KERNFORSCHUNGSZENTRUM

KARLSRUHE

November 1977 KFK 2548

Institut für Datenverarbeitung in der Technik

Prozeßbildcompiler: Sprache und System

H. Grauer, V. Jarsch, W. Müller, W. Lemperle

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.

KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 2548

Institut für Datenverarbeitung in der Technik

Prozeßbildcompiler: Sprache und System

H. Grauer
V. Jarsch
W. Müller
W. Lemperle

Gesellschaft für Kernforschung m.b.H., Karlsruhe

Kurzfassung

Der Prozeßbildcompiler ist ein System zur Definition und Erzeugung von Prozeßbildern für die graphische Prozeßüberwachung mit Farbvideosymbolsichtgeräten.

Nach einer kurzen Einführung in den Systemaufbau werden eine genaue Beschreibung der Sprachelemente für die Bilddefinition von Prozeßbildern und ihrer Anwendung sowie eine Anleitung für die Benutzung und Installation der Module des Prozeßbildcompilers gegeben.

Process-image-compiler: language and system

Abstract

The process-image-compiler is a system for the definition and generation of process images for graphical process control with coloured video-symbol-displays.

After a short introduction into the design of the system an exact description of the language elements for process image definition and their usage is given followed by a guide for the usage and installation of the systems modules.

I n h a l t

Seite

1. Einleitung

Der Prozeßbildcompiler ist ein System zur Definition und Erzeugung von Prozeßbildern für die graphische Überwachung von Prozessen mit Farbvideo-Symbolsichtgeräten. Grundlage dieses Systems ist die Prozeßbildbeschreibung durch Bildprozeduren und die Beschreibung der graphischen Repräsentation von Prozeßvariablen durch Parameter von Bildprozeduren /Gra 1; Gra 2/. Dieser Ansatz ermöglicht

- den strukturierten Aufbau von Prozeßbildern durch eine Menge von Bildprozeduren (Teilbilder). Mehrere gleichartige oder ähnliche Bildkomponenten brauchen nur einmal durch eine Bildprozedur definiert werden und können dann beliebig oft mit verschiedenen Parametern in Bildern benutzt werden.

- eine flexible Repräsentation der Zustände von Prozeßvariablen durch Bildvariable. Das Aussehen des Bildes kann in beliebiger Weise durch den Prozeß, bzw. das Prozeßprogramm gesteuert werden.

- eine sehr einfache datenorientierte Schnittstelle zwischen Prozeßprogramm und graphischem System.

Da bei diesem Ansatz die Prozeßbilder sehr wenig Speicherplatz in Anspruch nehmen und daher kernspeicherresident gehalten werden können, sowie der bildschreibende Teil des Systems und der zur Laufzeit nötige bilderzeugende Teil auf verschiedenen Rechnern laufen können, ist der Prozeßbildcompiler besonders für graphische Prozeßüberwachungssysteme mit sehr kleinen kostengünstigen Rechneranlagen ohne schnellen Externspeicher geeignet.

Dieser Bericht beschreibt die Sprache für die Definition der Prozeßbilder sowie kurz das System für ihre Erzeugung und gibt eine ausführliche Benutzungs- und Installationsanleitung des Prozeßbildcompilers.

Der Prozeßbildcompiler wurde im Rahmen eines Industrieforschungsauftrages für den Einsatz in der Prozeßwarte eines Energieversorgungsunternehmens (EVU) auf einer PDP 11/35 entwickelt /Gra 3/. Alle rechner- und anwendungsspezifischen Angaben beziehen sich auf diese Installation /DEC/.

Das System an sich ist durch Implementierung in FORTRAN IV und eine sichtgeräteunabhängige Schnittstelle rechner- und geräteunabhängig.

2. Arbeitsweise des Prozeßbildcompilers

Der Prozeßbildcompiler besteht aus zwei Teilsystemen: einem bildbeschreibenden System und einem bilderzeugenden System. Das bildbeschreibende Teilsystem wird in der Bilddefinitionsphase benutzt, das bilderzeugende Teilsystem wird für die Bildausgabe in der Bildgenerierungsphase benötigt.

2.1 Bilddefinitionsphase

In der Bilddefinitionsphase werden die Prozeßbilder mit der Bildbeschreibunssprache erstellt. Dazu werden die Teilbilder,aus denen die Prozeßbilder bestehen, durch Bildprozeduren beschrieben. Ein Teilbild kann anlagenspezifische und prozeßspezifische Parameter enthalten. Die anlagenspezifischen Parameter werden zur Bilddefinitionszeit in den das Teilbild aufrufenden (Teil)-Bildern durch aktuelle Argumente erklärt. Sie ermöglichen eine flexible Definition und damit eine vielseitige Verwendbarkeit der Teilbilder. Die prozeßspezifischen Parameter werden erst in der Bildgenerierungsphase durch aktuelle Argumente (Prozeßvariable) von dem den Prozeßbildcompiler aufrufenden Prozeßprogramm besetzt. Sie bestimmen das dynamische Aussehen des Bildes abhängig vom Zustand der Prozeßvariablen.

Die Teilbilder werden von einem Übersetzer einzeln in einen internen Code übersetzt und auf die Objektbilddatei(en) abgelegt.

Für den Aufbau eines gewünschten Prozeßüberwachungssystems werden alle in den zugehörigen Prozeßbildern benötigten Teilbilder einem Binder übergeben. Dieser löst die in den Teilbildern vorhandenen externen Referenzen auf, führt eine Codeoptimierung durch und legt den resultierenden Bildcode in eine Bilddatei ab.

2.2 Bildgenerierungsphase

In der Bildgenerierungsphase braucht nur das für die Bilderzeugung benötigte Teilsystem des Prozeßbildcompilers (Modul I) und die Bilddatei, in der die Prozeßbilder abgelegt sind, auf dem Rechner vorhanden sein.

Das Prozeßprogramm ruft den Prozeßbildcompiler, bzw. den bilderzeugenden Systemteil, zyklisch oder ereignisgesteuert mit den Prozeßvariablen als Parameter auf.

Der Prozeßbildcompiler erzeugt das aktualisierte Prozeßbild (interpretativ) und gibt es im Code des jeweiligen Sichtgerätes auf eine Gerätedatei aus. Von dort wird der Bildcode durch einen Gerätetreiber auf das Sichtgerät übertragen.

3. Bildbeschreibungssprache

Im folgenden werden die Elemente der Sprache, die zur Definition von Bildern benutzt wird, beschrieben:

3.1 Namen

Namen bestehen aus max. 6 alphanumerischen Zeichen. Sie müssen mindestens 1 Alpha- bzw. Sonder-Zeichen enthalten. (FORTRAN.) Sonderzeichen, die keine Trenn- oder Vorzeichen sind, sind als Elemente von Namen zugelassen.

Namen werden zur Bezeichnung von Variablen und Bildkonstanten benutzt.

Daneben gibt es reservierte Namen für Befehlskeyworte, Datenarten, Symbolkonstanten und Texturkonstanten.

3.2 Konstanten

- Symbolkonstanten

  im System festgelegte Namen für Symbole (des Hardware-Symbolsatzes).

- Texturkonstanten

  im System festgelegte Namen für Texturen bzw. Zustände con Symbolen (Farbe, Blinken, Ein/Aus).

- Integer

  ganze Zahlen, die vorzeichenbehaftet sein können.

- Zeichenstrings

  Zeichenstrings können alle gültigen FORTRAN-Zeichen enthalten und werden in Hochkommas eingeschlossen (max. Länge 64 Zeichen).

- Meßwertkonstanten

  sind Dezimalbrüche mit einem Punkt vor den Dezimalziffern. Sie dienen der Darstellung von Meßwerten

und werden auf dem Bildschirm durch den entsprechenden ASCII-String repräsentiert.

3.3 Bildkonstante

vom Benutzer frei wählbare Namen für Bilder.

3.4 Trennzeichen

Blank
(allgemeine Trennzeichen)

,
(
) (Klammerung)

; (Befehlsende)

end of record (Kartenende)

3.5 Elementare Datenarten

Die elementaren Datenarten der Sprache sind:

INT	ganze Zahl
MW (n, m)*)	Meßwert = Festkommazahl n = gesamte Anzahl der Ziffern m = Anzahl der Ziffern nach dem Komma Meßwerte werden im Bild als ASCII-Strings im gewählten Format dargestellt
STRINGS (n)	ASCII-String der Länge n
TEX	Textur (Farbe, Blinken, Ein/Aus)
SY	Symbol
FIGUR	Bild (Bildkonstante oder -nummer)

Daneben gibt es Datenarten, die vom Benutzer selbst erklärt werden.

---

*) Fehlen n, m, so wird angenommen, daß das Format der Festkommazahl in dem dem Meßwert folgenden Wort erklärt wird: li Byte = n, re Byte = m (nur für Prozeßvariable).

3.6 Systemkonstantennamen

3.6.1 Symbolkonstantennamen

Symbol	Name	Interncode
	LEITG	80
	LS	85
	TR	89
	ETR	72
	KMP	87
	TRFHL	94
	TRFHR	93
	ERD	91
	LERD	69
	LKN	65
	LEER	Ø

3.6.2 Texturkonstantennamen

Aussehen	Name	Intercode
nicht vorhanden	NULL	Ø
schwarz	SCHWA	1
rot	ROT	2
grün	GRUEN	3
gelb	GELB	4
blau	BLAU	5
magenta	MAGTA	6
cyan	CYAN	7
weiß	WEISS	8
schwarz und blinkt	BSCHWA	9
rot und blinkt	BROT	10
grün und blinkt	BGRUEN	11
gelb und blinkt	BGELB	12
blau und blinkt	BBLAU	13
magenta und blinkt	BMAGTA	14
cyan und blinkt	BCYAN	15
weiß und blinkt	BWEISS	16

3.7 Kommentare

Kommentare werden durch * eingeschlossen. Sie können an jeder Stelle im Bildprogramm stehen und alle gültigen Zeichen enthalten sowie sich über mehrere Records erstrecken.

```
* DIESES BILD IST SCHOEN *
```

3.8 Befehle

Die Bilddefinitionssprache kennt folgende Befehlstypen:

- deklarative Befehle
  - zur Deklaration einer Bildprozedur (BILD) (END)
  - zur Deklaration der Variablen einer Bildprozedur (DECL)
- graphische Befehle
  - zur Ausgabe von Symbolen und alphanumerischen Zeichen (SYMBOL)
  - zum Setzen des Texturattributs (TEXTUR)
  - zur Positionierung (POSIT)
  - zum Setzen der Zeichenrichtung (ROTATE)
- Bildprozeduraufruf (CALL)
  - zur Benutzung einer Bildprozedur mit aktuellen Parametern
- Bildnummernzuweisung (ASSIGN)

Die Operanden dieser Befehle können Konstante oder Variable sein. Variable erhalten ihre Werte über Parameter der Bildprozeduren.

Befehlsbeschreibung

In diesem Abschnitt werden die einzelnen Befehle beschrieben und kurz Beispiele für ihre Anwendung gegeben.

Verwendete Notation:

$\left\{\begin{matrix} a \\ b \\ c \end{matrix}\right\} \triangleq$ a oder b oder c

$[a]_{n1}^{n2} \triangleq$ minimal n1 maximal n2 Verwendungen von a .

Die Befehle sind formatfrei.

ASSIGN [bildkonstante, integer]$_1^n$;

Bildnummernzuweisung

Hierdurch werden Bildnamen (Bildkonstante) an Bildnummern (integer) zugewiesen. Die Bildnummern können im Prozeßparametersatz benutzt werden. Dem vom Prozeßprogramm aufgerufenen (obersten) Bild muß immer eine Bildnummer zugewiesen werden, und es muß mit dieser Bildnummer aufgerufen werden.

Beispiel:

ASSIGN DINKBL , 1 ;

Dem Bild DINKBL wird die Bildnummer 1 zugewiesen.

BILD bildkonstante $[\text{variable}]_{\emptyset}^{n}$

Bildbefehl zur Eröffnung eines Bildes

Hinter dem Namen des Bildes wird die Liste der formalen Parameter des Bildes aufgeführt.

Alle Parameter müssen im Bild mit einem DECL-Befehl erklärt werden.

Beispiel:

BILD ABZWEI ANAME, MESSW ;

eröffnet ein Bild mit dem Namen ABZWEI und den Variablen ANAME und MESSW als Parameter.

CALL {variable / bildkonstante} [{variable / konstante / bildkonstante}]$_0^n$;

Bildaufrufbefehl

Mit dem Bildaufrufbefehl kann in einem Bild ein anderes Bild benutzt werden.

Hinter dem Namen des aufgerufenen Bildes, der eine Variable oder Bildkonstante sein kann, steht die Liste der aktuellen Parameter, mit denen das Bild aufgerufen wird. Die Elemente der Liste der aktuellen Parameter müssen in Anzahl und Datenart mit den Elementen der Liste der formalen Parameter des aufgerufenen Bildes übereinstimmen.

Beim Betreten eines aufgerufenen Bildes wird die aktuelle Position, die aktuelle Zeichenrichtung und die aktuelle Textur des aufrufenden Bildes übernommen. Beim Verlassen des aufgerufenen Bildes (mit END) wird die aktuelle Position, die aktuelle Zeichenrichtung und die aktuelle Textur des aufrufenden Bildes wiederhergestellt.

Beispiel:

```
CALL   ABZWEI  'XDORF' ,  WERT  ;
```

ruft das Bild mit dem Namen ABZWEI auf. Aktuelle Parameter sind die Zeichenstringkonstante XDORF und die Variable WERT .

DECL variable datenart ;

Deklaration von Variablen

a) Deklaration von einfachen Variablen

$$\text{datenart} = \left\{\begin{matrix}\text{variable} \\ \text{elementare Datenart}\end{matrix}\right\} .$$

Die Datenart ist eine elementare Datenart (vgl. 3.5) oder der Name einer vom Benutzer selbst definierten Datenart, d.h. der Name eines Variablenverbundes (vgl. b)).

Beispiel:

```
DECL A SYMBOL  ;
DECL B KOMPLEX ;
```

Die Variable A wird als von der Datenart SYMBOL erklärt und darf nur Symbolwerte enthalten.

Die Variable B wird als von der Datenart KOMPLEX erklärt. Die Erklärung der Datenart (des Verbunds) KOMPLEX geschieht an anderer Stelle (vgl. b)).

b) Deklaration von Variablenverbunden

$$\text{datenart} = \left[\text{variable} \left\{\begin{matrix}\text{variable} \\ \text{elementare Datenart}\end{matrix}\right\}\right]_1^n .$$

Die Datenart eines Variablenverbundes besteht aus der Menge der Variablenverbundkomponenten.

Eine Variablenverbundkomponente ist eine einfache Variable nach a) . Ihre Datenart kann elementar oder wiederum der Name eines Variablenverbundes sein. Eine rekursive Erklärung ist nicht erlaubt.

Die Erklärung eines Variablenverbundes erfolgt in dem Bild, wo Komponenten des Verbundes benutzt werden.

Die Datenart Variablenverbund ermöglicht eine hierarchisch strukturierte und dem jeweils nötigen Detaillierungsgrad angepaßte Verwendung von Variablen. Sie dient zur Erklärung der Variablen des Prozeßparametersatzes (vgl. 4).

Beispiel;

```
DECL KOMPLEX

     HEADER  STRING (10)
       X1    MW (4,2)
      AB1    ABZW
      AB2    ABZW  ;
```

Der Variablenverbund KOMPLEX besteht aus den Variablen HEADER und X1 mit den elementaren Datenarten Zeichenstring und Meßwert und den Variablen AB1 und AB2 mit der Datenart ABZW. ABZW ist hierbei der Name eines Variablenverbundes der an anderer Stelle erklärt wird.

END ;

Endbefehl zum Abschluß eines Bildes

Jedes Bild muß mit einem END-Befehl abgeschlossen werden.

POSIT $\left[\begin{Bmatrix}\text{variable} \\ \text{integer}\end{Bmatrix}_{2}^{2}\right]$;

Positionierungsbefehl

Verschiebt die aktuelle Position von der ab die nächsten Bildelemente gezeichnet werden relativ zur letzten aktuellen Zeichenposition in der aktuellen Zeichenrichtung (vgl. ROTATE). Anfangsposition (0,0) des obersten Bildes ist die untere linke Ecke des Bildschirms (positive rechtshändige kartesische Koordination).

Beispiel:

```
POSIT 10,5 ;
```

Setzt die Position von der ab die folgenden Bildelemente gezeichnet werden um 10 Koordinateneinheiten in x-Richtung und 5 Koordinateneinheiten in y-Richtung relativ zur aktuellen Position weiter.

ROTATE { variable / integer } ;

Richtungsbefehl

Er setzt die Zeichenrichtung (dreht das Koordinatensystem) für die nachfolgend gezeichneten Bildelemente. Die Zeichenrichtung wird in Vielfachen von 90^{o} relativ zur vorhergehenden Richtung festgelegt, wobei $\emptyset = \emptyset^{o}$, $1 = 90^{o}$, $2 = 180^{o}$, $3 = 270^{o}$ etc. Drehung im positiven mathematischen Sinn (entgegen Uhrzeiger).

Die Anfangszeichenrichtung im obersten auf Bild ist 0^{o} (also Ausgabe der Symbole von links nach rechts).

Soweit der Symbolvorrat des eingesetzten Gerätes das zuläßt wird dabei das Symbol entsprechend der Zeichenrichtung mitgedreht.

Beispiel:

ROTATE 1 ;

Drehung der nachfolgenden Bildelemente um 90^{o} zur bisherigen Zeichenrichtung.

SYMBOL [{ variable / Symbolkonstante / Zeichenstring / Meßwert / Wiederholungsausdruck }]$_{0}^{n}$;

Symbolbefehl

Der Symbolbefehl dient zum Zeichnen von Symbolen, ASCII-String und Meßwerten.

Dabei erlaubt

Wiederholungsausdruck := { variable / integer } ([{ variable / Symbolkonstante / Zeichenstring / Meßwert }]$_{0}^{n}$)

die Anwendung von Wiederholungsfaktoren.

Jedes ausgegebene Zeichen setzt die aktuelle Position um 1 Koordinateneinheit in der aktuellen Zeichenrichtung weiter.

Beispiel:

SYMBOL 2 (SART) , REPEAT (LEITG) , "TEXT" ;

Es wird 2mal das Symbol gezeichnet, dessen aktueller Wert durch die Variable SART bestimmt ist, dann wird das Symbol LEITG sooft gezeichnet, wie der aktuelle Wert der Variablen REPEAT ist, abschließend wird der ASCII-String TEXT gezeichnet.

TEXTUR {variable / Texturkonstante} ;

Texturbefehl

Er setzt die Textur (Farbe, Blinken, (nicht) vorhanden) der nachfolgenden Bildelemente.

Beispiel:

TEXTUR BBLAU ;

Die nachfolgenden Bildelemente sind blau und blinken.

TEXTUR ANW ;

Alle nachfolgenden Bildelemente nehmen als Textur den aktuellen Wert der Variablen ANW an.

4. Prozeßparametersatz: Aufbau und Verwendung

Die Aktualisierung des Prozeßbildes durch die Werte der Prozeßvariablen erfolgt durch Parameterübergabe und wird zur Bilddefinitionszeit durch die Erklärung von entsprechenden Parametern und Variablen in den Teilbildern festgelegt.

Durch die Möglichkeit der Erklärung von Variablenverbunden und ihre Verwendung zur Parameterübergabe brauchen die Prozeßvariablen nicht einzeln an die Bilder übergeben werden, was bei einer großen Anzahl von Variablen aufwendig und unübersichtlich werden würde. Der Benutzer kann die Prozeßvariablen entsprechend ihrer Bedeutung ordnen und kann auf dieser geordneten Menge eine Verbundstruktur definieren, die er für die Verarbeitung in den Bildern benutzt.

Eine solche geordnete Menge von Prozeßvariablen soll als Prozeßparametersatz bezeichnet werden.

Der Prozeßparametersatz der Anwendung des Prozeßbildcompilers zur Überwachung eines Stromversorgungsnetzes enthält 361 Variable zur Darstellung des Prozeßzustandes von maximal 3 Spannungsschienen und 20 Stromabzweigen (Bild 1).

```
Decl  1   Prozeßparametersatz Stromversorgungsnetz
        2   Prozeßbildnummer
        2   Spannungsschienen
          3   Spannungsschiene 1
            4   Zustand Erdschluß 1
            4   Zustand Erdschluß 2
            4   Spannungslos-Anzeige 1
            4   Spannungslos-Anzeige 2
            4   Zustand Schaltelement 1
                          .
                          .
                          .
            4   Zustand Schaltelement 9
          3   Spannungsschiene 2
            4   wie Spannungsschiene 1
                bis Schaltelement 5
          3   Spannungschiene 3
              wie Spannungsschiene 2
        2   Betriebs- und Warnmeldungen
          3   Betriebs- und Warnmeldung 1
                          .
                          .
                          .
          3   Betriebs- und Warnmeldung 30
        2   Stromabzweige
          3   Abzweig 1
            4   Angewählt ja/nein
            4   Name
            4   Meßwert 1
            4   Meßwert 2
            4   Zustand Schaltelement 1
                          .
                          .
                          .
            4   Zustand Schaltelement 8
            4   Elementart Abschluß
            4   Zustand Erdschluß
          3   Abzweig 2
              wie Abzweig 1
                        .
                        .
                        .
          3   Abzweig 20
```

Bild 1: Prozeßparametersatz Stromversorgungsnetz

Die Verwendung dieses Prozeßparametersatzes zur Bildaktualisierung demonstriert das nachstehende Bildprogramm für ein Prozeßbild mit 1 Spannungsschiene und 9 Stromabzweigen (Bild 2)*).

Die oberste Bildprozedur 2Ø OEST beschreibt den Bildaufbau und die Grobstruktur des Prozeßparametersatzes. Sie benutzt die Bildprozeduren SSU2 für die Spannungsschiene, ABZ1M für die Stromabzweige und NAMWRN für die Betriebs- und Warnmeldungen. In diesen werden die Teilstrukturen des Prozeßparametersatzes detailliert beschrieben und ihre Elemente zur Bildaktualisierung benutzt.

---

*) Ohne Betriebs- und Warnmeldungen.

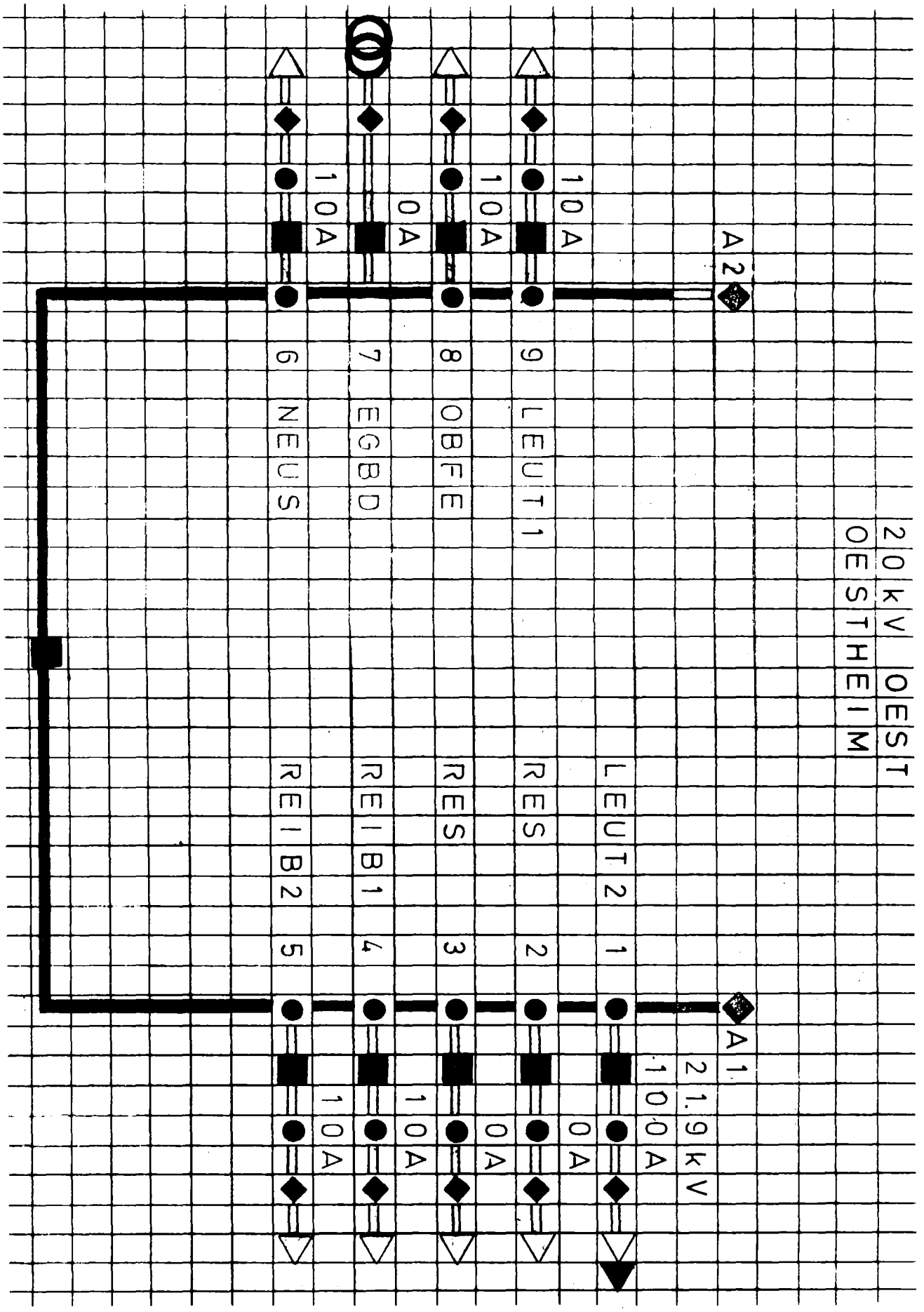


Bild 2: Prozeßbild OESTHEIM

```
*  20 KV OESTHEIM  *
BILD 20OEST,P20OES;
        DECL P20OES
             BILNR  INT
             PL2     INT
             PSS1   SSTYP1
             PSS2   SSTYP2
             PSS3   SSTYP3
             WARN   NWWARN
             FELD1  PABZ1M
             FELD2  PABZ1M
             FELD3  PABZ1M
             FELD4  PABZ1M
             FELD5  PABZ1M
             FELD6  PABZ1M
             FELD7  PABZ1M
             FELD8  PABZ1M
             FELD9  PABZ1M
                 FELD10  MESSPE  ;
 ASSIGN  20OEST,13;
*  BILDNAME UND WARNMELDUNGEN  *
        CALL NAMWRN,' 20KV OEST',    '      OESTHEIM      ',WARN;
*  SPANNUNGSSCHIENE  *
        CALL SSU2,'A2','A1',PSS1;
*  ABZWEIGE RECHTE SEITE  *
        POSIT  36,0;
  *  MESSPUNKT UEBER FELD 1.  PARAMETER AUS FELD 10  *
         CALL MSPUE,FELD10,34;
        CALL ABZ1M,' 1',FELD1,1,31,0;
        CALL ABZ1M,' 2',FELD2,1,29,0;
        CALL ABZ1M,' 3',FELD3,1,27,0;
        CALL ABZ1M,' 4',FELD4,1,25,0;
        CALL ABZ1M,' 5',FELD5,1,23,0;
*  ABZWEIGE LINKE SEITE  *
        POSIT  -2,0;   ROTATE  2;
        CALL ABZ1M,' 6',FELD6,1,-23,2;
        CALL ABZ1M,' 7',FELD7,1,-25,2;
        CALL ABZ1M,' 8',FELD8,1,-27,2;
        CALL ABZ1M,' 9',FELD9,1,-29,2;
END;                        * ENDE  20 KV OESTHEIM *
```

Bildprozedur 2Ø OEST

```
*  INNERE SPANNUNGSSCHIENE, U-FOERMIG, TYP 1  *
BILD SSU2, NAMLI, NAMRE, SSTYP1;
        DECL NAMLI STRING (2);
        DECL NAMRE STRING (2);
        DECL SSTYP1
                ERDRE TEX
                ERDLI TEX
                SPGLRE   TEX
                SPGLLI   TEX
                SSEL1    TEX
                SSEL2    TEX
                SSEL3    TEX
                SSEL4    TEX
                SSEL5    TEX
                SSEL6    TEX
                SSEL7    TEX
                SSEL8    TEX
                SSEL9    TEX          ;
*  POSITION BEI AUFRUF :  BILDURSPRUNG  *
   POSIT  25,40;
*  LINKER ERDUNGSTRENNER  *
    TEXTUR SSEL1; SYMBOL ETR;
  *LINKER NAME  *
  TEXTUR GELB;  SYMBOL NAMLI;
  POSIT 15,0;
*  RECHTER NAME  *
    TEXTUR GELB;SYMBOL NAMRE  ;
*  RECHTER ZWEIG SPANNUNGSLOS  *
  TEXTUR SSEL5;  SYMBOL ETR;
     POSIT -1,-1;   ROTATE -1;
    TEXTUR SPGLRE;  SYMBOL LEITG;
*  LINKER ZWEIG SPANNUNGSLOS  *
     POSIT -1,-20;  TEXTUR SPGLLI;  SYMBOL LEITG;
*  LINKER AST  *
    TEXTUR ROT;
     SYMBOL  25(LEITG);
    ROTATE  1;
*  QUERVERBINDUNG  *
     SYMBOL LKN, 19(LEITG);
*  RECHTER AST  *
    ROTATE 1;SYMBOL LKN, 25(LEITG);
*  ERDSCHLUSS RECHTS  *
     ROTATE -1;  POSIT 0,3;  TEXTUR ERDRE;  SYMBOL 'E';
     POSIT -21,0;  TEXTUR ERDLI;  SYMBOL 'E';
*  TRENNER AN QUERVERBINDUNG  *
     POSIT  7,-29;
    TEXTUR SSEL2;SYMBOL ETR,LEER;
     TEXTUR SSEL3;  SYMBOL LS, LEER;
     TEXTUR SSEL4;  SYMBOL ETR;
END;
*
```

Bildprozedur SSU2

```
*  BILD ABZWEIG TYP1, LANG, MIT MESSWERT *
BILD ABZ1M,A1NRM,PABZ1M,LGE1M,POSY1M,ROT1M;
        DECL A1NRM STRING (2);
        DECL PABZ1M
             ANGEW   TEX
             NAME    STRING (6)
             MESSA   MW (4,0)
             MESSI   MW (3,0)
             EL1     TEX
             EL2     TEX
             EL3     TEX
             EL4     TEX
             EL5     TEX
             EL6     TEX
             EL7     TEX
             EL8     TEX
             EL8ART  FIGUR
             ERDE    TEX    ;
*  LGE1M =0 WENN KEINE INNERE SPANNUNGSSCHIENE, SONST 1 *
        DECL LGE1M   INT ;
        DECL POSY1M  INT ;
        DECL ROT1M   INT ;
*  ANFANGSPOSITION,  1. ZEICHEN DES NAMENS  *
        POSIT  0,POSY1M;
        TEXTUR GELB; SYMBOL NAME, LEER;
        TEXTUR ANGEW; SYMBOL A1NRM;
      POSIT LGE1M,0;
*  LEITUNG, ZUSAMMENGESETZT AUS LINKEM TEIL BEI 2 SS UND RECHTEM TEIL *
        TEXTUR CYAN;
        SYMBOL LGE1M(LEITG),13(LEITG);  POSIT -1,0;
*  ABGANGS- U. ERDUNGSSYMBOL  *
        TEXTUR EL8; CALL EL8ART,ERDE;
*  TRENNER INNERE SPANNUNGSSCHIENE  *
        POSIT  -14,0;  TEXTUR EL1; SYMBOL  TR, LEER ;
*  TRENNER AEUSSERE SPANNUNGSSCHIENE  *
        TEXTUR EL2; SYMBOL TR, LEER, LEER;
*  LEISTUNGSSCHALTER  *
        TEXTUR EL3;  SYMBOL  LS, LEER, LEER;
*  TRENNER 6  *
        TEXTUR EL6;  SYMBOL  TR, LEER , LEER ;
*  ERDUNGSTRENNER  *
        TEXTUR EL7; SYMBOL ETR;
*  MESSWERT  *
        TEXTUR GELB;
        POSIT -8 ,0;
        ROTATE ROT1M;
        POSIT  0,1;
        ROTATE ROT1M;
        SYMBOL '     A';
        POSIT  -5,0;
        SYMBOL MESSA;
END;                      * ENDE ABZWEIG TYP1  MIT MESSWERT *
```

Bildprozedur ABZ1M

```
BILD NAMWRN,NWKENN,NWNAME,NWWARN;
     DECL NWKENN STRING(10);
     DECL NWNAME STRING (18);
     DECL NWWARN
                T1   TEX
                 .    .
                 .    .
                 .    .
                T30  TEX

*  KENNZEICHNUNG DES BILDES  *
   POSIT 31,42;
     TEXTUR GELB;
     SYMBOL NWKENN;
*  BILDNAME  *
   POSIT -14,-1;SYMBOL NWNAME;
*  WARN- UND BETRIEBSMELDUNGEN  *
   POSIT 17,1; SYMBOL 'WARN-U.';
   POSIT -7,-1; SYMBOL 'BETRIEBS-';
   POSIT -9,-1; SYMBOL 'MELDUNGEN';
   POSIT -4,-2;
  ROTATE -1;  *  SCHREIBRICHTUNG UNTEN  *
     TEXTUR T1 ;  SYMBOL '1';
     TEXTUR T2 ;  SYMBOL '2';
     TEXTUR T3 ;  SYMBOL '3';
     TEXTUR T4 ;  SYMBOL '4';
     TEXTUR T5 ;  SYMBOL '5';
     TEXTUR T6 ;  SYMBOL '6';
     TEXTUR T7 ;  SYMBOL '7';
     TEXTUR T8 ;  SYMBOL '8';
     TEXTUR T9 ;  SYMBOL '9';
  ROTATE 1;  *SCHREIBRICHTUNG RECHTS* POSIT -1,0;
     TEXTUR T10; SYMBOL '10';
   POSIT -2,-1; TEXTUR T11; SYMBOL '11';
   POSIT -2,-1; TEXTUR T12; SYMBOL '12';
   POSIT -2,-1; TEXTUR T13; SYMBOL '13';
   POSIT -2,-1; TEXTUR T14; SYMBOL '14';
   POSIT -2,-1; TEXTUR T15; SYMBOL '15';
   POSIT -2,-1; TEXTUR T16; SYMBOL '16';
   POSIT -2,-1; TEXTUR T17; SYMBOL '17';
   POSIT -2,-1; TEXTUR T18; SYMBOL '18';
   POSIT -2,-1; TEXTUR T19; SYMBOL '19';
   POSIT -2,-1; TEXTUR T20; SYMBOL '20';
   POSIT -2,-1; TEXTUR T21; SYMBOL '21';
   POSIT -2,-1; TEXTUR T22; SYMBOL '22';
   POSIT -2,-1; TEXTUR T23; SYMBOL '23';
   POSIT -2,-1; TEXTUR T24; SYMBOL '24';
   POSIT -2,-1; TEXTUR T25; SYMBOL '25';
   POSIT -2,-1; TEXTUR T26; SYMBOL '26';
   POSIT -2,-1; TEXTUR T27; SYMBOL '27';
   POSIT -2,-1; TEXTUR T28; SYMBOL '28';
   POSIT -2,-1; TEXTUR T29; SYMBOL '29';
   POSIT -2,-1; TEXTUR T30; SYMBOL '30';
END;
```

Bildprozedur NAMWRN

5. Module und Dateien des Prozeßbildcompilers

Der Prozeßbildcompiler besteht aus den in Bild 3 aufgeführten Moduln und Dateien.

Zur Bilddefinitionsphase gehören:

Module:	Aufgabe
LA	Lexikalische Analyse der Bildprogramme
UE	Syntaxanalyse und Übersetzung der Bildprogramme [1)]
B	Auflösen der Referenzen der Bildprogramme untereinander und Bestimmung der Adressen der Verbundkomponenten

Dateien:	
Objektbilddatei	Enthält die einzelnen übersetzten Bilder
Bilddatei	Enthält die gebundenen Bilder in Interndarstellung

Zur Bildgenerierungsphase gehören:

Module:	Aufgabe
I	Erzeugt den Gerätecode des aufgerufenen Bildes unter Einbeziehung der aktuellen Werte der Prozeßparameter

Dateien:	
Bilddatei	wie oben
Gerätedatei	Enthält den Gerätecode des Bildes
Prozeßparametersatz	Enthält die aktuellen Werte der Prozeßvariablen

Gerätedatei und Bilddatei sind kernspeicherresident.

---

1) LA und UE werden in den folgenden Abschnitten unter dem Begriff Übersetzer (UE) zusammengefaßt.

Für die Bilderstellung beim Anwender (EVU) sind nur die Bilddatei und der Modul I nötig.

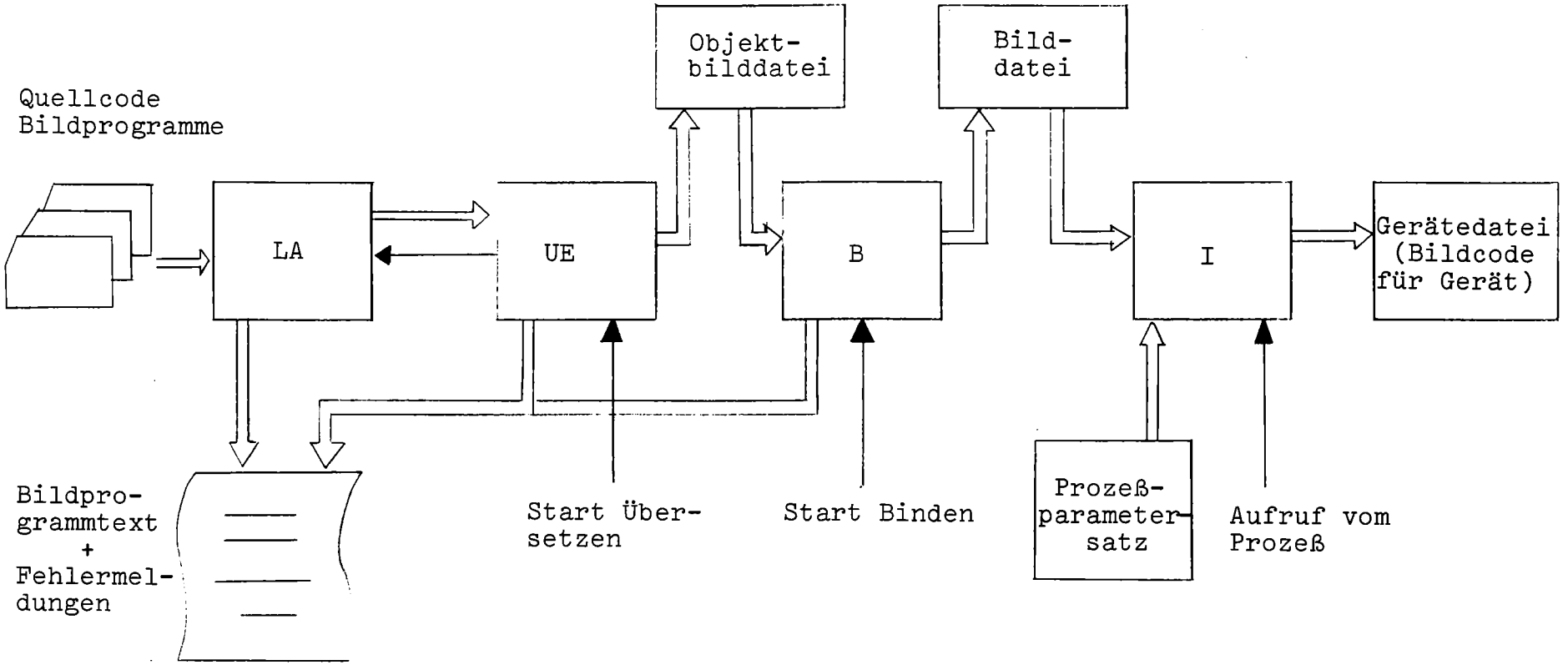
Die Bilddatei muß auf eine dem Prozeßprogramm bekannte Adresse im Speicher geladen werden. Der Modul I wird als Unterprogramm mit dem Prozeßprogramm zusammengebunden.

Die Module LA, UE und B sowie die Objektbilddatei sind nur zur Übersetzung und zum Binden der Bildprogramme notwendig. Sie brauchen nicht beim Anwender (EVU) zu laufen.

Beim Übersetzen der Bilder wird ein Quellcodelisting mit Fehlermeldungen ausgegeben.

Beim Binden der Bilder werden die im Bild definierte Struktur des Prozeßparametersatzes und Fehlermeldungen ausgegeben.

Bild 3: Moduln und Dateien des Bildcompilers



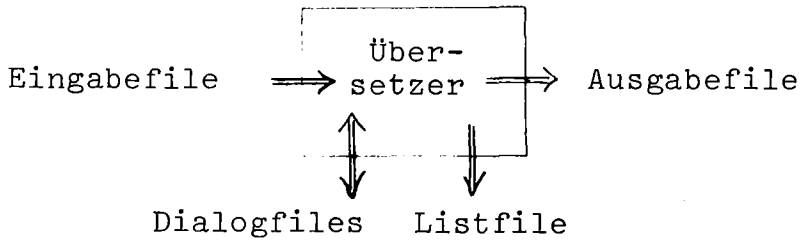
LA = Lexikalische Analyse
UE = Syntaxanalyse und Übersetzung
B = Binder
I = Bildinterpreter

6. Benutzung der Module des Prozeßbildcompilers

6.1 Übersetzer (UE)

Der Übersetzer übersetzt den vom Benutzer mit den Befehlen von 3.8 erstellten und auf einem Eingabefile bereitgestellten Quellcode der Prozeßbilder bildweise in Objektbildcode und legt den Objektbildcode nach Übersetzung aller in dem Eingabefile enthaltenen Bilder auf ein Ausgabefile ab. Der Übersetzer wird als normales Benutzerprogramm aufgerufen (mit .RUN UE).

6.1.1 Übersetzerdateien



Bedeutung der Files:

Eingabefile: Enthält den zu übersetzenden Quellcode eines oder mehrerer Bilder.

Ausgabefile: In diesen File wird der übersetzte Code (Objektbildcode) geschrieben. Er dient als Eingabefile für den Binder.

Dialogfiles: Auf einem Dialogfile werden die Systemanfragen ausgegeben (logische Einheit Nr. 7), von dem anderen Dialogfile die Benutzerkommandos eingegeben (logische Einheit Nr. 5).

Listfile: Auf diesem File werden ein Listing des Quellcodes, eine Beschreibung der externen Variablen sowie Fehlermeldungen zu jedem Bild ausgegeben. Er liegt auf der logischen Einheit mit der Nummer 6.

Die Zuordnung zwischen den ersten beiden logischen Files und physikalischen Files geschieht im Dialog. Die physikalischen Files müssen vor dem Aufruf des Übersetzers angelegt werden. Sie werden nicht dynamisch generiert.

6.1.2 Übersetzeroptionen

Ohne/mit Quellprogrammlisting, Tabelle der externen Variablen und Fehlermeldungen:
Im Bedienerdialog wird angegeben, ob zu jedem Bild ein Quellprogrammlisting, die Tabelle der externen Variablen und Fehlermeldungen ausgegeben werden sollen.
Die Tabelle der externen Variablen enthält alle Namen, die in dem Bild definiert sind und von anderen Bildern referenziert werden können bzw. die in anderen Bildern definiert sind und in dem Bild referenziert werden. Sie gliedert sich nach:

- Namen der Variablen
- Beschreibung der Variablenart

Wird keine Ausgabe des Quellprogramms, der Tabelle der externen Variablen und der Fehlermeldungen gewünscht, so wird für jedes Bild nur der Bildname und die Anzahl der Fehler sowie am Ende der Übersetzung die Summe aller Fehler ausgegeben.

Arten von externen Variablen:

Folgende Variablenarten werden in der Tabelle der externen Variablen aufgeführt:

Variablenart	Bedeutung
BILDDEFINITION	Name des übersetzten Bildes
BILDVERWENDUNG	Name eines mit einem CALL-Befehl aufgerufenen Bildes
BILDNUMMER	Name eines Bildes,dem eine Bildnummer zugewiesen wurde
ARTDEFINITION	Name eines Variablenverbundes (cf. 3.6.1), der in dem Bild deklariert wurde
ARTVERWENDUNG	Name eines Variablenverbundes, der als Datenart in einer Verbunddeklaration benutzt wurde

6.1.3 Fehlermeldungen

Auswirkung von Fehlern:

Treten während des Übersetzens Fehler auf, wird keinkorrekter Code auf das Ausgabefile ausgegeben.

Typen von Fehlern:

- Explizite Fehlermeldungen
  Bei einem Fehler im Quellcode eines Bildes wird der Ort des Fehlers markiert und die Art des Fehlers angegeben.

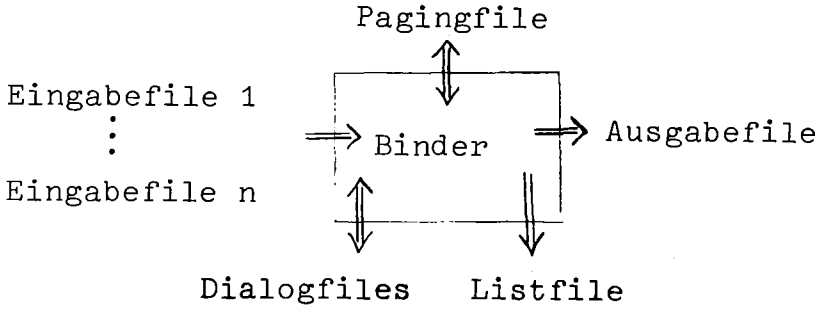
- Systemfehlermeldungen
  Systemfehlermeldungen sind durch eine ganze Zahl gekennzeichnet. Dieser Typ tritt auf bei Überlauf von internen Tabellen des Übersetzers.

- Betriebssystemfehlermeldungen

6.2 Binder (B)

Der Binder bindet die vom Übersetzer erzeugten auf den Eingabefile(s) stehenden Bilder.
Dabei wird ein für den Interpreter ausführbarer Bildcode erzeugt.
Der Binder wird als normales Benutzerprogramm aufgerufen (mit .RUN BINDER)

6.2.1 Binderdateien



Bedeutung der Files:

Eingabefile i: In diesen Files wird der zu bindende Code erwartet.

Ausgabefile: In diesen File wird der gebundene Code geschrieben. Die Größe dieses Files ist maximal 63 Blöcke (zu 256 Worten).

Pagingfile: Dieser File enthält die ausgelagerten virtuellen Tabellen des Binders. Er hat eine Größe von 134 Blöcken (zu 256 Worten) und muß auf einem Gerät mit Direktzugriff liegen.

Dialogfiles: Auf einem Dialogfile werden die Systemanfragen ausgegeben (logische Einheit Nr. 7), von dem anderen Dialogfile die Benutzer-

kommandos eingegeben (logische Einheit Nr. 5).

LISTFILE: Auf diesen File werden Fehlermeldungen und die Beschreibung von Parameterblöcken ausgegeben. Er ist der logischen Einheit mit der Nummer 6 zugeordnet.

Die Zuordnung zwischen den ersten drei logischen Files und physikalischen Files geschieht im Dialog. Die physikalischen Files müssen vor dem Aufruf des Binders angelegt werden. Sie werden nicht dynamisch generiert.

6.2.2 Bindероptionen

Ohne/mit Parameterbeschreibung:
Im Bediendialog wird angegeben, ob der Binder eine Parameterbeschreibung erstellen soll. Wenn ja, dann wird zu jedem Bild, dem eine Nummer zugeordnet wurde und das höchstens einen Parameter hat, eine sölche Beschreibung angefertigt. Darin findet man folgende Informationen:

- Die Nummer des Bildes, dessen Parameter beschrieben wird
- Die Relativadressen der Komponenten des Parameterbereichs, beginnend bei 0.
- Die Typen der Komponenten des Parameterbereichs. Unterstrukturen sind dadurch gekennzeichnet, daß die Typbeschreibung der Komponenten gegenüber der Oberstruktur eingerückt sind.

6.2.3 Fehlermeldungen

Auswirkung von Fehlern:
Treten während des Bindens Fehler auf, wird kein Code auf den Ausgabefile ausgegeben. In der Regel wird jedoch versucht,

den Bindevorgang vollständig abzuschließen.

Typen von Fehlern:

- Explizite Fehlermeldungen

- Systemmeldungen
  Systemfehlermeldungen sind durch eine ganze Zahl gekennzeichnet.
  Dieser Typ tritt auf bei Überlauf von internen Tabellen des Binders.

- Betriebssystemfehlermeldungen

6.3 Interpreter (I)

Der Interpreter (Bildgenerator) arbeitet auf der von Übersetzer und Binder erzeugten Bilddatei und dem Prozeßparametersatz und legt die Sichtgerät-orientierte Bildinformation im Display-File ab.
Er ist in zwei Versionen verfügbar:

(1) als stand-alone-Programm

(2) in einer RT11-Version, die in Verbindung mit einer interaktiven Testhilfe das Austesten der Bilder unterstützt.

6.3.1 Ausführung des Interpreters

Der Interpreter ist als Unterprogramm realisiert und hat nach außen folgende Schnittstelle:

```
subroutine  IXINT (BD, BNR, PRMBER, DSPFLE)
```

mit BD = Anfangsadresse der Bilddatei
(die vom Übersetzer und Binder erzeugte Datei)

BNR = Bildnummer (dezimal!)
Datentyp: integer

PRMBER = Anfangsadresse des Prozeßparameterbereichs relativ zum Anfang der Bilddatei (= Offset auf BD)

DSPFLE = Anfangsadresse des Display-File.
Größe bei der gegenwärtigen Anwendung 3456 integer * 2-Worte.

Das Display-File enthält als Ergebnis eines Interpreter-Laufes die vollständige Bildinformation im Sichtgerät-orientierten Code.
Der Inhalt von DSPFLE wird von einem Gerätetreiber an das Sichtgerät ausgegeben.

(1) Ausführung der stand-alone Version:

Der Interpreter wird als Unterprogramm mit dem oben beschriebenen Namen und Parametern aufgerufen.

(2) Ausführung der RT11-Version:

Der Interpreter wird mit .RUN INT gestarteten und dann über einen selbsterklärenden Dialog mit Parametern versorgt.

Beide Versionen können wahlweise mit oder ohne Fehlerprüfung arbeiten. Fehler werden bei der stand-alone Version auf dem Bildschirm markiert, bei der RT11-Version können
Form von Fehlermeldungen auf der logischen Einheit mit der Nummer 5 ausgegeben werden.

6.3.2 Fehlermeldungen

Fehlercode	Bedeutung	Verhalten des Programms
1	Bildraumüberschreitung	Zeichen wird unterdrückt, Position weitergeschaltet, Stelle im Bild wird markiert
2	integer over- od. underflow beim Setzen des Positionsregisters	Wertzuweisung wird ingoriert, Stelle im Bild wird markiert
3	falsche Textur	Textur bleibt wie vorher, Stelle im Bild wird markiert
4	falscher Zeichencode	falsches Zeichen wird durch Fehlerzeichen ersetzt und - außer bei Strings - die Stelle markiert. Bei Strings wird die Markierung erst am Ende des Strings gesetzt.
5	unbekannter Bildname	Aufruf wird ignoriert, Aufrufstelle wird im Bild markiert
6	falscher Parametertyp bei Call	Aufruf wird ignoriert, Aufrufstelle wird im Bild markiert
7	zu wenig Parameter in Call-Parameterliste	Aufruf wird ignoriert und die Aufrufstelle im Bild markiert
8	zu viel Parameter in Call-Parameterliste	Aufruf wird ignoriert und die Aufrufstelle im Bild markiert

Fehlercode	Bedeutung	Verhalten des Programms
9	Bildname fehlt in Parameterliste	Aufruf wird ignoriert und die Aufrufstelle im Bild markiert
10	zu viele Calls offen	Call wird nicht ausgeführt. Die Stelle wird im Bild markiert.
11	falscher Operationscode	Abbruch (wahrscheinlich falscher Offset beim Aufruf)
12	Meßwert falsch	Kennzeichnung der Stelle, an der der Wert ausgegeben werden soll. Die Meldung kommt vom Prozeßüberwachungsprogramm

6.3.3 Interaktive Testhilfe für FÜW-Stationsbilder

Diese Testhilfe erleichtert das Austesten der Bilder mit der RT11-Version des Interpreters.

Das Programm wird mit dem Interpreter zusammengebunden. Es lädt zunächst die Bilddatei und erlaubt dann interaktives Manipulieren des Prozeßparameterbereichs und somit eine Simulation des Prozesses über das Terminal.

Das Programm meldet sich mit '?'.

Eingaben folgender Art sind in beliebiger Reihenfolge möglich:

<Schlüsselwort><Elementnummer>,|<Teil-El.Nr.>|,<Neuer Eintrag> CR

Schlüsselwörter:	S	Spannungsschiene
	W	Warnmeldung
	A	Abzweig-Element
	B	Abzweig-Erdschluß
	D	Abzweig angewählt
	M	Abzweig-Meßwert
	T	Typ des Abgangselements
	N	Abzweigname
	X	Bildausführung
	I	Unmittelbare Bildausführung 'EIN'
	R	Unmittelbare Bildausführung 'AUS'
	U	Automatische Anwahl aller Abzweige
	E	Programmende

Elementnummer:	1, 2, 3	Typ einer Spannungsschiene	(bei S)
	1-30	Nummer der Warnmeldung	(bei W)
	1-X	Nummer des Abzweigs	(bei A,N,M,D,B)
	Bildnummer		(bei X)

Teil-Elem. Nr.:	1-8	Abgangselemente
	1,2	1. oder 2. Meßwert des Abzweigs
	1-X	Spannungsschienen-Element

Neuer Wert:	Bei Namen	Erst CR drücken, dann ASCII-Zeichen eingeben,
	Bei Meßwert	den Wert entsprechend den Konventionen eingeben,
	Bei Textur	den Interncode (1-16) der Textur eingeben (vgl.3.7.2),
	Bei Abgangselementtypen	den Namen (Nummer) des Typs eingeben.

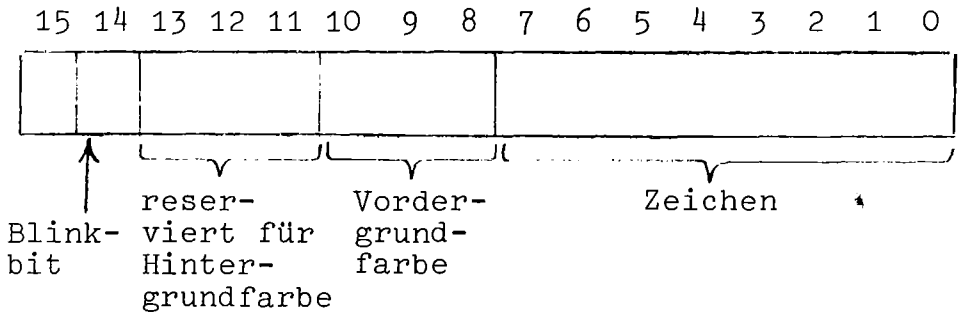
Anmerkung:

Bei der unmittelbaren Bildausführung wird nach jeder Eingabe das Bild ausgegeben, dessen Nummer bei der letzten expliziten Bildausführung (mit X) angegeben wurde.

Das Testprogramm setzt voraus, daß der Prozeßparametersatz wie bei FÜW (26.05.76) aufgebaut ist.

6.3.4 Belegung des Display-Files

Die Information für je 1 Rasterfeld des Bildschirms ist in je 1 Wort codiert:



Die Bildinformation ist zeilenweise (von oben nach unten) in dem eindimensionalen Display-File (nach aufsteigenden Adressen) abgelegt.

6.4 Beispielprogrammlisting

Mit dem folgenden Beispielprogramm sollen neben Sprach-eigenschaften der Dialog, die Optionen und die Meldungen von Übersetzer (UE) und Binder (B) des Prozeßbildcompilers vorgestellt werden.

```
RUN DX1:UE


PROZESSBILDCOMPILER - UEBERSETZER -
    REVISION 1  LEVEL 1


LISTING GEWUENSCHT ? (NEIN = FALSE/ JA=TRUE)
TRUE

AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:UEFIL.DAT

EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:BEISP DAT

          1   *MIT DIESEM BEISPIEL SOLLEN  SPRACHEIGENSCHAFTEN  VORGESTELLT
          2    WERDEN. DIE BEDEUTUNG DES DAMIT ERZEUGTEN BILDES KANN AUSSER
          3    ACHT BLEIBEN.*
          4   BILD BILD1,PARAM;
          5   DECL PARAM        *VEREINBARUNG DES PARAMETERS*
          6      ELEM1 TEX      *TEXTUR*
          7      WERT1 MW(5,2) *MESSWERT MIT 5 ZIFFERN, DAVON 2 DEZIMALSTELLEN*
          8      WERT2 MW( )    *MESSWERT MIT VARIABLEM FORMAT*
          9      CHAR  SY       *EINZELZEICHEN*
         10      SATZ2 KOMP     *UNTERVERBUND*
         11      ROTA  INT;     *INTEGERWERT*
         12   TEXTUR ELEM1;
         13   POSIT 2,36;
         14   ROTATE ROTA;
         15   SYMBOL ETR,LS,WERT1,6('+'),WERT2,CHAR;
         16   ROTATE 1;
         17   CALL BILD2,SATZ2;
         18   ASSIGN BILD1,1;        *WEISE DEM BILD1 DIE NUMMER 1 ZU*
         19   END;                   *ENDE DES BILDES BILD1*


       TABELLE DER EXTERNEN VARIABLEN

       NAME              TYP
       BILD1             BILDDEFINITION
       PARAM             ARTDEFINOTION
       KOMP              ARTVERWENDUNG
       BILD2             BILDVERWENDUNG
       BILD1             BILDNUMMER

       FEHLER:     0

       BILDENDE


                 ********************


          1   BILD BILD2,KOMP;
          2   DECL KOMP     *VEREINBARE DEN PARAMETER, DER ZUSAMMENGESETZT  IST
          3                  AUS EINER TEXTUR, EINEM TEXT MIT 12 ZEICHEN, EINEM
          4                  INTEGERWERT UND EINER BILDVARIABLEN. *
          5        ELEM1    TEX
          6        TEXT      STRING12)
( FEHLT
```

```
          7         ZAHL      INT,
          8         BILD      FIGUR;
          9    DECL HILF INT;
                      !
VARIABLE NICHT ALS PARAMETER ANGEGEBEN
         10    POSIT 2,INT;     *SETZE DIE POSITION*
                        !
BENUTZTE GROESSE MUSS VARIABLE ODER INTEGER SEIN
         11    TEX       ZAHL;  *SETZE DIE TEXTUR*
                 !
UNGUELTIGER BEFEHLSCODE
         12    SYMBOL 'FEHLER:    ',TEXT;    *GIB EINE ZEICHENKETTE AUS*
                                      !
DATENTYP DER VARIABLEN FUER DIESEN BEFEHL UNZULAESSIG
         13    CALL BILD,TEXT,ELEM1,12,3;  *AUFRUF EINES BILDES MIT PARAMETERN*
         14    ASSIGN BILD2,-1;            *WEISE DEM BILD2 DIE NUMMER 2 ZU*
         15    END;                        *ENDE DES BILD2*

        TABELLE DER EXTERNEN VARIABLEN

        NAME              TYP

        BILD2             BILDDEFINITION
        KOMP              ARTDEFINITION
        BILD              BILDVERWENDUNG
        BILD2             BILDNUMMER

        FEHLER:     5

        BILDENDE


                  ********************


       FEHLER IM UEBERSETZERLAUF:    5

      UEBERSETZERENDE
```

```
RUN DX1:UE

PROZESSBILDCOMPILER - UEBERSETZER -
    REVISION 1  LEVEL 1

LISTING GEWUENSCHT ? (NEIN = FALSE/ JA = TRUE)
F

AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:UEFIL.DAT

EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:BEISP.DAT

        BILD  BILD1

        FEHLER:    0

        BILDENDE


        BILD  BILD2

        FEHLER:    5

        BILDENDE


        FEHLER IM UEBERSETZERLAUF:    5

        UEBERSETZERENDE
```

```
RUN DX1:UE


PROZESSBILDCOMPILER - UEBERSETZER -
    REVISION 1  LEVEL 1


LISTING GEWUENSCHT ? (NEIN = FALSE/ JA = TRUE)
T

AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:UEFIL.DAT

EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:BEISP.DAT

     1   *MIT DIESEM BEISPIEL SOLLEN  SPRACHEIGENSCHAFTEN  VORGESTELLT
     2    WERDEN. DIE BEDEUTUNG DES DAMIT ERZEUGTEN BILDES KANN AUSSER
     3    ACHT BLEIBEN*
     4   BILD BILD1,PARAM;
     5   DECL PARAM         *VEREINBARUNG DES PARAMETERS*
     6      ELEM1 TEX       *TEXTUR*
     7      WERT1 MW(5,2)   *MESSWERT MIT 5 ZIFFERN, DAVON 2 DEZIMALSTELLEN*
     8      WERT2 MW( )     *MESSWERT MIT VARIABLEM FORMAT*
     9      CHAR  SY        *EINZELZEICHEN*
    10      SATZ2 KOMP      *UNTERVERBUND*
    11      ROTA  INT;      *INTEGERWERT*
    12   TEXTUR ELEM1;
    13   POSIT 2,36;
    14   ROTATE ROTA;
    15   SYMBOL ETR,LS,WERT1,6('+'),WERT2,CHAR;
    16   ROTATE 1;
    17   CALL BILD2,SATZ2;
    18   ASSIGN BILD1,1;          *WEISE DEM BILD1 DIE NUMMER 1 ZU*
    19   END;                     *ENDE DES BILDES BILD1*


TABELLE DER EXTERNEN VARIABLEN

NAME              TYP

BILD1             BILDDEFINITION
PARAM             ARTDEFINITION
KOMP              ARTVERWENDUNG
BILD2             BILDVERWENDUNG
BILD1             BILDNUMMER

FEHLER:    0
BILDENDE


          ********************


     1   BILD BILD2,KOMP;
     2   DECL KOMP    *VEREINBARE DEN PARAMETER, DER ZUSAMMENGESETZT  IST
     3                 AUS EINER TEXTUR, EINEM TEXT MIT 12 ZEICHEN, EINEM
     4                 INTEGERWERT UND EINER BILDVARIABLEN*
     5      ELEM1  TEX,
```

```
 6       TEXT    STRING(12)
 7       ZAHL    INT,
 8       BILD    FIGUR;
 9    POSIT 2,ZAHL;                  *SETZE DIE POSITION*
10    TEXTUR ELEM1;*SETZE DIE TEXTUR*
11    SYMBOL 'FEHLER:  ',TEXT;       *GIB EINE ZEICHENKETTE AUS*
12    CALL BILD,TEXT,ELEM1,12,3;  *AUFRUF EINES BILDES MIT PARAMETERN*
13    ASSIGN BILD2,-1;               *WEISE DEM BILD2 DIE NUMMER -1 ZU*
14    END;                           *ENDE DES BILDES BILD2*

TABELLE DER EXTERNEN VARIABLEN

NAME                TYP

BILD2               BILDDEFINTION
KOMP                ARTDEFINITION
BILD2               BILDNUMMER

FEHLER:     0

BILDENDE


              ********************


FEHLER IM UEBERSETZERLAUF:    0

UEBERSETZERENDE
```

```
RUN DX1:UE


PROZESSBILDCOMPILER - UEBERSETZER -
     REVISION 1   LEVEL 1



LISTING GEWUENSCHT ? (NEIN = FALSE/ JA = TRUE)
F

AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:UEFIL.DAT

EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:BEISP.DAT


          BILD BILD1

          FEHLER:    0

          BILDENDE



          BILD BILD2

          FEHLER:    0

          BILDENDE



          FEHLER IM UEBERSETZERLAUF:    0

          UEBERSETZERENDE
```

```
RUN DX1:BINDER


PROZESSBILDCOMPILER - BINDER -
    REVISION 1  LEVEL 1



PARAMETERBESCHREIBUNG GEWUENSCHT (NEIN=FALSE / JA =TRUE)
T

AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:BIFIL.DAT

PAGINGFILE ? (GERAET:FILE.EXTENSION)
*DX0:PAGE.DAT

EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:UEFIL.DAT

WEITERER EINGABEFILE ? (GERAET:FILE.EXTENSION ODER LEERZEILE)
*


BILD MIT DER NUMMER  1
       0  TEX
       1  MW
       2  MW
       3
       4  SY
       5    TEX
       6    STRING
       7
       8
       9
      10
      11
      12    INT
      13    FIGUR
      14  INT




BILD MIT DER NUMMER  -1
       0  TEX
       1  STRING
       2
       3
       4
       5
       6
       7   INT
       8   FIGUR

   0 FEHLER BEIM BINDEN

LAENGE DES ERZEUGTEN CODES:    305 WORTE


******* ENDE DES BINDERS *******
```

```
RUN DX1:BINDER


PROZESSBILDCOMPILER - BINDER -
     REVISION 1   LEVEL 1



PARAMETERBESCHREIBUNG GEWUENSCHT ? (NEIN=FALSE / JA=TRUE)
F

AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:BIFIL.DAT

PAGINGFILE ? (GERAET:FILE.EXTENSION)
*DX0:PAGE.DAT

EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX0:UEFIL.DAT

WEITERER EINGABEFILE ? (GERAET:FILE.EXTENSION ODER LEERZEILE)
*


     0   FEHLER BEIM BINDEN

LAENGE DES ERZEUGTEN CODES:    305 WORTE


******* ENDE DES BINDERS *******
```

7. Übersetzen und Binden der Moduln des Prozeßbildcompilers

7.1 Übersetzer (UE)

7.1.1 Übersetzen

Die Quellprogramme des Übersetzers stehen auf der Datei UE.FOR

7.1.2 Binden

Objektcode-Bibliotheken:

UE.OBJ:	Hauptprogramm + BLOCK DATA UP
LALIB.OBJ:	Lexikalische Analyse und Tabellenverwaltung
USLIB.OBJ:	Syntaxprüfung
ULIB.OBJ:	Semantische Routinen

Aufruf zum Binden:

```
.R LINK
*DX1:UE[85]/B:2000=DX1:UE,USLIB,ULIB,LALIB/F
```

Beachte:

Der Übersetzer benötigt einen großen Laufzeitkeller und muß daher auf die Adresse 2000 gebunden werden.
Die obige Reihenfolge der Bibliotheken ist unbedingt einzuhalten.

7.2 Binder (B)

7.2.1 Übersetzen

Die Quellprogramme des Binders stehen auf der Datei BINDER.FOR

7.2.2 Binden

Objektcode-Bibliotheken:

BINDHP.OBJ:	Hauptprogramm
BINLIB.OBJ:	Binderroutinen
TABLIB.OBJ:	Tabellenverwaltung

Aufruf zum Binden:

```
.R LINK
*DX1:BINDER[70]=DX1:BINDHP,BINLIB,TABLIB/F
```

7.3 Interpreter

7.3.1 Übersetzen

Das Quellprogramm des Interpreters steht auf der Datei INT.FOR.

(1) Version mit Fehlerprüfung

```
.R FORTRA
*DX1:INT=DX1:INT.FOR/D/S
```

In dieser Version prüft der Interpreter zur Laufzeit Fehler (z.B. falsche Parameter) ab. Die Fehler werden im stand-alone-Betrieb nur auf dem Bildschirm gemeldet, indem die Stellen des Bildes, an denen sie auftreten, auffällig markiert werden.

Unter RT11 ist außerdem eine Fehlermeldung über den Drucker möglich, wenn die Datei IXFEDR.OBJ beim Binden angegeben wird. Eine gedruckte Meldung enthält einen Fehlercode und die Position, an der der Fehler aufgetreten ist.

(2) Version ohne Fehlerprüfung

```
.R FORTRA
*DX1:INT=DX1:INT.FOR/S
```

liefert eine Version, die keine der in 6.3.2 genannten Fehler prüft und deswegen schneller läuft. Tritt ein Fehler auf, sind die Ergebnisse nicht vorhersehbar.

7.3.2 Binden

Objektcode-Bibliotheken:

JLIB.OBJ:	Objektbibliothek der fehlerprüfenden Version des Interpreters
QLIB.OBJ:	Objektbibliothek der nicht-fehlerprüfenden Version des Interpreters
TABLIB.OBJ	enthält die externe Referenz HASH1S
IXFEDR.OBJ:	für Fehlerausgabe über Drucker
HAUPT.OBJ:	Assembler-Hauptprogramm für stand-alone-Betrieb
HPINT.OBJ:	Hauptprogramm und interaktive Testhilfe für RT11-Betrieb

Aufruf zum Binden:

(1) Stand-alone ausführbares Programm

```
.R LINK
*DX1:INT/B:20000=DX1:HAUPT,JLIB,TABLIB/L/F
```

(2) Unter RT11 ausführbares Programm

```
.R LINK
*DX1:INT=DX1:HPINT,IXFEDR,JLIB,TABLIB/F
```

8. Umwandlung der Bilddatei in LDA-Format

Während unter RT11 der Interpreter die Bilddatei von der Platte holt, muß sie bei stand-alone Betrieb über Lochstreifen geladen werden.

Vorgehensweise:

1. Umformung der Binder-Ausgabedatei in ein MAC-File mit dem Programm CPMAC.
   (Das MAC-File muß vorher angelegt sein. Größe bei FÜW : 400 Blöcke)

2. Übersetzen des MAC-Files

3. Binden des Objekt-Files
   (Basisadresse angeben!)

4. Ausstanzen des LDA-Files mit PIP.

```
RUN DX1:CPMAC


EINGABEFILE ? (GERAET:FILE.EXTENSION)
*DX1:BIFIL.DAT

LAENGE DER BILDDATEI:    9935 WORTE
AUSGABEFILE ? (GERAET:FILE.EXTENSION)
*DX1:BD.MAC

PROGRAMMENDE



R MACRO
*DX1:BDC80]=DX1:BD.MAC
FREE CORE: 18555. WORDS

*^C




R LINK
*DX1:BD/B:50000/L,TT:=DX1:BD.OBJ



RT-11 LINK    V04-02     LOAD MAP
BD     .LDA              10-DEC-76

SECTION ADDR     SIZE     ENTRY   ADDR     ENTRY   ADDR     ENTRY   ADDR
. ABS.  000000   050000
        050000   046636

TRANSFER ADDRESS = 000001
HIGH LIMIT = 116636

*
```

Literatur:

/Gra 1/ H. Grauer, V. Jarsch, W. Müller
Kostengünstige graphische Prozeßüberwachung durch interpretative Verarbeitung von Prozeßbildprozeduren.
7. Jahrestagung der Gesellschaft für Informatik (wissenschaftliches Programm), Erlangen, 26. - 28.9.1977.

/Gra 2/ H. Grauer, V. Jarsch, W. Müller
Ein kostengünstiges System zur graphischen Prozeßüberwachung mit Farbvideosymbolsichtgeräten.
7. Jahrestagung der Gesellschaft für Informatik (Industrieprogramm), Erlangen, 26. - 28.9.1977.

/Gra 3/ H. Grauer, V. Jarsch, W. Müller
Prozeßbildcompiler: Sprachdefinition und Systembeschreibung.
(1976) unveröffentlicht

/DEC/ RT-11 Fortran Compiler and Object Time System, Users Manual
DEC-11-LRFPA-A-D.