

ESPRESO-W, ein Werkzeug für die Spezifikation
von Prozeßrechner-Software

Klaus Eckert, Jochen Ludewig *)
Kernforschungszentrum Karlsruhe
Institut für Datenverarbeitung in der Technik
Postfach 3640, D-7500 Karlsruhe

Zusammenfassung

ESPRESO (System zur Erstellung der Spezifikation von Prozeßrechner-Software) ist ein rechnerunterstütztes Spezifikationssystem, das spezielle Anforderungen der Spezifikation von Prozeßrechner-Software berücksichtigt. Die Komponenten von ESPRESO sind eine formale Sprache ESPRESO-S und ein Programmsystem ESPRESO-W. Im vorliegenden Beitrag werden die wesentlichen Aspekte von ESPRESO-S kurz zusammengefaßt. Danach werden Konzeption und Aufbau von ESPRESO-W und Probleme seiner Implementierung ausführlich dargestellt.

1. Einleitung

Ein beträchtlicher Teil der Softwarefehler entsteht in der Spezifikations- und Entwurfsphase der Softwareentwicklung (Boehm, 1976). Dies ist vor allem auf den Mangel an geeigneten formalen Hilfs- und Prüfmitteln zurückzuführen. Am Institut für Datenverarbeitung in der Technik, Kernforschungszentrum Karlsruhe, beschäftigt sich eine Gruppe mit der Entwicklung eines Spezifikationssystems. Zuerst wurden Informationen über bereits vorhandene Mittel für die Spezifikation und den Entwurf gesammelt (Ludewig, Streng, 1978). Danach wurde das PSL/PSA-System vom ISDOS-Project (Teichrow, Hershey, 1977) installiert. Dieses System war für die Entwicklung von Prozeßrechner-Software nicht geeignet und wurde durch PCSL (Ludewig, 1980) erweitert. Im Anschluß daran wurde ESPRESO entwickelt und teilweise installiert (Eckert, 1980; Ludewig, 1981a).

*) jetzt BBC Forschungszentrum, CH-5405 Baden-Dättwil

2. Die Sprache ESPRESO-S

ESPRESO-S und ESPRESO-W wurden auf der Grundlage eines geschlossenen Konzepts entwickelt und bedingen sich daher gegenseitig. Deshalb ist hier eine kurze Einführung in die Sprache notwendig; eine ausführliche Darstellung ist bei Ludwig (1981a,b) zu finden.

Anstelle einer Beschreibung von ESPRESO-S sei hier ein kurzes Beispiel angegeben, das die wichtigsten Merkmale der Sprache zeigt. Das Beispiel wird in zwei Schritten entwickelt. Zunächst wird die Aufgabenstellung in ESPRESO-S formuliert:

informal Aufgabenstellung:

```
text ‡ Es ist ein Datenerfassungssystem zu entwerfen, das Prozeßdaten
      (Temperaturen) einliest, filtert und die gefilterten Daten zur
      Weiterverarbeitung bereitstellt. Das Einlesen und Filtern der Daten
      soll parallel ablaufen. Die gefilterten Daten sollen in !Filterdaten
      abgelegt werden. ‡
```

end Aufgabenstellung.

Wie man sieht, sind für informale Beschreibungen in ESPRESO-S Texte vorgesehen. Jeder Text ist einem Objekt zugeordnet. Jedes Objekt ist durch eine Art und einen Namen charakterisiert. In der obigen ESPRESO-Spezifikation wird ein Objekt mit dem Namen "Aufgabenstellung" und der Art "Textobjekt" (informal) vereinbart. In den Texten können Verweise (z.B. "!Filterdaten"), die mit "!" markiert sein müssen, zu anderen Objekten enthalten sein.

Im nächsten Schritt wird das "Datenerfassungssystem" spezifiziert:

```
(* Datenerfassungssystem          Stand vom 7.Jan 1981 *)   (* 1 *)
module Datenerfassungssystem:                               (* 2 *)
                                                           (* 3 *)
comprises                                                  (* 4 *)
  buffer Rohdaten;                                         (* 5 *)
                                                           (* 6 *)
and                                                         (* 7 *)
  procedure Datenerfassen:                                  (* 8 *)
    text Zweck ‡ Einlesen und Filtern der Prozessdaten ‡; (* 9 *)
                                                           (* 10 *)
    parallel                                               (* 11 *)
      block Einlesen:                                       (* 12 *)
        reads Temperatur;                                   (* 13 *)
        produces Rohdaten                                  (* 14 *)
        end Einlesen                                       (* 15 *)
      parallel                                               (* 16 *)
        block Filtern:                                       (* 17 *)
          consumes Rohdaten;                               (* 18 *)
          tests Obergrenze;                               (* 19 *)
          produces Filterdaten                             (* 20 *)
          end Filtern                                       (* 21 *)
        end Datenerfassen                                   (* 22 *)
    end Datenerfassungssystem.                             (* 24 *)
```

Die Beschreibung von "Datenerfassungssystem" enthält mehrere in "(*" und "*)" eingeschlossene Kommentare. Sie sind im Sinne der Spezifikation nicht relevant und werden bei der Verarbeitung durch ESPRESO-W ignoriert. Einem Text kann ein Selektor (z.B. "Zweck", Zeile 9) vorangestellt sein. Unter dem gleichen Selektor kann einem Objekt nur ein Text zugeordnet werden.

Die Spezifikation enthält die Definition mehrerer Objekte. Einige davon sind:

Objektname	Art	definiert in
Datenerfassungssystem	Modul	Zeile 2
Rohdaten	Puffer	Zeile 5
Datenerfassen	Prozedur	Zeile 8
Einlesen	Block	Zeile 12

Die Objekte sind durch Verknüpfungen verbunden, die einer durch die Syntax von ESPRESO-S vorgegebenen Menge von Relationen entnommen sind. Z.B. wird eine Verknüpfung der 2-stelligen Relation "enthält" (comprises) zwischen dem Modul "Datenerfassungssystem" und dem Puffer "Rohdaten" (Zeilen 2-5) angegeben. Weitere Verknüpfungen sind zum Beispiel:

Relation	1. Objekt	2. Objekt	definiert in
lesen	Einlesen	Temperatur	Zeilen 12,13
liefern	Einlesen	Rohdaten	Zeilen 12,14
liefern	Filtern	Filterdaten	Zeilen 17,20
holen	Filtern	Rohdaten	Zeilen 17,18
enthält	Datenerfassungssystem	Datenerfassen	Zeilen 2,4,7,8

Die Anzahl der Komponenten einer Verknüpfung wird von der Relation bestimmt, zu der die Verknüpfung gehört. Es gibt 2- und 3-stellige Relationen, sowie 1-stellige, die als Attribute bezeichnet werden. Die Objekte können entweder indirekt durch Nennung ihres Namens (z.B. "Temperatur", Zeile 13) oder direkt durch eine Sektion (z.B. buffer Rohdaten) eingeführt werden. Objekte, die durch eine Sektion definiert werden, können mit weiteren Objekten verknüpft werden (z.B. procedure Datenerfassen). Die Syntax ist also rekursiv. Die Art eines indirekt genannten Objekts ist nicht vollständig festgelegt, sondern nur durch die Syntax eingeschränkt. Z.B. könnte "Filterdaten" (Zeile 20) die Art Puffer oder Trigger besitzen.

Formal läßt sich eine ESPRESO-Spezifikation als bipartiter Graph darstellen, dessen Knoten Objekte und Verknüpfungen sind. Eine Kante verbindet immer eine Verknüpfung mit einem Objekt und gibt an, daß das Objekt Komponente in der Verknüpfung ist. Die Kanten sind markiert, da die Komponenten eine geordnete Menge bilden. Die Anzahl der Kanten an einem Objekt gibt an, mit wievielen Verknüpfungen das Objekt verbunden ist.

Im folgenden wird ein Teil des bipartiten Graphen, der durch die vorhergehende Spezifikation beschrieben wird, dargestellt:

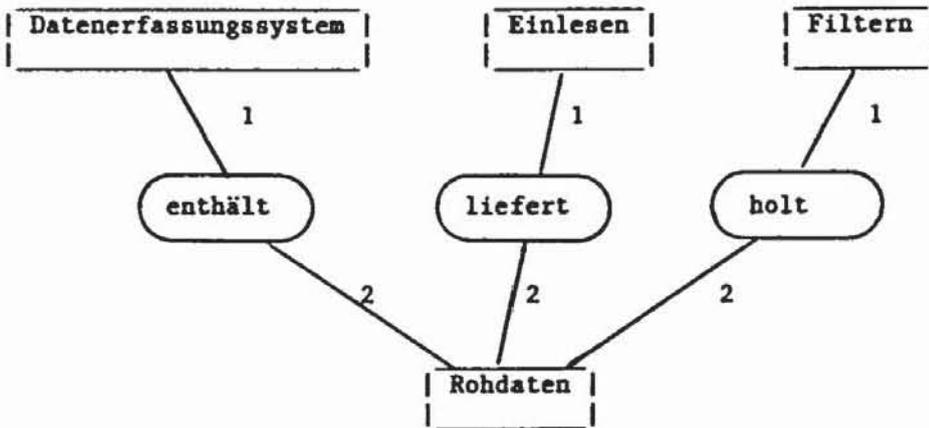


Bild 1: Darstellung einer Spezifikation als bipartiter Graph

ESPRESO-S ist durch eine Erweiterte Attribut-Grammatik (Watt, Madsen, 1977) formal definiert (Ludwig, 1981a). Diese Definition schließt ein:

- die kontextfreie Syntax,
- die kontextsensitive Syntax und
- die Auswirkungen einer Eingabe auf den Inhalt der ESPRESO-Datei (siehe 3.1).

Bei der Definition der Syntax ist darauf geachtet worden, daß ESPRESO-S sehr einfach analysierbar ist. Dies wird durch folgende Eigenschaften gewährleistet:

1. Die kontextfreie Grammatik ist so definiert, daß ESPRESO-S zur Klasse der LL(1)-Sprachen gehört (Knuth, 1971).
2. Die kontextsensitive Grammatik ist so aufgebaut, daß bei einer Linksableitung alle Attributwerte top-down in einem Durchgang bestimmt werden können.

Aufgrund der beiden Eigenschaften läßt sich ESPRESO-S in einem Durchgang sackgassenfrei analysieren.

3. Das Spezifikationswerkzeug ESPRESO-W

ESPRESO-W ist ein Programmsystem, das dem Anwender bei der Speicherung, Prüfung, Veränderung und Auswertung einer ESPRESO-Spezifikation als Werkzeug dient (siehe Bild 2). Der Gebrauch des Werkzeugs ist möglichst einfach gehalten, so daß der Anwender nur wenige Funktionen kennen muß.

ESPRESO-W analysiert eine ESPRESO-Spezifikation syntaktisch und prüft, ob die durch die kontextsensitive Grammatik definierten Restriktionen eingehalten sind. Korrekte ESPRESO-Spezifikationen werden abgespeichert. Da ESPRESO so konzipiert ist, daß eine Spezifikation schrittweise vervollständigt werden kann, erfolgt die Speicherung so, daß sie erweitert werden kann. Bereits verarbeitete ESPRESO-Spezifikationen können verändert werden, wobei durch die Korrekturmöglichkeiten sichergestellt ist, daß die Korrektheit der Spezifikation erhalten bleibt.

ESPRESO-W unterstützt den Anwender bei der Dokumentation und bei der Auswertung der Spezifikation. Hierfür stellt ESPRESO-W verschiedene Report- und Prüffunktionen zur Verfügung, die in jedem Entwicklungsstadium angewandt werden können.

Eines der Ziele bei der Entwicklung von ESPRESO-W war, das Programmsystem auf Kleinrechnern installieren zu können. Die Auswahl von PASCAL als Implementierungssprache ist in dieser Hinsicht günstig, da PASCAL auf vielen Kleinrechnern zur Verfügung steht und dadurch die Portabilität gewährleistet ist.

3.1 Funktionen von ESPRESO-W

Für alle Funktionen von ESPRESO-W ist die Aufbewahrung der Spezifikation von zentraler Bedeutung. Die Überprüfung der Restriktionen, die Reports und die Auswertung einer Spezifikation können nicht auf der Grundlage der Darstellung in ESPRESO-S erfolgen. Deshalb ist es notwendig, eine interne Repräsentation der Spezifikation zu erzeugen und in einer Datei (ESPRESO-Datei) aufzubewahren. Die ESPRESO-Datei enthält die relevante Information einer Spezifikation. Das sind im wesentlichen die Objekte und ihre Verknüpfungen untereinander. In der ESPRESO-Datei werden nur diejenigen Teile einer ESPRESO-Spezifikation gespeichert, in denen kein Fehler aufgetreten ist. Der Anwender kann die interne Repräsentation nicht direkt (z.B. mit Hilfe eines Editors) ändern, so daß für den Inhalt der ESPRESO-Datei die Konsistenz sichergestellt ist.

Die Funktionen von ESPRESO-W sind so aufgebaut, daß in der Spezifikationsphase folgende Vorgehensweise festgelegt ist:

- Die Konvertierung

Eine ESPRESO-Spezifikation wird bei der Eingabe entsprechend der Grammatik geprüft. Dies schließt eine Prüfung auf Konsistenz - auch gegenüber bereits früher gespeicherten Teilen - ein. Die korrekten Abschnitte der Spezifikation werden in die interne Darstellung überführt (konvertiert) und in der ESPRESO-

Datei gespeichert. Abschnitte, in denen Fehler aufgetreten sind, werden in eine gesonderte Datei gespeichert, die der Benutzer korrigieren und erneut konvertieren kann.

- Die Dekonvertierung

Für Änderungen in Spezifikationen, die bereits konvertiert sind und somit nur in interner Darstellung vorliegen, wird die interne Darstellung in ESPRESO-S-Form zurückgewandelt (dekonvertiert). Die dekonvertierten Teile der internen Darstellung werden in der ESPRESO-Datei gelöscht, so daß die rekonstruierte ESPRESO-Spezifikation mit Hilfe eines Editors geändert und erneut konvertiert werden kann.

- Prüfungen und Reports

Die interne Repräsentation dient als Grundlage für Prüfungen auf Vollständigkeit der Spezifikation und für verschiedenartige Reports. Reports und Prüfungen dienen der Kontrolle der Spezifikation durch den Anwender. Mit Hilfe der Reportfunktionen kann der Inhalt der ESPRESO-Datei unter verschiedenen Gesichtspunkten dokumentiert werden. Prüffunktionen sind zusätzlich notwendig, da ESPRESO die Reihenfolge der Präzisierung offenläßt und auch Spezifikationen akzeptiert werden, die inhaltlich unvollständig sind. Z.B. würde eine Prüffunktion, die die Vollständigkeit der Definition aller Objekte überprüft, feststellen, daß das Objekt "Filterdaten" in der vorhergehenden Spezifikation nur indirekt definiert ist.

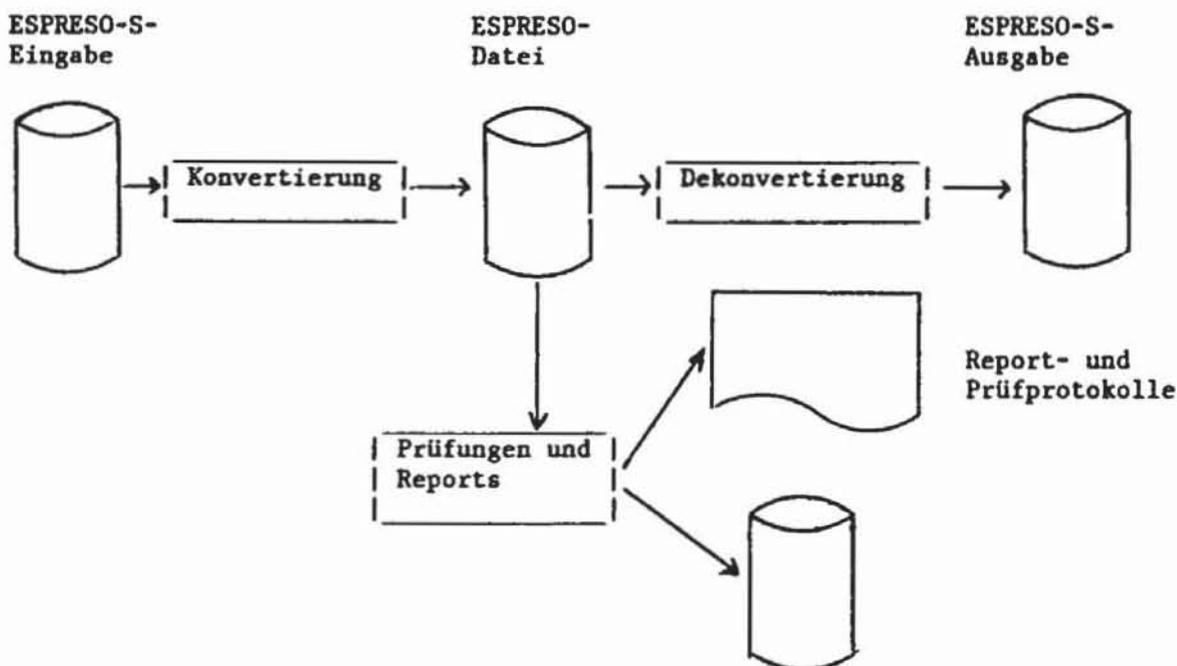


Bild 2: Übersicht über die Funktionen von ESPRESO-W

3.2 Programmstruktur von ESPRESO-W

ESPRESO-W besteht aus eigenständigen, jeweils modular aufgebauten Programmen. Für die Modularisierung des Gesamtsystems und der Teilsysteme wurden die beiden Entwurfsmethoden nach Parnas (1972) und Jackson (1975) angewandt. Beiden Entwurfsmethoden ist gemeinsam, daß in ihnen zuerst eine genaue Analyse der zu verarbeitenden Daten vorgeschlagen wird.

Bei der Entwurfsmethode nach Parnas sind die Zugriffsfunktionen auf die Daten von zentraler Bedeutung, wohingegen bei der Entwurfsmethode nach Jackson die Struktur der E/A-Daten im Mittelpunkt des Programmentwurfs steht. Die Entwurfsmethode nach Parnas eignet sich für die Verwaltung interner Daten. Dagegen ist die Entwurfsmethode nach Jackson für die Verarbeitung extern vorliegender Daten geeignet.

Die externen Daten sind:

- die ESPRESO-S-Eingabe bei der Konvertierung,
- die ESPRESO-S-Ausgabe bei der Dekonvertierung und
- die Report- und Prüfprotokolle.

Jede Funktion wird daher durch ein eigenständiges Programm realisiert, das wegen der Verarbeitung externer Daten nach der Jackson-Methode entworfen wurde.

Die internen Daten bestehen aus den Syntaxtabellen (siehe 3.3.3) und der ESPRESO-Datei ohne die Texte, die auf einen Sekundärspeicher ausgelagert werden. Beide werden jeweils von einem eigenständigen, nach der Parnas-Methode entworfenen Programm verwaltet. Jedes dieser Programme ist modularisiert, wobei ein Modul aus der Vereinbarung lokaler Daten und einer Menge von Prozeduren, die diese Daten verwalten, besteht. Andere Programme können auf diese Daten nicht direkt, sondern nur mit Hilfe der vorhandenen Prozeduren zugreifen.

Das Verbergen von Daten vor ihrer Umgebung ist durch den zunächst verwendeten PASCAL/360-Compiler (Kieburz, 1979) unterstützt worden. In PASCAL/360 ist eine Aufteilung der Programme in getrennt übersetzbare Moduln möglich, wobei innerhalb eines Moduls die Vereinbarung modullokaler Variablen zulässig ist. Die Schnittstellen eines Moduls nach außen sind die Zugriffsfunktionen, die er anderen Moduln zur Verfügung stellt. Zusätzlich können globale Variable vereinbart werden, durch die zwar das Prinzip des Verbergens von Daten durchbrochen wird, ohne die jedoch in der Praxis schwer auszukommen ist.

Die Programme von ESPRESO-W lassen sich zwei Schichten (Bild 3) zuordnen. Die Programme in Schicht I realisieren die Funktionen von ESPRESO-W. In Schicht II sind

die Programme eingeordnet, die interne Daten verwalten, so daß Aufrufe nur in der Richtung

Schicht I --> Schicht II

erfolgen. Oberhalb von Schicht I ist noch eine bei der ersten Implementierung nur virtuelle Schicht vorhanden, in der das Programm für die Bedienung der Programme in Schicht I liegt. Dieses startet die PASCAL-Programme und ordnet den im PASCAL-Programm vereinbarten internen Dateien externe Dateien zu.

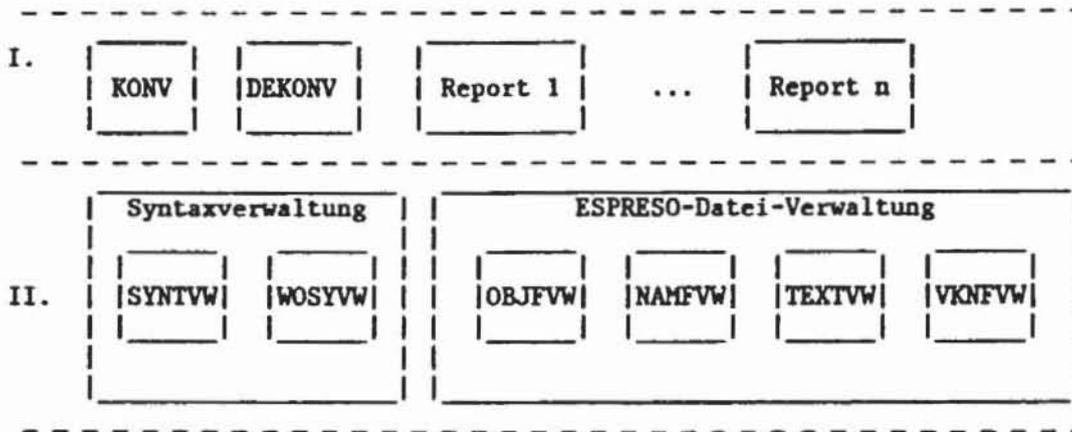


Bild 3: Schichtenstruktur der Programme von ESPRESO-W

3.3 Die Programme von ESPRESO-W

3.3.1 Das Programm ESPRESO-Datei-Verwaltung

Die ESPRESO-Datei ist aus folgenden vier Listen zusammengesetzt:

1. Liste für Objekte,
2. Liste für Objektnamen,
3. Liste für die Verwaltung der Texte,
4. Liste für Verknüpfungen.

Jede Liste wird von einem eigenen Modul verwaltet, der Zugriffsfunktionen für die jeweilige Liste bereitstellt.

Die logischen Verbindungen zwischen den Objekten und den Verknüpfungen, die durch die Kanten des bipartiten Graphen beschrieben werden, sind in der ESPRESO-Datei wie folgt dargestellt (Bild 4):

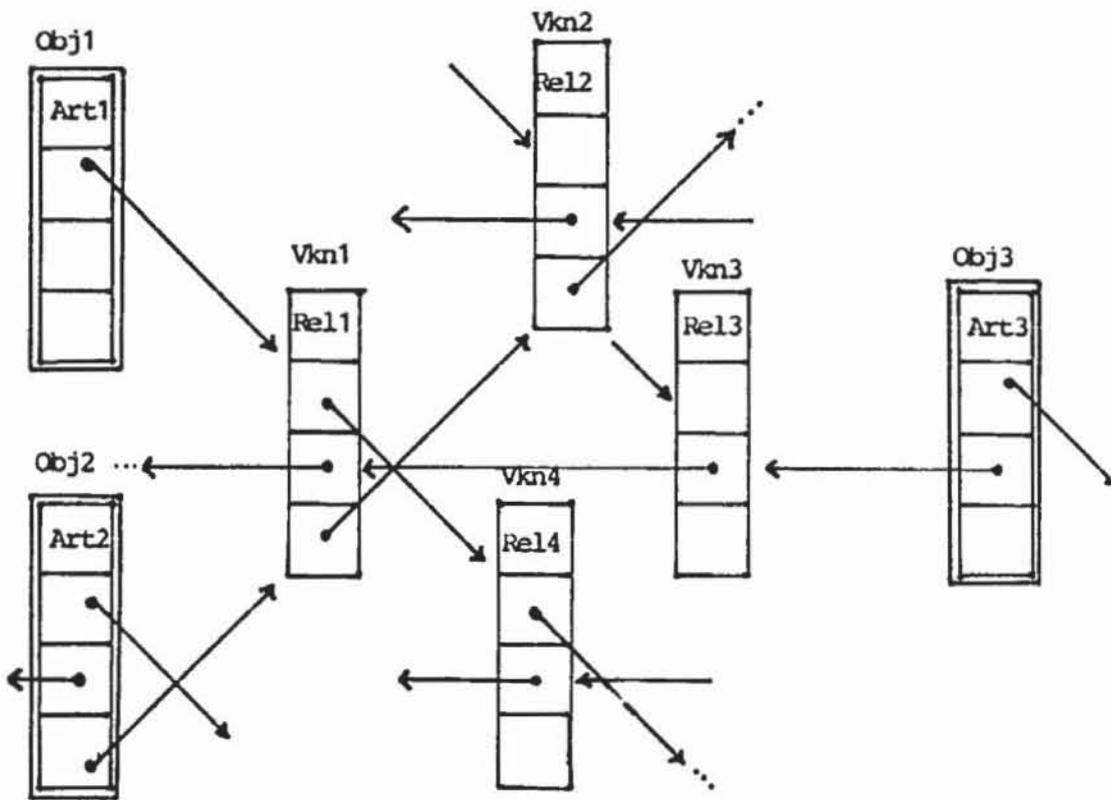


Bild 4: Schematische Darstellung der Verkettungen

Obj1, Obj2 und Obj3 sind durch Vkn1 verknüpft. Wie man sieht, gehen von jedem Objekt drei Ketten aus, die erste für diejenigen Verknüpfungen, in denen das Objekt Komponente 1 ist, die beiden anderen entsprechend für das Auftreten als Komponente 2 bzw. 3. In den Verknüpfungen werden die Ketten jeweils fortgesetzt. Die Verweisketten enden mit einem speziellen leeren Verweis. Nicht dargestellt sind die Verzeigerungen mit Namens- und Textliste (nur für Objekte, Verknüpfungen tragen keine Namen!) sowie die Zeiger, die von jeder Verknüpfung auf die Komponenten weisen.

3.3.2 Das Programm KONV

KONV führt die Konvertierung durch. Für KONV sind folgende Teilaufgaben festgelegt:

1. Lexikalische Analyse der ESPRESO-S-Eingabe.
2. Zerteilung der ESPRESO-S-Eingabe und Steuerung der Zugriffoperationen auf die ESPRESO-Datei.
3. Durchführung der notwendigen Konsistenzprüfungen. Zu diesen zählen z.B. Prüfung der Modulhierarchie, Prüfung der Baumstruktur für Blöcke oder Prüfung der Typ- und Variablendefinitionen auf Zyklenfreiheit.

Alle Aufgaben werden in einem Durchlauf erledigt, so daß keine temporären Daten

anfallen. Da dadurch auch solche Daten in die ESPRESO-Datei gelangen, die beim Auftreten eines Fehlers hinfällig werden und wieder gelöscht werden müssen, ist es notwendig, die Neueinträge speziell zu markieren. Bei der Fehlerbehandlung ist zunächst eine recht grobe Strategie angewandt worden. Beim Auftreten eines Fehlers in einem Abschnitt wird der Rest des Abschnittes überlesen und die Konvertierung erst mit dem nächsten Abschnitt fortgesetzt. Das Auftreten von Folgefehlern wird dadurch weitgehend vermieden.

Für die Zerteilung wird folgendes Verfahren ausgewählt (Bild 5):

- für die globale Syntax und die Syntax der Texte wird eine top-down-Zerteilung vorgenommen. Diese ist aufgrund der LL(1)-Eigenschaft von ESPRESO-S sehr einfach.
- für die Zerteilung der Angabensyntax wird ein tabellengesteuertes Verfahren angewandt, wobei die Tabellen so aufgebaut sind, daß sie sowohl vom Speicherplatzbedarf her günstig sind, als auch bei der Konvertierung und Dekonvertierung benutzt werden können.

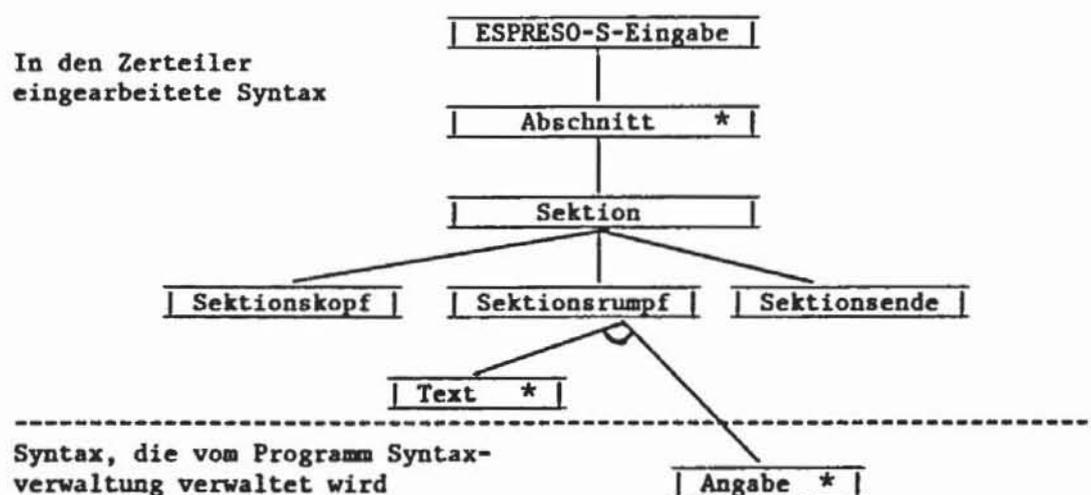


Bild 5: Struktur der in den Zerteiler eingearbeiteten globalen Syntax (Jackson-Design-Methode)

Eine Kante zu einem Kästchen ohne "*" hat die Bedeutung "wird einmal ausgeführt". Eine Kante zu einem Kästchen mit "*" hat die Bedeutung "wird nicht, einmal oder mehrmals ausgeführt". Mehrere durch einen Kreisbogen verbundene Kanten bedeuten "wird alternativ ausgeführt".

3.3.3 Das Programm Syntaxverwaltung

Das Programm Syntaxverwaltung verwaltet die kontextfreie und die kontextsensitive Syntax der Angaben. Die Angabensyntax wird vom Programm SYNTCH vorverarbeitet und in komprimierter Form auf eine externe Datei geschrieben. Vor der Konvertierung und

Dekonvertierung wird dann die komprimierte Form vom Programm Syntaxverwaltung eingelesen.

Die Angabensyntax besteht im wesentlichen aus folgenden Tabellen:

- einer Wortsymboltabelle mit allen in ESPRESO-S vorkommenden Wortsymbolen.
- einer Relationentabelle, in der für jede Relation die zulässigen Arten der verschiedenen Komponenten festgelegt sind.
- einer Syntaxtabelle, in der die kontextfreie Syntax der Angaben abgelegt ist. Zusätzlich enthält die Syntaxtabelle Informationen über Konsistenzprüfungen, die vor dem Eintragen einer Verknüpfung durchgeführt werden müssen.

3.3.4 Das Programm DEKONV

DEKONV reproduziert eine Spezifikation in ESPRESO-S-Form aus dem Inhalt der ESPRESO-Datei. Die Reproduktion kann entweder die gesamte Spezifikation umfassen, oder es können selektiv Objekte ausgewählt werden, die mit allen ihren Verknüpfungen dekonvertiert werden. Die rekonstruierte ESPRESO-Spezifikation ist so aufgebaut, daß in ihr die schrittweisen Vervollständigungen in sinnvoller Reihenfolge geordnet ausgegeben werden. Die Programmstruktur entspricht weitgehend der von KONV und wird daher hier nicht näher erläutert.

4. Stand der Arbeit (Februar 1981)

ESPRESO-W wurde teilweise im Rahmen einer Diplomarbeit entworfen und implementiert (Eckert, 1980). Die Codierung wurde zunächst in der Programmiersprache PASCAL/360 auf der Rechenanlage IBM 3033 vorgenommen. Da sich dieser Compiler als unbrauchbar erwies, wird das System derzeit auf die Anlage Siemens R30 in Standard-PASCAL übertragen und installiert. Die Übertragung hat Schwierigkeiten bereitet, da der Siemens PASCAL-Compiler nicht die Vereinbarung modullokaler und globaler Variablen zuläßt, so daß alle Informationen, die eine modulexterne Funktion benötigt, in der Parameterliste der Funktion übergeben werden müssen.

Die Programme in Schicht II und das Programm KONV sind vollständig implementiert und getestet. Das Programm DEKONV wird demnächst installiert, so daß ab März 1981 ein brauchbares Minimalsystem zur Verfügung steht. Die Report- und Prüffunktionen werden im Laufe des Jahres 1981 erstellt.

ESPRESO wird im Kernforschungszentrum Karlsruhe für die Spezifikation eines Reaktorschutzrechnersystems eingesetzt.

5. Literaturangaben

- Boehm, B.W. (1976):
Software engineering.
IEEE Trans.Comp., C-25, S. 1226-1241.
- Eckert, K. (1980):
Implementierung eines Spezifikationssystems
für Prozeßrechner-Software.
Diplomarbeit, Universität Karlsruhe.
- Jackson, M. (1975):
Principles of program design.
Academic Press, London, New York.
- Kieburtz, R.B., W. Barabash, C.R. Hill (1979):
STONY BROOK PASCAL/360, User's Guide - Release 2 -
State University of New York at Stony Brook.
- Knuth, D.E. (1971):
Top-down syntax analysis.
Acta Informatica, 1.
- Ludewig, J., W. Streng (1978):
Überblick und Vergleich verschiedener Mittel
für die Spezifikation und den Entwurf von Software.
KfK 2506.
- Ludewig, J. (1980):
PCSL - a process control software specification system.
KfK 2874.
- Ludewig, J. (1981a):
Zur Erstellung der Spezifikation von Prozeßrechnersoftware.
Dissertation, TU München; erscheint ebenfalls als KfK 3060.
- Ludewig, J. (1981b):
PCSL und ESPRESO - zwei Ansätze zur Formalisierung
der Prozeßrechner-Softwarespezifikation.
in R. Baumann (Hrsg.);
GI Fachtagung Prozeßrechner (in Vorbereitung).
- Parnas, D.L. (1972):
A technique for software module specification with examples.
CACM Vol.15, No.5, S. 330-336.
- Teichroew, D., E.A. Hershey III (1977):
PSL/PSA: a computer-aided technique for structured
documentation and analysis of information processing systems.
IEEE Trans. Software Eng., SE-3, 41-48.
- Watt, D.A., O.L. Madsen (1977):
Extended attribute grammars.
Report 10, University of Glasgow.