      DO 20 I=1,MAXLEV
0034          LEVADR(I)=0
0035 20      CONTINUE
0036      DO 30 I=1,MAXIDX
0037          LEVEL(I)=0

```

```

0038 30    CONTINUE
C
0039      CALL ASSIGN(IOC'DR,'CR:',3)
0040      CALL ASSIGN(IOLPT,'LP:',3)
0041      CALL ASSIGN(IODSK,'DK:INTER.FCE',12)
0042      READ(IOC'DR,1000) PAR1,PAR2,PAR3,PAR4,PAR5,BREITE,HOEHE,RAND,ABSTAN,SRAND
0043 1000    FORMAT(5I1,3X,5F6.2)
C
C      EINLESEN DER VERHAELTNISRELATIONEN
C
0044 100    CALL INPUT(NODE1,NODE2,ITYP)
0045      IF (ITYP.EQ.0) GOTO 110
0047      IF (ITYP.EQ.1) VORGES(NODE1)=NODE2
0049      IF (ITYP.EQ.2) STABV(NODE1)=NODE2
0051      GOTO 100
C
C      SUCHE UND POSITIONIERE TOP-LEVEL ELEMENTE
C
0052 110    DO 120 I=1,NELM
0053      IF ((VORGES(I).NE.0).OR.(STABV(I).NE.0)) GOTO 120
0055      VORGES(I)=-1
0056      IDX=IDX+1
0057      LEVEL(IDX)=I
0058      STATE(I)=IDX
0059      DO 115 J=1,NELM
0060      IF (STABV(J).NE.I) GOTO 115
0062      IDX=IDX+1
0063      LEVEL(IDX)=J
0064      STATE(J)=IDX
0065 115    CONTINUE
0066 120    CONTINUE
C
C      POSITIONIEREN DER ELEMENTE INNERHALB DER LEVELS (TOP DOWN)
C
0067 200    IF (IDX.EQ.0) GOTO 905
0069      IDX=IDX+1
0070      NXTIDX=IDX+1
0071      LEVEL(IDX)=0
0072      LEVADR(2)=NXTIDX
0073      LEVADR(1)=1
0074      LEVNR=1
0075      NELM2=NELM
C
0076 210    GCOUNT=0
0077      IDX=LEVADR(LEVNR)
0078      NXTIDX=LEVADR(LEVNR+1)
0079 220    NODE=LEVEL(IDX)
0080 230    COUNT=0
0081      IF (NODE.EQ.0) GOTO 270
0083      DO 250 I=1,NELM
0084      IF (VORGES(I).NE.NODE) GOTO 250
0086      IF (RANGE(NXTIDX,1,MAXIDX).NE.0) GOTO 901
0088      STATE(I)=NXTIDX
0089      LEVEL(NXTIDX)=I

```

```

0090      NXTIDX=NXTIDX+1
0091      COUNT=COUNT+1
0092      GCOUNT=GCOUNT+1
0093      DO 240 J=1,NELM
0094      IF (STABV(J),NE.1) GOTO 240
0096      IF (RANGE(NXTIDX,1,MAXIDX).NE.0) GOTO 901
0098      LEVEL(NXTIDX)=J
0099      STATE(J)=NXTIDX
0100      NXTIDX=NXTIDX+1
0101      COUNT=COUNT+1
0102      GCOUNT=GCOUNT+1
0103 240   CONTINUE
0104 250   CONTINUE
0105      IF (COUNT.NE.0) GOTO 260
0107      NELM2=NELM2+1
0108      IF (RANGE(NELM2,1,MAXELM).NE.0) GOTO 902
0110      NAME(NELM2)=DUMMY
0111      IF (RANGE(NXTIDX,1,MAXIDX).NE.0) GOTO 901
0113      LEVEL(NXTIDX)=NELM2
0114      NXTIDX=NXTIDX+1
0115      VORGES(NELM2)=NODE
0116 260   IDX=IDX+1
0117      GOTO 220
0118 270   IF (GCOUNT.LE.0) GOTO 300
0120 280   LEVNR=LEVNR+1
0121      LEVADR(LEVNR+1)=NXTIDX+1
0122      LEVEL(NXTIDX)=0
C  PRUEFEN OB ALLE ELEMENTE POSITIONIERT SIND
0123      DO 290 NE=1,NELM
0124      IF (STATE(NE).LE.0) GOTO 210
0126 290   CONTINUE
C
C  POSITIONIEREN DER ELEMENTE (BOTTOM UP)
C
0127 300   FAKT=HOEHE+ABSTAN
0128      ADDX=RAND / 2.0
0129      ADDY=ABSTAN/2.0
0130      NLEVEL=LEVNR
0131      POS=0.0
0132      IDX=LEVADR(LEVNR)
0133 320   NODE=LEVEL(IDX)
0134      IF (NODE.EQ.0) GOTO 350
0136      SPOS(NODE)=POS+ADDX
0137      FL=LEVNR-1.0
0138      ZPOS(NODE)=FL*FAKT+ADDY
0139      VOR=VORGES(NODE)
0140      IF (VOR.EQ.0) GOTO 330
0142      MAXPOS(VOR)=POS
0143      IF (MINPOS(VOR).LT.0.0) MINPOS(VOR)=POS
0145 330   POS=BREITE+RAND+POS
0146      IDX=IDX+1
0147      GOTO 320
0148 350   LEVNR=LEVNR-1
0149      IF (LEVNR.LE.0) GOTO 380

```



```

0151      IDZ=LEVADR(LEVNR)
0152 360   NODE=LEVEL(IDX)
0153      IF (NODE.EQ.0) GOTO 350
0155      IF ((MINPOS(NODE).LT.0.0).OR.(MAXPOS(NODE).LT.MINPOS(NODE)))
1GOTO 904
0157      POS=MINPOS(NODE)+(MAXPOS(NODE)-MINPOS(NODE))/2.0
0158      SPOS(NODE)=POS+ADDX
0159      FL=LEVNR-1.0
0160      ZPOS(NODE)=FL*FAKT+ADDY
0161      VOR=VORGES(NODE)
0162      IF (VOR.EQ.0) GOTO 370
0164      MAXPOS(VOR)=POS
0165      IF (MINPOS(VOR).LT.0.0) MINPOS(VOR)=POS
0167 370   IDZ=IDZ+1
0168      GOTO 360
0169 380   HGHPOS=0.0
0170      HG2POS=0.0
0171      DO 390 NODE=1,NELM
0172      IF (SPOS(NODE).GT.HGHPOS) HGHPOS=SPOS(NODE)
0174      IF (SPOS(NODE).LT.0.0) GOTO 904
0176      IF (ZPOS(NODE).GT.HG2POS) HG2POS=ZPOS(NODE)
0178      IF (ZPOS(NODE).LT.0.0) GOTO 904
0180 390   CONTINUE
0181      HGHPOS=HGHPOS+BREITE+ADDX
0182      HG2POS=HG2POS+HOEHE+ADDY

C
C C
C C C
C C C
0183 500   IF (PAR1.EQ.0) GOTO 600
0185      WRITE(IOLPT,5000) MAXIDZ,MAXLEV,MAXELM,NELM,NELM2,NLEVEL,NXTIDZ
0186 5000   FORMAT('O*** ERGEBNIS DER POSITIONIERUNG ***'/
1'OKAPAZITAETEN          IDZ=',15/
2'                        LEV=',15/
3'                        ELM=',15/
4'OANZAHL DEFINIERTER ELEMENTE      NELM=',15/
5'OANZAHL ELEMENTE EINSCHL. DUMMIES  NELM2=',15/
6'OANZAHL LEVELS                  NLEVEL=',15/
7'HOECHSTER INDEX IN LEVEL-FELD     NXTIDZ=',15)
0187 5001   FORMAT(' ',20I6)
0188      WRITE(IOLPT,5002)
0189      WRITE(IOLPT,5001) (LEVADR(I),I=1,NLEVEL)
0190 5002   FORMAT('OSTARTADRESSEN FUER LEVELLISTEN :')
0191      WRITE(IOLPT,5003)
0192      WRITE(IOLPT,5001) (LEVEL(I),I=1,NXTIDZ)
0193 5003   FORMAT('OLEVELLISTEN:')
0194      WRITE(IOLPT,5004)
0195 5004   FORMAT('OLISTE DER ELEMENTE :')
1'ONUM NAME VORG STBV STAT  X-POS. / Y-POS.  MINPOS',
1'      MAXPOS')
0196      DO 510 I=1,NELM2
0197      WRITE(IOLPT,5005) I,NAME(I),VORGES(I),STBV(I),STATE(I),
1      SPOS(I),ZPOS(I),MINPOS(I),MAXPOS(I)
0198 5005   FORMAT(' ',13,2X,A4,3I6,4F10.3)
0199 510   CONTINUE

```

```

0200      WRITE(IOLPT,5101) BREITE,HOEHE,RAND,ABSTAN
0201      WRITE(IOLPT,5102) HGHPOS,FAKT,SRAND
0202  5101  FORMAT('OPPOSITIONIERUNGSPARAMETER :'/
      1' BREITE DER ELEMENTE (X-RI.)          BREITE=',F12.4/
      2' HOEHE DER ELEMENTE (Y-RI.)          HOEHE=',F12.4/
      3' ABSTAND DER ELEMENTE (X-RI.)          RAND=',F12.4/
      4' ABSTAND DER ELEMENTE (Y-RI.)          ABSTAN=',F12.4/
0203  5102  FORMAT(
      5' GROSSTE X-KOORDINATE          HGHPOS=',F12.4/
      6' FAKTOR FUER BERECHNUNG Y-KOORDINATEN  FAKT=',F12.4/
      7' SEITENRAND                    SRAND=',F12.4/

```

C  
C TESTPLOT  
C

```

0204  600  BR2=BREITE/2.0
0205      H2=HOEHE/2.0
0206      IF (PAR2.EQ.0) GOTO 700
0208      CALL PLOTST(0.01,'CM')
0209      CALL SYMBOL(2.0,1.0,0.42,'ORGA2 - TESTPLOT',90.0,16)
0210      CALL WHERE(X0,Y0,X1)
0211      CALL SYMBOL(X0,Y0+2.0,0.35,DATUM,90.0,9)
0212      CALL WHERE(X0,Y0,X1)
0213      CALL SYMBOL(X0,Y0+2.0,0.35,ZEIT,90.0,8)
0214      CALL PLOT(5.0,30.0,-3)
0215      DO 650 NL=1,NLEVEL
0216      IDX=LEVADR(NL)
0217  605  I=LEVEL(IDX)
0218      IF (I.EQ.0) GOTO 650
0220      IF (PAR3.NE.0) GOTO 606
0222      IF (NAME(I).EQ.DUMMY) GOTO 640
0224  606  X0=SPOS(I)
0225      Y0=-ZPOS(I)
0226      X1=X0+BREITE
0227      Y1=Y0-HOEHE
0228      CALL PLOT(X0,Y0,3)
0229      CALL PLOT(X1,Y0,2)
0230      CALL PLOT(X1,Y1,2)
0231      CALL PLOT(X0,Y1,2)
0232      CALL PLOT(X0,Y0,2)
0233      CALL SYMBOL(X0+0.1,Y0-0.6,0.28,NAME(I),0.0,4)
0234      VOR=VORGES(I)
0235      IF (VOR.LE.0) GOTO 630
0237      X0=X0+BR2
0238      X1=SPOS(VOR)+BR2
0239      IF (PAR5.NE.0) GOTO 610
0241      Y1=-ZPOS(VOR)-HOEHE
0242      CALL PLOT(X0,Y0,3)
0243      CALL PLOT(X1,Y1,2)
0244      GOTO 630
0245  610  CALL PLOT(X0,Y0,3)
0246      CALL PLOT(X0,Y0+ABSTAN/2.0,2)
0247      DO 620 IJ=1,NELM
0248      IF (VORGES(IJ).EQ.1) GOTO 625
0250  620  CONTINUE

```

```

0251      GOTO 630
0252 625   Y0=Y0-HOEHE
0253      Y1=Y0-ABSTAN/2.0
0254      CALL PLOT(X0,Y0,3)
0255      CALL PLOT(X0,Y1,2)
0256      X1=MAXPOS(I)-MINPOS(I)
0257      IF (X1.LT.0.01) GOTO 630
0259      CALL PLOT(X0-X1/2.0,Y1,3)
0260      CALL PLOT(X0+X1/2.0,Y1,2)
0261 630   ST=SIARV(I)
0262      IF (ST.EQ.0) GOTO 640
0264      X0=SPOS(I)
0265      Y0=-ZPOS(I)-H2
0266      J=LEVEL(IDX-1)
0267      X1=SPOS(J)+BREITE
0268      Y1=-ZPOS(J)-H2
0269      CALL PLOT(X0,Y0,3)
0270      CALL PLOT(X1,Y1,2)
0271 640   IDX=IDX+1
0272      GOTO 605
0273 650   CONTINUE
0274      CALL PLOTND

```

C  
C  
C

# AUSGABE DER ELEMENTPOSITIONEN

```

0275 700   IF (PAR4.EQ.0) GOTO 800
0277      WRITE(IODSK,7003)
0278 7003   FORMAT(' PLOT CREATED BY ORGA2')
0279      WRITE(IODSK,7001) BREITE,HOEHE,RAND,ABSTAN,HGHPOS,HG2POS,SRAND
0280 7001   FORMAT(8F10.4)
0281      WRITE(IODSK,7002) NELM
0282 7002   FORMAT(I4)
0283      DO 701 I=1,NELM
0284      WRITE(IODSK,7004) NAME(I),SPOS(I),ZPOS(I)
0285 7004   FORMAT(1X,A4,2F10.4)
0286 701    CONTINUE
0287      DO 750 NL=1,NLEVEL
0288      IDX=LEVADR(NL)
0289 705    I=LEVEL(IDX)
0290      IF (I.EQ.0) GOTO 750
0292      IF (PAR3.NE.0) GOTO 707
0294      IF (NAME(I).EQ.DUMMY) GOTO 740
0296 707    X0=SPOS(I)
0297      Y0=ZPOS(I)
0298      X1=X0+BREITE
0299      Y1=Y0+HOEHE
0300      VOR=VORGES(I)
0301      IF (VOR.LE.0) GOTO 730
0303      X0=X0+BR2
0304      X1=SPOS(VOR)+BR2
0305      IF (PAR5.NE.0) GOTO 710
0307      Y1=ZPOS(VOR)+HOEHE
0308      WRITE (IODSK2,7010) X0,Y0,X1,Y1
0309 7010   FORMAT(4F8.4)

```

```

0310      GOTO 730
0311 710   Y1=Y0+ABSTAN/2.0
0312      WRITE(IODSK2,7010) X0,Y0,X0,Y1
0313      DO 720 IJ=1,NFILM
0314      IF (VJRGES(IJ).EQ.1) GOTO 725
0316 720   CONTINUE
0317      GOTO 730
0318 725   Y0=Y0+HDEHE
0319      Y1=Y0+ABSTAN/2.0
0320      WRITE(IODSK2,7010) X0,Y0,X0,Y1
0321      X1=MAXPOS(1)-MINPOS(1)
0322      IF (X1.LT.0.01) GOTO 730
0324      X0=X0-X1/2.0
0325      X1=X0+X1
0326      WRITE(IODSK2,7010) X0,Y1,X1,Y1
0327 730   ST=STABV(1)
0328      IF (ST.EQ.0) GOTO 740
0330      X0=SPOS(1)
0331      Y0=ZPOS(1)+H2
0332      J=LEVEL(IDX-1)
0333      X1=SPOS(J)+BREITE
0334      Y1=ZPOS(J)+H2
0335      WRITE(IODSK2,7010) X0,Y0,X1,Y1
0336 740   IDX=IDX+1
0337      GOTO 705
0338 750   CONTINUE
0339 800   STOP

```

C

C FEHLERMELDUNGEN

C

```

0340 901   WRITE(IOLPT,9011)
0341 9011  FORMAT('0? UEBERLAUF DER LEVEL-LISTE')
0342      STOP
0343 902   WRITE(IOLPT,9021)
0344 9021  FORMAT('0? UEBERLAUF DER ELEMENTLISTE BEI EINFUEGEN DUMMY!')
0345      STOP
0346 903   WRITE(IOLPT,9031) NAME(NODE)
0347 9031  FORMAT('0? KEIN VORGESETZTER ZU ELEMENT ',A4)
0348      STOP
0349 904   WRITE(IOLPT,9041) NAME(NODE)
0350 9041  FORMAT('0? ELEMENT ',A4,' NICHT POSITIONIERT')
0351      STOP
0352 905   WRITE(IOLPT,9051)
0353 9051  FORMAT('0? KEIN TOP-LEVEL ELEMENT')
0354 999   STOP
0355      END

```

```

0001      SUBROUTINE INPUT(NODE1,NODE2,ITYP)

```

```

0002      IMPLICIT INTEGER(A-Z)

```

```

0003      DIMENSION RCHAR(4)

```

```

0004      REAL NAME(500),N1,N2

```

```

0005      COMMON IOCDR,IOLPT,IODSK,MAXIDX,MAXLEV,MAXELM,NELM,NELM2,NAME(500)

```

```

0006      DATA RCHAR/'U ','US','V ','VS'/

```

```

0007      DATA MAXTYP/4/

```

```

0008      ITYP=0

```

```

0009      READ(IOCDR,1000,END=50) N1,R,N2

```

```

0010 1000  FORMAT(A4,1X,A2,1X,A4)

```

```

0011      ITYP=-1

```

```

0012      DO 4 I=1,MAXTYP

```

```

0013      IF (R.EQ.RCHAR(I)) ITYP=I

```

```

0015 4      CONTINUE

```

```

0016      IF (ITYP.EQ.-1) GOTO 102

```

```

0018      NODE1=0

```

```

0019      NODE2=0

```

```

0020      IF (N1.EQ.N2) GOTO 103

```

```

0022      IF (NELM.GT.0) GOTO 5

```

```

0024      NAME(1)=N1

```

```

0025      NAME(2)=N2

```

```

0026      NELM=2

```

```

0027 5      DO 10 I=1,NELM

```

```

0028      IF (NAME(I).EQ.N1) NODE1=I

```

```

0030      IF (NAME(I).EQ.N2) NODE2=I

```

```

0032 10      CONTINUE

```

```

0033      IF (NODE1.NE.0) GOTO 20

```

```

0035      NELM=NELM+1

```

```

0036      IF (RANGE(NELM,1,MAXELM).NE.0) GOTO 101

```

```

0038      NAME(NELM)=N1

```

```

0039      NODE1=NELM

```

```

0040 20      IF (NODE2.NE.0) GOTO 30

```

```

0042      NELM=NELM+1

```

```

0043      IF (RANGE(NELM,1,MAXELM).NE.0) GOTO 101

```

```

0045      NAME(NELM)=N2

```

```

0046      NODE2=NELM

```

```

0047 30      IF (ITYP.LE.2) RETURN

```

```

0049      ITYP=ITYP-2

```

```

0050      N=NODE1

```

```

0051      NODE1=NODE2

```

```

0052      NODE2=N

```

```

0053 50      RETURN

```

```

0054 101     WRITE(IOLPT,1001)

```

```

0055 1001    FORMAT('0? UEBERLAUF DER ELEMENTLISTE')

```

FORTRAN IV VOIC-03  
CJRH=31K, UIC={1,7}

MON 17-DEC-79 13:49:54

PAGE 002

NLO:,LP0:=DT1:{11,13}URGA3

```
0056      STOP
0057 102   WRITE(10LPT,1002) R
0058 1002  FORMAT('0% UNZULAESSIGE RELATION  ',A2,'')
0059      RETURN
0060 103   WRITE(10LPT,1003) N1
0061 1003  FORMAT('0? ELEMENT ',A4,' STEHT MIT SICH SELBST IN RELATION')
0062      ITYP=-1
0063      RETURN
0064      END
```

```
0001      INTEGER FUNCTION RANGE(VAL,LOW,HIGH)
```

```
0002      IMPLICIT INTEGER (A-Z)
0003      I=0
0004      IF (VAL.LT.LOW) I=-1
0006      IF (VAL.GT.HIGH) I=1
0008      RANGE=I
0009      RETURN
0010      END
```

C 3

ORGPLT

DISTNC

KTRANS

LTRANS

NEXT



PROGRAMM

ORGPLT

AUSGABE DES POSITIONIERUNGSERGEBNISSES AUS ORGPOS AUF DEM PLOTTER

GFR 30.05.78

```

0001  DIMENSION XPDS(500),YPDS(500),XNAME(500),IST(500)
0002  DIMENSION SHOEHE(20),SABST(20),NCHAR(20),TEXT(20)
0003  DIMENSION DATUM(3),ZEIT(3)
0004  DIMENSION UEBSCH(300),ZH(6),ZA(6),JSTART(6),JPARAM(6),JCHAR(6)
0005  COMMON /PAGING/ NPAGE,PSIZE,XMAX,SRAND,PBOUND(20)
0006  DATA VERSN/2,15/
0007  DATA MAXUEC/75/,MAXZUE/6/
0008  DATA MAXCHR/80/,MAXEL4/500/,MAXPAG/20/,MAXLIN/20/
0009  DATA IOINP1/1/,IOINP2/2/,IOINP3/1/,IOLPT/4/,IOSCR/5/
0010  CALL DATE(DATUM)
0011  CALL TIME(ZEIT)
0012  PSIZE=25.0
0013  CALL ASSIGN(IOINP1,'DK:INTER.FCE',12)
0014  CALL ASSIGN(IOINP2,'CR:',3)
0015  CALL ASSIGN(IOLPT,'KB:',3)
0016  CALL ASSIGN(IOSCR,'DK:SCRATC.H',11)
0017  READ(IOINP1,4201) TEXT
0018  CALL PLOTST(0.01,'CM')
0019  CALL PLOT(2.0,-30.0,-3)
0020  CALL SYMBOL(0.0,2.0,0.42,'ORGPLT VERSION ',90.0,15)
0021  CALL NUMBER(999.0,999.0,0.42,VERSN,90.0,2)
0022  CALL SYMBOL(0.0,10.0,0.42,DATUM,90.0,9)
0023  CALL SYMBOL(0.0,15.0,0.42,ZEIT,90.0,8)
0024  CALL SYMBOL(1.3,2.0,0.35,TEXT,90.0,80)
0025  CALL PLOT(3.0,30.0,-3)
0026  CALL PLOT(0.0,-1.0,-3)
0027  EINLESEN DER PARAMETERKARTE UND DER UEBERSCHRIFTEN
0027  READ(IOINP2,3501,END=902) NLines,ISUPC,IFARB1,IFARB2,IFARB3,
1IFARB4,ISMT,ISMP,NZUE
0028  READ(IOINP1,3001) BREITE,HOEHE,RAND,ABSTAN,XMAX,YMAX,SRAND
0029  IE=0
0030  SYY=0.0
0031  SUMUE=0.0
0032  IF (NZUE.GT.MAXZUE) GOTO 913
0034  IF (NZUE.LE.0) GOTO 300
0036  DO 150 I=1,NZUE
0037  READ(IOINP2,4401,END=908) J,ZH(1),ZA(1),JPARAM(1)

```

```

0038 4401 FORMAT(13,2F5.2,11)
0039 IF (ZH(I).LT.0.07.OR.ZA(I).LT.0.07) GOTO 912
0041 IA=IE+1
0042 JCHAR(1)=J
0043 J1=J/4
0044 IF (MOD(J,4).EQ.0) J1=J1-1
0046 IE=IA+J1
0047 IF (IE.GT.MAXUEC) GOTO 909
0049 JSTART(1)=IA
0050 SUMUE=SUMUE+ZH(I)+ZA(I)
0051 READ(IOINP2,4201,END=911) (UEBSCH(J),J=IA,IE)
0052 150 CONTINUE
0053 HA=HOEHE+ABSTAN
0054 SY=SUMUE/HA
0055 IY=SY
0056 SY=IY
0057 SYY=SY*HA
0058 IF(SYY.LT.SUMUE) SYY=SYY+HA

```

C  
 C EINLESEN DER POSITIONIERUNGSPARAMETER UND POSITIONEN DER ELEMENTE  
 C

```

0060 3001 FORMAT(7F10.4)
0061 300 READ(IOINP1,3002) NELM
0062 3002 FORMAT(14)
0063 DO 310 I=1,NELM
0064 READ(IOINP1,3003,END=901) XNAME(I),XPOS(I),YPOS(I)
0065 3003 FORMAT(1X,A4,2F10.4)
0066 310 CONTINUE

```

C  
 C KOORDINATENTRANSFORMATIONEN  
 C

```

0067 FAKT=HOEHE+ABSTAN
0068 PSIZE=PSIZE/FAKT
0069 IS=PSIZE
0070 YMAX=YMAX+SYY
0071 PSIZE=IS*FAKT
0072 IF (SYY.GT.PSIZE) GOTO 914
0074 Y=YMAX/PSIZE
0075 NPAGE=INT(Y)+1
0076 YP=ABS(Y*YMAX-PSIZE)
0077 IF (YP.LE.0.01) NPAGE=NPAGE-1
0079 DO 320 I=1,NPAGE
0080 PROUND(I)=PSIZE*FLOAT(I)
0081 320 CONTINUE
0082 DO 330 I=1,NELM
0083 YP=YPOS(I)+SYY
0084 CALL KTRANS(XPOS(I),YP,X,Y)
0085 XPOS(I)=X
0086 YPOS(I)=Y
0087 330 CONTINUE

```

C  
 C EINLESEN DER BESCHRIFTUNGEN UND SPEICHERN IN EINEM  
 C RANDOM-ACCESS-FILE  
 C

```

0088 3501 FORMAT(I2,7I1,I2)
0089 350 IF ((NLINES.LE.0).OR.(NLINES.GT.MAXLIN)) GOTO 903
0091 READ(IOINP2,3502,END=902)(SHOEHE(I),I=1,NLINES)
0092 3502 FORMAT(16F5,2)
0093 SUM=0.0
0094 SHMIN=999.9
0095 DO 360 I=1,NLINES
0096 S=SHOEHE(I)
0097 IF (S.LT.0.07) GOTO 904
0099 SUM=SUM+S
0100 IF (S.LT.SHMIN) SHMIN=S
0102 360 CONTINUE
0103 IF (SUM.GT.HOEHE) GOTO 905
0105 IF (ISUPC.NE.0) GOTO 380
0107 READ(IOINP2,3502,END=902) (SABST(I),I=1,NLINES)
0108 SUM2=0.0
0109 DO 370 I=1,NLINES
0110 S=SABST(I)
0111 IF (S.LE.0.0) GOTO 906
0113 SUM2=SUM2+S
0114 370 CONTINUE
0115 IF ((SUM+SUM2).GE.HOEHE) GOTO 905
0117 GOTO 400
C BERECHNUNG DER ZEILENABSTAENDE
0118 380 SUM2=HOEHE-SUM-0.25*SHMIN
0119 DO 390 I=1,NLINES
0120 SABST(I)=(SHOEHE(I)/SUM)*SUM2
0121 390 CONTINUE
C BERECHNEN DER SCHRITTANFANGSKOORDINATEN
0122 400 S=0.0
0123 DO 410 I=1,NLINES
0124 S=S+SHOEHE(I)+SABST(I)
0125 SABST(I)=S
0126 410 CONTINUE
0127 MX=0
0128 BRTE2=0.8*BREITE
0129 DO 420 I=1,NLINES
0130 S = BRTE2 / ((SHOEHE(I) *6.0) / 7.0)
0131 N = S
0132 IF (N.GT.MAXCHR) N=MAXCHR
0134 IF (N.GT.MX) MX=N
0136 NCHAR(I)=N
0137 420 CONTINUE
0138 MX=MX/4
0139 IF (MOD(MX,4).NE.0) MX=MX+1
0141 N = NELM * NLINES
C EINLESEN UND SPEICHERN DER TEXTE
0142 MX4=MX*4
0143 DEFINE FILE IOSCR(N,MX4,U,IASSV2)
0144 DO 440 K=1,NELM
0145 READ(IOINP2,4201,END=902) XNM
0146 DO 425 KK=1,NELM
0147 J=KK
0148 IF (XNM.EQ.XNAME(KK)) GOTO 426

```

PLOTTEN DER UEBERSCHRIFTEN

PLOTTEN DER ELEMENTE

33

C EINLESEN UND TRANSFORMIEREN DER LINKS

```

0201 500 CALL PLOT(0.0,0.0,3)
0202 510   IDX=0
0203       IFLAG=0
0204 520 READ(IQINP3,5201,END=560) X0,Y0,X1,Y1
0205 5201  FORMAT(4F8.4)
0206       Y0=Y0+SYX
0207       Y1=Y1+SYX
0208       CALL BTRANS(X0,Y0,X1,Y1,XS,YS,J)
0209       IF (J.EQ.0) GOTO 550
0211 530   IDX=IDX+1
0212       IFLAG=1
0213       IF (IDX.GE.MAXELM) GOTO 560
0215       WRITE(IQPLT,5203) XS,YS
0216 5203  FORMAT(' SEITENGRENZE BEI ',2F12.4)
0217       XPOS(IDX)=X0
0218       YPOS(IDX)=Y0
0219       IDX=IDX+1
0220       XPOS(IDX)=XS-0.0001
0221       YPOS(IDX)=YS-0.0001
0222 540   IDX=IDX+1
0223       IFLAG=2
0224       IF (IDX.GE.MAXELM) GOTO 560
0226       XPOS(IDX)=XS+0.0001
0227       YPOS(IDX)=YS+0.0001
0228       IDX=IDX+1
0229       XPOS(IDX)=X1
0230       YPOS(IDX)=Y1
0231       IFLAG=0
0232       GOTO 520
0233 550   IDX=IDX+1
0234       IFLAG=3
0235       IF (IDX.GE.MAXELM) GOTO 560
0237       XPOS(IDX)=X0
0238       YPOS(IDX)=Y0
0239       IDX=IDX+1
0240       XPOS(IDX)=X1
0241       YPOS(IDX)=Y1
0242       IFLAG=0
0243       GOTO 520

```

```

0244 560   IF (IFLAG.NE.0) IDX=IDX-1
0246       DO 570 I=1,IDX
0247       CALL KTRANS(XPOS(I),YPOS(I),X,Y)
0248       XPOS(I)=X
0249       YPOS(I)=Y
0250 570   CONTINUE

```

C PLOTEN DER DEFINIERTEN LINKS

```

0251       DO 580 I=1,IDX
0252       IST(I)=0
0253 580   CONTINUE

```

```

0254 590 K=-1
0255      X0=0
0256      Y0=0
0257      IDX2=IDX/2
0258      DO 600 I=1,IDX2
0259        CALL NEXT(IST,XPOS,YPOS,X0,Y0,IDX,K)
0260        K2=K/2
0261        KM=MOD(K,2)
0262        IF (KM.EQ.0) KM=-1
0264        I2=KM+K
0265        I1=K
0266        X0=XPOS(I2)
0267        Y0=YPOS(I2)
0268 610 IST(11)=-1
0269      IST(12)=-1
0270      CALL PLOT(XPOS(11),-YPOS(11),3)
0271      CALL PLOT(XPOS(12),-YPOS(12),2)
0272 600 CONTINUE
0273 620 IDX=0
0274      IF (IFLAG.EQ.0) GOTO 630
0276      GOTO (530,540,550),IFLAG
0277 630 PXS=XMAX+SRAND
0278      NP=NPAGE
  
```

cccc

PLOTTEN DER SEITENMARKIERUNGEN

```

0279 640 XST=NP*PXS-SRAND
0280      CALL PLOT(XST,1.0,3)
0281      CALL PLOT(XST,0.0,2)
0282      CALL PLOT(XST-1.0,0.0,2)
0283      XP=NP
0284      CALL SYMBOL(XST-3.0,0.2,0.21,'END OF PAGE ',0.0,12)
0285      CALL NUMBER(999.0,999.0,0.21,XP,0.0,-1)
0286      XST=XST-XMAX
0287      CALL PLOT(XST,1.0,3)
0288      CALL PLOT(XST,0.0,2)
0289      CALL PLOT(XST+1.0,0.0,2)
0290      CALL SYMBOL(XST+0.5,0.2,0.21,'START OF PAGE ',0.0,14)
0291      CALL NUMBER(999.0,999.0,0.21,XP,0.0,-1)
0292      NP=NP-1
0293      IF (NP.GT.0) GOTO 640
0295      PS2=-PSIZE-0.42
0296      DO 650 I=1,NPAGE
0297        XST=(I-1)*PXS
0298        XP=I
0299        CALL PLOT(XST,-PSIZE-1.0,3)
0300        CALL PLOT(XST,-PSIZE,2)
0301        CALL PLOT(XST+1.0,-PSIZE,2)
0302        CALL SYMBOL(XST+0.5,PS2,0.21,'START OF PAGE ',0.0,14)
0303        CALL NUMBER(999.0,999.0,0.21,XP,0.0,-1)
0304        XST=XST+XMAX
0305        CALL SYMBOL(XST-3.0,PS2,0.21,'END OF PAGE ',0.0,12)
0306        CALL NUMBER(999.0,999.0,0.21,XP,0.0,-1)
0307        CALL PLOT(XST-1.0,-PSIZE,3)
  
```

```

0308      CALL PLOT(XST,-PSIZE,2)
0309      CALL PLOT(XST,-PSIZE-1.0,2)
0310 650    CONTINUE
0311      XXX=NPAGE*PXS+SRAND+1.0
0312      CALL SYMBOL(XXX,-PSIZE,0.21,'END OF PLOT AT ',90.0,16)
0313      CALL TIME(ZEIT)
0314      CALL SYMBOL(999.0,999.0,0.21,ZEIT,90.0,8)
0315      CALL PLOTND
0316      STOP
  
```

C

C

FEHLERMELDUNGEN

C

```

0317 901    WRITE(IOLPT,9011)
0318 9011   FORMAT('0? EOF BEIM LESEN POSITIONIERUNGSERGEBNIS')
0319      STOP 1
0320 902    WRITE(IOLPT,9021)
0321 9021   FORMAT('0? EOF BEIM LESEN BESCHRIFTUNGEN')
0322      STOP 2
0323 903    WRITE(IOLPT,9031)
0324 9031   FORMAT('0? UNZULAESSIGE ZEILENANZAHL')
0325      STOP 3
0326 904    WRITE(IOLPT,9041)
0327 9041   FORMAT('0? SCHRIFTGROESSE ZU KLEIN')
0328      STOP 4
0329 905    WRITE(IOLPT,9051)
0330 9051   FORMAT('0? SUMME SCHRIFTGROESSEN GROESSER ALS ELEMENTHOEHE')
0331      STOP 5
0332 906    WRITE(IOLPT,9061)
0333 9061   FORMAT('0? NEGATIVE ZEILENABSTAND')
0334      STOP 6
0335 907    WRITE(IOLPT,9071) XNM
0336 9071   FORMAT('0? ELEMENT ',A4,' NICHT POSITIONIERT')
0337      STOP 7
0338 908    WRITE(IOLPT,9081)
0339 9081   FORMAT('0? EOF BEIM LESEN PARAMETER FUER UEBERSCHRIFTEN')
0340      STOP 10
0341 909    WRITE(IOLPT,9091)
0342 9091   FORMAT('0? NICHT AUSREICHENDER RAUM FUER UEBERSCHRIFTEN')
0343      STOP 11
0344 911    WRITE(IOLPT,9111)
0345 9111   FORMAT('0? EOF BEIM LESEN UEBERSCHRIFTEN')
0346      STOP 13
0347 912    WRITE(IOLPT,9121)
0348 9121   FORMAT('0? SCHRIFT IN UEBERSCHRIFT ZU KLEIN')
0349      STOP 14
0350 913    WRITE(IOLPT,9131)
0351 9131   FORMAT('0? ZUVIELE ZEILEN IN UEBERSCHRIFT')
0352      STOP 15
0353 914    WRITE(IOLPT,9141)
0354 9141   FORMAT('0? MEHR ALS EINE SEITE FUER UEBERSCHRIFT')
0355      STOP 16
0356 915    WRITE(IOLPT,9151)
0357 9151   FORMAT('0? UEBERSCHRIFT BREITER ALS SEITE')
0358      STOP 17
  
```

FORTRAN IV V01C-03  
CORE=31K, UIC=[1,7]

MON 17-DEC-79 13:39:53

PAGE 001

NLO:,LPO:=DT1:[11,13]DISTNC

```
0001      REAL FUNCTION DISTNC(X0,Y0,X1,Y1)
0002      BERECHNUNG DES ABSTANDES ZWEIER PUNKTE
0003
0004      A=X1-X0
0005      B=Y1-Y0
0006      A=A*A+B*B
0007      DISTNC=SQRT(A)
0008      RETURN
0009      END
```

FORTRAN IV V01C-03  
CORE=31K, UIC=[1,7]

MON 17-DEC-79 13:43:14

PAGE 001

NLO:,LPO:=DT1:[11,13]KTRANS

```
0001      SUBROUTINE KTRANS(X,Y,XS,YS)
0002      KOORDINATENTRANSFORMATION
0003
0004      COMMON /PAGING/ NPAGE,PSIZE,XMAX,SRAND,PROUND(20)
0005      A=Y/PSIZE
0006      NP=INT(A)
0007      B=FLOAT(NP)
0008      YS=Y-B*PSIZE
0009      XS=B*(XMAX+SRAND)+X
0010      RETURN
0011      END
```



```
-----
0001 SUBROUTINE LTRANS(XOP,YOP,X1P,Y1P,XS,YS,J)
      C
      C PRUEFUNG OB EINE VERBINDUNG DURCH EINE SEITENGRENZE UNTERBROCHEN WIRD
      C UND BERECHNUNG DES SCHNITTPUNKTES MIT DER SEITENGRENZE
      C (XS,YS) SCHNITTPUNKT
      C J 0, FALLS KEINE UNTERBRECHUNG VORLIEGT
      C
      C -----
0002 COMMON /PAGING/ NPAGE,PSIZE,XMAX,SRAND,PBOUND(20)
      C
0003 X0=XOP
0004 Y0=YOP
0005 X1=X1P
0006 Y1=Y1P
0007 IF (Y0.LE.Y1) GOTO 5
0009 X0=X1P
0010 Y0=Y1P
0011 X1=XOP
0012 Y1=YOP
0013 XOP=X0
0014 YOP=Y0
0015 X1P=X1
0016 Y1P=Y1
0017 5 J=0
0018 IF (Y0.EQ.Y1) RETURN
0020 DO 10 I=1,NPAGE
0021 IF (Y0.GE.PBOUND(I)) GOTO 10
0023 IF (Y1.LT.PBOUND(I)) RETURN
0025 YS=PBOUND(I)
0026 XS=X0+((X1-X0)*(YS-Y0))/(Y1-Y0)
0027 J=1
0028 RETURN
0029 10 CONTINUE
0030 END
```

0001      SUBROUTINE NEXT(IST,XPOS,YPOS,X0,Y0,NELM,NE)

        NEXT   WAHLT DAS NAECHSTE ZU PLOTTENDE ELEMENT AUS.

        PARAMETER

        IST     STATUS FUER JEDES ELEMENT

        XPOS    X - KOORDINATE FUER JEDES ELEMENT

        YPOS    Y - KOORDINATE FUER JEDES ELEMENT

        (X0,Y0)   POSITION DES ZEICHENSTIFTES

        NELM    ANZAHL DER ELEMENTE

        NE     NUMMER DES AUSGEWAELHTEN ELEMENTES

0002      DIMENSION XPOS(500),YPOS(500),IST(500)

0003      NE=0

0004      XMIN=999.99

0005      ZUSLG=100.0

0006      DO 10 I=1,NELM

0007      IF (IST(I).NE.0) GOTO 10

0009      X1=XPOS(I)

0010      Y1=YPOS(I)

0011      D=DISTNC(X0,Y0,X1,Y1)

0012      IF(X1.GT.X0) D=D+ZUSLG

0014      IF (D.GT.XMIN) GOTO 10

0016      XMIN=D

0017      NE=I

0018      10   CONTINUE

0019      IF (NE.GT.0) RETURN

0021      STOP 7777

0022      END

Manuskripte aus dem Institut für Betriebswirtschaftslehre  
der Universität Kiel

Letzte Nummern:

- Nr. 70 K. Brockhoff, Die Beeinflussung von Kostenabweichungen  
im Entwicklungsbereich, September 1979
- Nr. 71 W. Söhnchen, Zur Entwicklung eines Modells hierarchi-  
scher Gesamtplanung, September 1979
- Nr. 72 R. Schmidt und L. Ahrens,  
Explanation of Stock Price Behavior by  
Computer-Aided Content Analysis,  
September 1979
- Nr. 73 R. Schmidt und W. Janowski,  
PLASMA II, An Interactive Modeling System  
for Mathematical Programming,  
September 1979