A n w e n d e r h a n d b u c h

- N P S -

Normiertes Programmier-System

VEB Baumwollspinnerei
und Zwirnerei Leinefelde
- ORZ -

Die vorliegende Anwender-Dokumentation entspricht dem Stand vom 15. 4. 1974 und unterliegt dem Änderungsdienst.

Die Ausarbeitung erfolgte durch ein Kollektiv des VEB Baumwollspinnerei und Zwirnerei Leinefelde.

Herausgeber:
VEB Baumwollspinnerei und Zwirnerei Leinefelde
- ORZ -
56 Leinefelde
Birkunger Straße

Diese Schrift enthält die Anwenderdokumentation für das Programmiersystem -NPS-.

Die Schrift gliedert sich sachlich in zwei Teile. Im Einführungsteil (Kapitel 1 bis 3), der sich an alle Anwender von EDVA der 3. Rechnergeneration wendet, werden allgemeine Prinzipien, Möglichkeiten und Vorteile des Systems -NPS-, seine Methodik und sein Aufbau beschrieben.

Für das Verständnis des Einführungsteiles genügen allgemeine Kenntnisse über Datenverarbeitungsprojektierung und -programmierung sowie über Inhalt und Struktur von Plattenbetriebssystemen.

Im Anwendungsteil werden dem -NPS-Nutzer exakte Angaben über die Handhabung des Systems -NPS- im DOS/ES, über die -NPS-Makros und über die mit ihnen realisierbaren Funktionen gegeben und anhand von Beispielen illustriert.
Die Lieferung des Anwendungsteiles erfolgt nur nach vertraglicher Vereinbarung.

Für das Verständnis des Anwendungsteiles ist die Kenntnis des Einführungsteiles notwendig. Ferner sind spezielle und detaillierte Kenntnisse des Betriebssystems DOS/ES bzw. analoger Betriebssysteme sowie mindestens einer der Programmiersprachen PL/I und Assembler erforderlich.

Literaturverzeichnis

/1/ Bär, D.
Bilden wir auf dem Gebiet der Programmierung richtig aus?
RT/DV 10 (1973) 7, S. 13 - 15

/2/ Meyer, F. A.
Normierte Programmierung
BTA 11 (1970) 8, S. 484 - 493

/3/ Helm, B.
Neue Wege in der Datenverarbeitung:
NORMIERTE PROGRAMMIERUNG
Computer-Praxis (1970) 12, S. 230 - 237

/4/ Seyd, W., Franke, H.
Anwendung der Normierten Programmierung
RT/DV 11 (1974) 1, S. 25 - 31

Inhaltsverzeichnis

1. Einleitung

Programmierung ist ein in sich geschlossener und zusammenhängender Komplex aus

- Programmplanung
- Programmcodierung
- Programmtestung
- Programmerprobung
- Programmdokumentation
- Programmpflege und
- Organisation der Programmnachnutzung.

Aus diesem Komplexcharakter ergeben sich folgende Prinzipforderungen /1/ an ein "gutes" EDV-Programm:

- logisch begründete, klar übersehbare und problembezogene Gliederungen des Gesamtprogramms in Teilprogramme

- übersichtliche und verständliche Programmierung der Komponenten, Vermeidung von "Programmtricks"

- Möglichkeiten, den Test- und Dokumentationsaufwand entscheidend zu senken

- schnelle Einarbeitungsmöglichkeit in das Programm für fremde Programmierer

- leichte Erweiterungsfähigkeit, Anpaßbarkeit an spezielle Bedürfnisse sowie Kombinierbarkeit mit anderen Programmen.

Mit dem - Normierten Programmier-System - (-NPS-) wird dem Anwender von EDVA der 3. Rechnergeneration ein System zur Realisierung dieser Forderungen angeboten.

-NPS- enthält sowohl eine Methodik zur Rationalisierung der Programmplanung als auch ein System von Komponenten zur Rationalisierung und Intensivierung der Programmierung.

-NPS- ist eine Ergänzung der für die Anlagen der 3. Rechnergeneration vorhandenen Betriebssysteme, arbeitet auf deren Basis und verwendet sowohl eigene als auch Systemkomponenten aller drei Bibliotheksebenen.

-NPS- ermöglicht

- eine wesentliche Reduzierung der für Planung, Codierung, Testung, Erprobung, Dokumentierung und Pflege eines Programmes notwendigen Arbeitszeit

- eine relevante Senkung der mit den angegebenen Phasen verbundenen Kosten, insbesondere der Kosten für Programmtestung

- eine kurzfristige und problemlose Änderung mit -NPS- realisierter Programme bei organisatorischen Problemänderungen bzw. maschinentechnischen Ausfällen im Gerätesystem der EDVA

- eine Unterteilung größerer Programme und die Aufteilung an ein Bearbeiterkollektiv

- eine Vereinfachung der Programmdokumentation und deren Vereinheitlichung

- eine Verbesserung der Übersichtlichkeit und Sicherheit eines Programms durch klare Trennung der Programmablaufsteuerung von dem eigentlichen Verarbeitungsteil

- eine einfache und einheitliche Handhabung von Programmen durch eine festgelegte Hauptspeicheraufteilung und standardisierte Bedienerverständigung.

Methodik und Handhabung des Systems -NPS- sind leicht und schnell erlernbar und vom konkreten Anlagentyp unabhängig.

-NPS- unterstützt die Programmiersprachen Assembler und PL/I, wobei bereits bei Kenntnis einer dieser Sprachen eine Anwendung möglich ist.

-NPS- wird in dieser Schrift für das Betriebssystem DOS/ES beschrieben. Eine Übertragung auf adäquate oder analoge Betriebssysteme ist vollinhaltlich möglich.

Neben den beschriebenen prinzipiellen Möglichkeiten bietet das System -NPS- eine Palette weiterer organisatorischer und funktioneller Möglichkeiten. Dazu gehören unter anderem

- die Möglichkeit zur getrennten Übersetzung von Steuerteil und Verarbeitungsteil

- die Einsparung von Bibliothekskapazität durch nur

einmalige Katalogisierung universeller Steuer- und Verarbeitungsteile

- das Bereitstellen eines einheitlichen Programmverständigungsbereiches

- das Bereitstellen zusätzlicher Schalter, Werte und Möglichkeiten zur Beeinflussung bzw. Erkennung der Programmablaufsteuerung

- Möglichkeiten für

  - sequentielle Lochbandeingabe
  - aufbereitete Listenausgabe auf Magnetband
  - Steuerkarteneingabe (wahlweise über SYSIPT oder SYSLOG)
  - universelle Datensatzprüfungen (Reihenfolge, Alphazeichen, Prüfziffernkontrolle, Intervallkontrolle)
  - Verarbeitungsaufspaltung auf Grund automatisch erkannter Satzkennzeichen.

Neben der Anwendung zur Rationalisierung der Eigenprogrammierung ist -NPS- auch für die Realisierung der Anwenderteile bei Nutzung der sachgebietsorientierten Systemunterlagen sowie zum Aufbau allgemeiner Datenübernahme- und Datenprüfprogramme einsetzbar.

Die Komponenten des Systems -NPS- lassen sich in zwei Gruppen einteilen.
Die erste Gruppe beinhaltet die Komponenten zur Anpassung an das spezielle Problem, die zweite Gruppe enthält alle -NPS-Komponenten zur Steuerung und Verarbeitung.
Kommunikation zwischen dem System -NPS- und dem -NPS-Anwender besteht ausschließlich über maximal sieben Komponenten der ersten Gruppe. Sie erfolgt über eine problembezogene Parameterspezifikation. Die Erzeugung von Bezugnahmen und der spätere Auf- bzw. Abruf aller anderen Komponenten wird vom System -NPS- selbst im Laufe der Programmerzeugung bzw. -abarbeitung organisiert und vorgenommen.

Die vollständige Zusammenstellung aller -NPS- Komponenten ist dem Anhang A dieser Schrift zu entnehmen.

Mit Hilfe von -NPS- erzeugte Programme bestehen aus mehreren Teilen. Bild 1-1 gibt eine Übersicht über diese Teile und die von ihnen realisierte Hauptspeicherbelegung während der Programmabarbeitung.

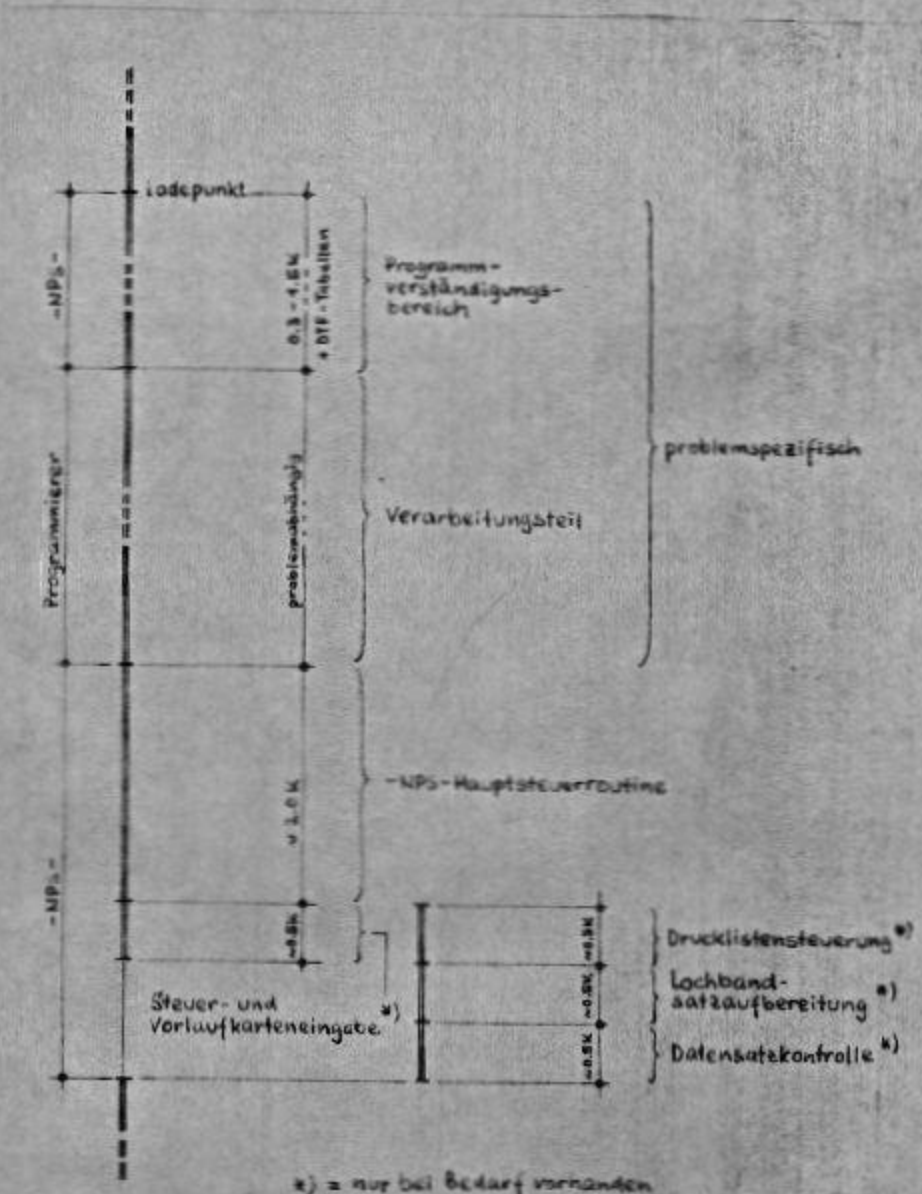


Bild 1-1. Programmteile und Hauptspeicherbelegung

2. -NPS-Methodik

2.1. Allgemeines

Methodische Basis des Systems -NPS- ist das organisatorische Prinzip der Normierten Programmierung /2/,/3/,/4/.
Anwendungsbereich der Normierten Programmierung ist das Gebiet der kommerziellen Datenverarbeitung. Normierungsgegenstände der Methode der Normierten Programmierung sind

- Programmstruktur
- Programmsteuermechanismus
- Programmablauffolge.

Dabei werden folgende wesentliche Normierungsziele angestrebt:

- Trennung von Programmablaufsteuerung und problemspezifischer Informationsverarbeitung
- vereinheitlichte Programmablaufsteuerung unter Verwendung universeller Steuermechanismen.

2.2. Programmstrukturierung

2.2.1. Grundprinzipien kommerzieller Programme

Unter dem Begriff "kommerzielle Programme" werden in dieser Schrift alle Aufgabenstellungen der ökonomischen Datenverarbeitung verstanden, die sich wie folgt abgrenzen lassen:

Die zu verarbeitenden Informationen sind zu logischen Sätzen zusammengefaßt. Logische Sätze sind durch Bedeutung, Anordnung und Länge der in ihnen enthaltenen Informationen definiert. Sie werden durch Satzkennzeichen identifiziert. Die Menge der in einem logischen Satz enthaltenen Informationen läßt sich in der Regel in Verarbeitungsinformationen und Zuordnungsinformationen gliedern.

Jeder logische Satz ist mit Hilfe einer primären Zuordnungsinformation (Ordnungsbegriff, Schlüssel) direkt einem bestimmten Objekt zugeordnet und widerspiegelt durch die Menge der Werte oder Zustände der im Satz enthaltenen Verarbeitungsinformationen einen sachlich abgegrenzten Zustand dieses Objektes.

Sätze mit gleichem primären Zuordnungsbegriff bilden eine Gruppe.
Neben der primären Zuordnung eines logischen Satzes erfolgen im allgemeinen weitere, in der Regel hierarchisch gegliederte Sekundärzuordnungen. So ist zum Beispiel ein Satz, der Lohninformationen eines Beschäftigten enthält, primär dem Werktätigen selbst zugeordnet. Sekundäre Zuordnungen bestehen innerhalb eines Betriebes beispielsweise

- zur Kostenstelle, der der Werktätige angehört
- zum Betriebsteil, der die Kostenstelle enthält
- zum Gesamtbetrieb als Menge aller Betriebsteile.

Diese Zuordnungskettung ist problemabhängig, d. h. in einem anderen Verarbeitungsprojekt kann für einen bestimmten Satz eine vollkommen andere Zuordnungshierarchie bestehen. Sekundärzuordnungen können direkt über ebenfalls im Satz enthaltene Zuordnungsbegriffe oder indirekt über das Kettprinzip anhand der Primärzuordnung aus einer Tabelle (Stammdatei, Vorlaufdatei) erfolgen. Häufig sind jedoch bereits die Zuordnungsinformationen (Schlüssel) hierarchisch aufgebaut und gegliedert.

Bezüglich der Sekundärzuordnungen werden ebenfalls Gruppenzusammenfassungen vorgenommen. In der Regel werden den Datenverarbeitungsprojekten Zuordnungshierarchien zugrundegelegt. Alle bestehenden Zuordnungen können entsprechend über ein hierarchisch aufgebautes Gruppenzuordnungsfeld (Gruppenbegriffsfeld) deutlich gemacht werden. Das Gruppenzuordnungsfeld enthält von rechts nach links hierarchisch aufsteigend so viele Teilfelder, wie Zuordnungsebenen existieren. Bild 2-1 zeigt den Aufbau des Gruppenzuordnungsfeldes und definiert den im folgenden oft verwendeten Begriff der Gruppenstufen.

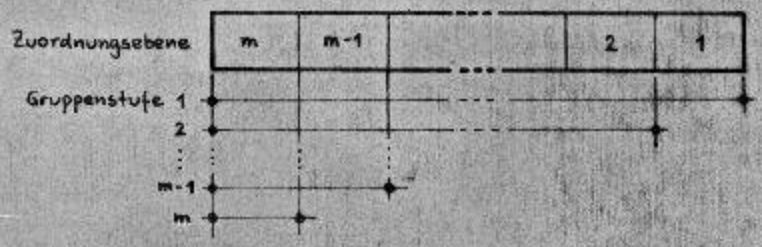


Bild 2-1. Gruppenzuordnungsfeld mit Gruppenstufen

Im Sinne der gegebenen Definitionen gehören also zur Gruppenstufe i (i=1(1)m) alle die Sätze, die in den Teilfeldern von i bis m gleiche Zuordnungsbegriffe enthalten.

Die zu verarbeitenden Sätze entstammen einer oder mehreren (Eingabe-) Dateien. Die Reihenfolge der Sätze in den Dateien wird durch eine der problembezogenen Zuordnungshierarchie entsprechende Sortierung der Sätze über die in ihnen enthaltenen Zuordnungsbegriffe vorgegeben. Aus diesem Grund ist die Anordnung der Zuordnungsfelder in allen Sätzen einer Datei deckungsgleich.

Die Verarbeitung der über die Eingabedateien satzweise einlaufenden Informationen erfolgt in einer programmgesteuerten Reihenfolge dateibezogen gemischt einzeln pro identifizierter Satzart (Einzelverarbeitung) und zusammengefaßt pro Gruppenstufe nach Abarbeitung aller zu dieser Gruppenstufe gehörenden Sätze (Gruppenwechselverarbeitung).
Auf Grund der Sortierreihenfolge der Sätze in den Dateien erfolgen Gruppenstufenwechsel ebenfalls in geordneter Folge.

Die vorstehend gegebene theoretische Abgrenzung kommerzieller Programme umfaßt ein außerordentlich breites Spektrum spezieller Anwendungen aus allen Bereichen der ökonomischen Datenverarbeitung. Trotz der Vielfalt der vorkommenden und möglichen Verarbeitungsalgorithmen der derart abgegrenzten Aufgabenklasse kann allen denkbaren Programmen dieser Klasse ein unifizierter, "natürlicher" Programmablauf zugrundegelegt werden. Dabei wird folgende zeitliche und sachliche Struktur deutlich:

- Übernahme von Steuerinformationen, problemabhängige Anfangswertzuweisung, Programmanfangsroutinen,

- dateiweises Übernehmen bzw. Nachziehen der zu verarbeitenden Datensätze in Eingabebereiche,

- Auswahl des in der Verarbeitungsfolge nächsten Satzes aus der Menge aller anliegenden Sätze der Eingabebereiche,

- Zuordnungsanalyse des ausgewählten Satzes, bei Gruppenwechsel in einer oder mehreren Gruppenstufen, Durchführung der entsprechenden Gruppenwechselverarbeitung(en) und Ausgaben,

- Satzidentifizierung und Einzelverarbeitung entsprechend der Satzart,

- Wiederholung der Aktionen 2 bis 5 bis zum Dateiende aller vorliegenden Eingabedateien,

- Programmendebehandlung.

Eine Variation der angegebenen allgemeinen Struktur eines kommerziellen Programms erfolgt im konkreten Fall im wesentlichen nur durch die Anzahl und durch die Speicherungs- und Verarbeitungsform der zu verarbeitenden Datenmengen.

2.2.2. Blockeinteilung

Im System der Normierten Programmierung wird die im vorhergehenden Punkt umrissene natürliche Struktur zur Grundlage einer nach logischen und funktionellen Gesichtspunkten erfolgenden Blockeinteilung (Bild 2-2).
Jeder Block realisiert eine klar umrissene Teilaufgabe und ist in sich geschlossen. Er hat in der Regel genau einen Eingang und einen Ausgang. Eine Unterteilung der Blöcke in lineare Unterblöcke erleichtert die Blockbeschreibung und das Verständnis der Blockfunktion.
Jeder Block wird durch einen zugeordneten Buchstaben gekennzeichnet.
Blockkennzeichnung und das im Bild 2-2 gezeigte Schema der Blockeinteilung wurden für das System -NPS- aus /3/ übernommen.

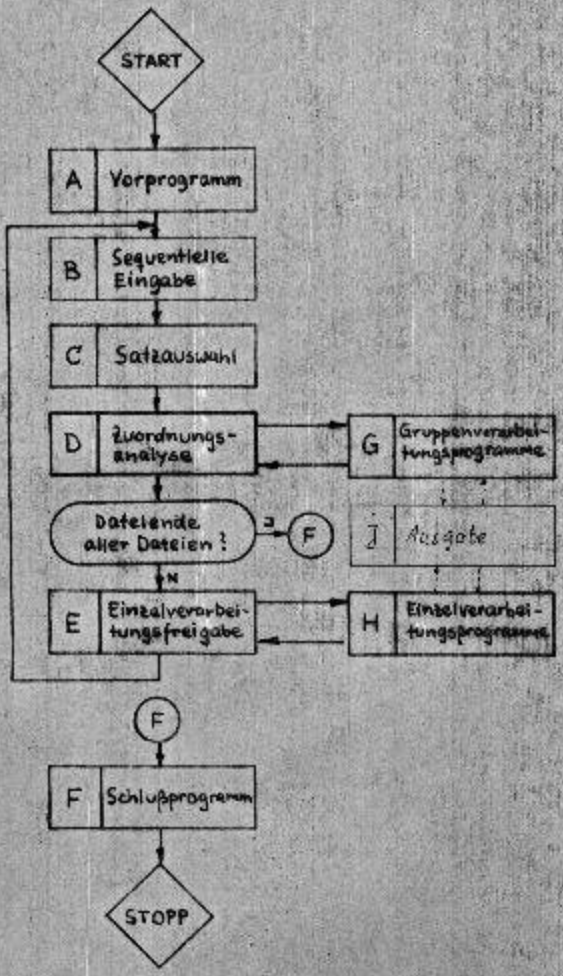


Bild 2-2. Blockeinteilung und Programmablauf

2.3. Programmsteuerung

Wird durch die standardisierte Blockstruktur der Normierten Programmierung erreicht, daß logisch und funktionell zusammengehörende Teile eines Programmes stets an der gleichen Stelle zu finden sind, so sorgt eine sinnvoll organisierte und in ihrem Mechanismus schematisierte Programmablaufsteuerung für die Regelung der problem- und datenabhängigen Verarbeitungsreihenfolge im Programm.

2.3.1. Datei-Status und Datei-Priorität

Grundlage der Verarbeitungslogik der Normierten Programmierung ist das Prinzip der satzweisen Verarbeitung. Als Eingabedateien werden alle sequentiell zu verarbeitenden Dateien zugelassen.
Außer den sequentiellen Eingabedateien in einem Programm noch vorhandene Eingabedateien mit wahlfreiem Zugriff werden in der Normierten Programmierung nicht als Eingabedateien, sondern als (extern verwaltete) Tabellen betrachtet und behandelt.

Jeder sequentiellen Eingabedatei (im folgenden kurz auch als -NPS-Datei bezeichnet) werden bei der Normierten Programmierung zwei Informationen zugeordnet, und zwar

erstens der Datei-Status S und
zweitens die Datei-Priorität D.

Der Datei-Status gibt für jede -NPS-Datei Auskunft über deren augenblicklichen Zustand. Im System der Normierten Programmierung werden folgende vier Zustände unterschieden:

S=0 : Von der Datei wird ein Satz eingegeben.

S=1 : Von der Datei wird kein Satz eingegeben.

S=2 : Die Datei ist abgeschlossen.

S=3 : Die Datei ist nicht vorhanden.

Bei der Normierten Programmierung sollte der Programmierer bei der Programmerstellung von einer organisatorischen Maximalvariante bezüglich der angeschlossenen Eingabedateien ausgehen und das Programm dafür auslegen.
Das System -NPS- ermöglicht es, durch Bedieneraktion oder über Steuerkarte unmittelbar bei Programmabarbeitung die nicht vorhandenen Dateien durch Veränderung ihres Datei-Status auf den Wert

3 für den anliegenden Verarbeitungslauf unwirksam zu machen, ohne die Funktion des Gesamtprogramms bezüglich der angeschlossenen Dateien zu zerstören.

Die Datei-Priorität steuert die Reihenfolge der Satzzufuhr von den Dateien, wenn sich diese nicht aus anderen Merkmalen herleiten läßt.
Die Datei-Priorität wird durch den Programmierer bei der Programmierung vergeben, kann im System -NPS- aber auch noch unmittelbar bei Programmabarbeitung von außen variiert werden.
Die Datei-Priorität wird durch eine natürliche Zahl festgelegt. Die (bei gleichen Zuordnungsbegriffen) zuerst zu bearbeitende Datei erhält die Datei-Priorität D=1, die zweite die Datei-Priorität D=2 und so weiter. Gleiche Datei-Prioritäten sind zwar zulässig, widersprechen aber der inhaltlichen Funktion dieses Steuerelements. Bei der Vergabe der Datei-Priorität an UPDATE-Dateien ist zu beachten, daß die Eingabesätze dieser Dateien erst nach der Verarbeitung aller übrigen zur gleichen Gruppe gehörenden Sätze zum Zurückschreiben freigegeben werden dürfen, weil sie von allen anderen Sätzen her eine Veränderung erfahren können. UPDATE-Dateien sind also jeweils die höchsten Nummern (=niedrigste Prioritäten) zuzuordnen.

2.3.2. Gruppenbegriffe

Bei einer sequentiellen Eingabe wird eine Datei nach bestimmten Sortier-Gruppiermerkmalen bearbeitet. Da diese Merkmale (Zuordnungsbegriffe, nach denen die Datei sortiert vorliegt) in unterschiedlichen Sätzen auch in unterschiedlicher Folge und Anordnung erscheinen, werden in der Normierten Programmierung für alle sequentiellen Eingabedateien nach einem (problembezogen) einheitlichen Schema sogenannte Gruppenbegriffsfelder aufgebaut. Dazu werden aus dem pro Datei eingelesenen bzw. nachgezogenen Satz alle für das Problem relevanten Ordnungsbegriffe (Schlüssel) entnommen und entsprechend der m-stufigen Gruppenhierarchie aufsteigend von rechts nach links in m entsprechend breite Teilfelder des Gruppenbegriffsfeldes über-

tragen. Das so gebildete Gruppenbegriffsfeld enthält folglich den zusammengesetzten, "vollständigen" Gruppenbegriff des jeweiligen Satzes.
Die Länge des Gruppenbegriffsfeldes ergibt sich aus der Summe der Längen der einzelnen Teilschlüssel. Gliederung und Struktur der Gruppenbegriffsfelder sind für alle Dateien gleich, Unterschiede ergeben sich beim Aufbau auf Grund der im allgemeinen unterschiedlichen Anordnung der Schlüssel in den Sätzen unterschiedlicher Dateien. Innerhalb einer Datei muß auch bei mehreren Satzarten anordnungsmäßige Deckungsgleichheit der Sortier-Gruppiermerkmale bestehen!

Im Bild 2-3 wird ein Beispiel für den Aufbau eines (dateibezogenen) Gruppenbegriffsfeldes angegeben.

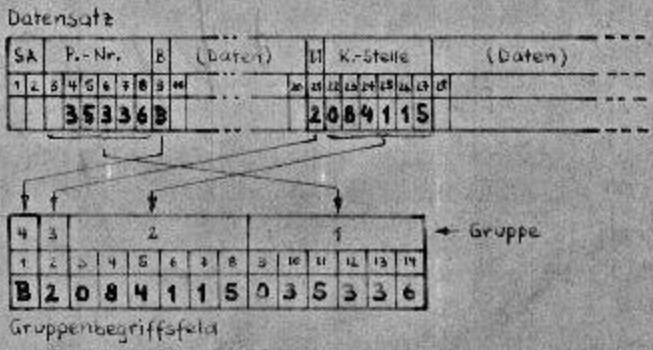


Bild 2-3. Bildung eines (dateibezogenen) Gruppenbegriffsfeldes (Beispiel)

2.3.3. Statuswörter

Als zentraler Steuerungsmechanismus im System der Normierten Programmierung werden die sogenannten (-NPS-) Statuswörter verwendet. Diese Statuswörter dienen zur normierten Weitergabe von Informationen über den augenblicklichen Verarbeitungszustand im Programm zwischen den einzelnen Blöcken. Es existieren zwei Arten von Statuswörtern, und zwar

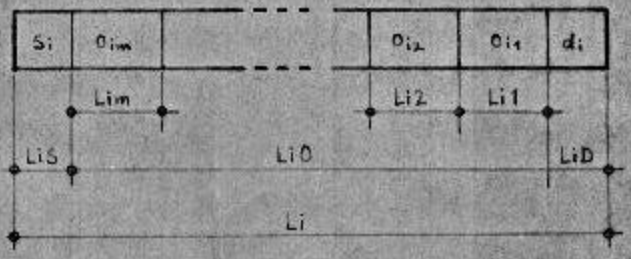
erstens die dateibezogenen Statuswörter und
zweitens die dateineutralen Statuswörter.

Dateibezogene Statuswörter werden für jede im Programm enthaltene sequentielle (also -NPS-) Eingabedatei angelegt. Sie enthalten für die jeweils betrachtete Datei von rechts nach links

- die Datei-Priorität (1 Byte)
- das Gruppenbegriffsfeld (Länge abhängig von Teilschlüssellängen)
- den Datei-Status (1 Byte).

Der Aufbau aller dateibezogenen Statuswörter ist identisch. Sie werden mit Li bezeichnet, wobei i die Nummer der Eingabedatei ist (i=1,2,...,n). L1 ist also das Statuswort der (Eingabe-) Datei 1, L2 das der Datei 2 usw.

Im Bild 2-4 ist die weitere Untergliederung der dateibezogenen Statuswörter und die Bezeichnung der Teilfelder angegeben.



Teilfeld	Inhalt
LiS	s_i = Status der Datei i
LiO	= Gruppenbegriff der Datei i mit den Teilen:
Li1	O_{i1} = Zuordnungsschlüssel zur Zuordnungsebene 1
Li2	O_{i2} = ——— " ——— 2
⋮	⋮ ⋮
Lim	O_{im} = ——— " ——— m
LiD	d_i = Priorität der Datei i

Bild 2-4. Aufbau eines dateibezogenen Statuswortes

Neben den n dateibezogenen Statuswörtern gibt es die beiden dateineutralen Statuswörter LA und LN. Die Struktur dieser Wörter stimmt stets mit der der dateibezogenen Statuswörter überein. Die Definitionen der Teilfelder der dateineutralen Statuswörter sind jedoch aus Steuerungsgründen anders gewählt (Bild 2-5).

Das dateineutrale Statuswort LA enthält im Laufe der Programmabarbeitung das dem zuletzt verarbeiteten Satz zugeordnete "alte" Statuswort, das Statuswort LN enthält das entsprechende "neue" Statuswort des für die Verarbeitung freigegebenen Satzes.
Die Untertoilung der dateineutralen Statuswörter LA und LN in die Teilfelder lt. Bild 2-5 ermöglicht ein einfaches Feststellen von Gruppenwechseln. Stimmt z. B. das LN2 nicht mit dem LA2 überein, so ist ein Wechsel in der 2. Gruppenstufe erfolgt. Für alle anderen Gruppenstufen gilt das analog.

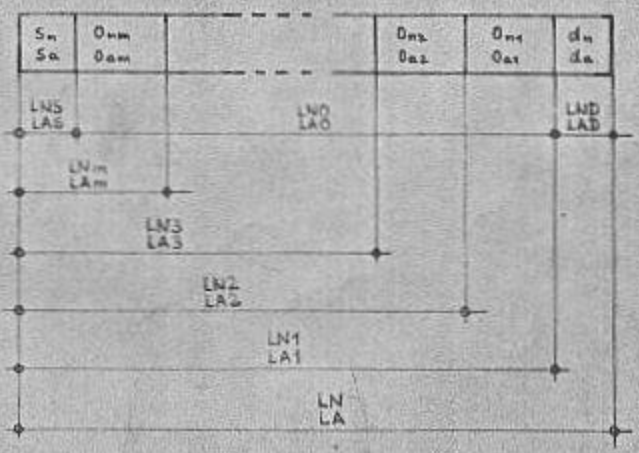


Bild 2-5. Aufbau der dateineutralen Statuswörter

2.3.4. Schalter

Im System der Normierten Programmierung sind zwei Gruppen von Indikatoren bzw. Schaltern von besonderer Bedeutung für die Programmsteuerung.

Die Indikatoren QLj (j=1,2,...,m entsprechend der Gruppenstufe) werden vom Programm gestellt, wenn der neu zur Verarbeitung anliegende Satz einen Gruppenwechsel in der entsprechenden Gruppenstufe ausgelöst hat, d. h. wenn der zu verarbeitende Satz der erste einer neuen Gruppe in dieser Gruppenstufe ist.
Sie können bei der Einzelverarbeitung des Satzes durch den Programmierer ausgewertet werden, um bestimmte Aktionen auszulösen. Nach Abschluß der Einzelverarbeitung werden gesetzte QLj vom Programm wieder zurückgesetzt.

Die Schalter QGj (j=1,2,...,m entsprechend der Gruppenstufe) geben dem Programmierer die Möglichkeit zur Steuerung der Ausführung der Gruppenverarbeitungsblöcke G1 bis Gm. Sie müssen vom Programmierer gestellt werden. Die Auswertung und das Rücksetzen erfolgt vom Programm. Ein anstehendes Gruppenverarbeitungsprogramm, also ein auf Grund eines erfolgten Gruppenwechsels angesprungener Gj-Block, wird nur dann abgearbeitet, wenn vorher der entsprechende QGj-Schalter auf 1 gestellt wurde. Andernfalls wird der Gj-Block übergangen.

2.4. Programmablauf

Mit den eingeführten Bezeichnungen und Steuerungselementen läßt sich der Programmablauf in allen gemäß Bild 2-2 strukturierten Programmen bei Verwendung der Prinzipien der Normierten Programmierung folgendermaßen präzisieren:

2.4.1. A-Block

Der A-Block enthält alle einmalig vor Beginn der eigentlichen Verarbeitung notwendigen Maßnahmen. Die folgende Aufstellung enthält eine Übersicht über notwendige und variable bzw. mögliche Maßnahmen, die im A-Block angeordnet werden müssen bzw. können:
Die mit einem Stern (*) gekennzeichneten Aktionen sind nutzer- bzw. problemindividuell und können entfallen, falls sie nicht benötigt werden.

- Startmeldung (*)
- Software-Vorspann (*)
- Eingabe, Prüfung und Protokollierung von Steuerinformationen (Steuerkarten, Vorlaufkarten, Bedienereingaben) (*)
- Programmodifikation auf Grund der eingegebenen Steuerinformationen (*)
- Eröffnung der sequentiellen Eingabedateien, falls deren Dateistatuswert verschieden von 3 ist
- Eingabe von Tabellen (*)
- Setzen von Anfangswerten (*)
- Löschen der Schalter, Indikatoren und Statuswörter

2.4.2. B-Block

Im Block B erfolgt das Einlesen der Sätze aus den sequentiellen Eingabedateien in Eingabebereiche, die Prüfung der Eingabedaten und das Beschicken der dateibezogenen Statuswörter. Der B-Block ist in so viele Unterblöcke B1 bis Bn unterteilt, wie es sequentielle Eingabedateien gibt. Die einzelnen Unterblöcke werden aufsteigend nacheinander durchlaufen. Ihr Aufbau ist identisch.
Je Eingabedatei wird im zugeordneten Unterblock

zunächst abgefragt, ob aus der Datei ein Satz einzugeben bzw. nachzuziehen ist (Abfragebedingung: LiS=∅ ?). Bei erfüllter Abfragebedingung wird ein Satz aus der Datei in den zugeordneten Eingabebereich übernommen. Die Schlüsselbegriffe des eingelesenen Satzes werden in entsprechender Folge (vergleiche auch Bild 2-3) in das Teilfeld LiO des dateibezogenen Statuswortes übertragen und das Feld LiS des dateibezogenen Statuswortes auf den Wert 1 gestellt.
Die Prüfung des eingegebenen Satzes erfolgt ebenfalls im entsprechenden Unterblock, damit gegebenenfalls dateispezifische Maßnahmen zur Fehlerbehandlung vorgenommen werden können.
Zu den im Unterblock eingegliederten Prüfungen gehört die ablaufbezogene dateiweise Satzreihenfolgekontrolle sowie feldbezogene Datenprüfungen (z. B. auf Alphazeichen, Prüfziffernrichtigkeit, Wertebereich). Die Satzreihenfolgekontrolle erfolgt durch Vergleich des Inhalts des Feldes LiO des jeweiligen dateibezogenen Statuswortes mit dem des Feldes LAO. Dabei muß stets gelten: LiO ≥ LAO.
Bei Erkennen eines Fehlers werden je nach Fehlerart bestimmte dateispezifische Maßnahmen notwendig, die vom Programmierer vorzusehen sind. Dazu gehören z. B. Ausgaben von Fehlermeldungen, Ausblenden des Satzes, Abbrechen des Programmes, Statusänderungen usw.
Unmittelbar nach der Eingabe oder dem Nachziehen eines Satzes aus einer Datei wird im zugeordneten Unterblock des B-Blockes auch die (in der Regel vom IOCS durchgeführte) Dateiendeabfrage durchgeführt. Bei erkanntem Dateiende wird die zugehörige Datei abgeschlossen, ihr Status auf den Wert 2 (=Datei abgeschlossen) gestellt und zum Ende des Unterblockes gesprungen. Zum Ende des Unterblockes wird auch dann gesprungen, wenn die Abfragebedingung LiS=0 am Anfang des Unterblockes nicht erfüllt wird.
In analoger Art werden alle entsprechend der Anzahl der sequentiellen Eingabedateien vorhandenen Unterblöcke Bi (i=1,2,...,n) durchlaufen.
Durch die Steuerung des B-Blockes über die Statusworte der einzelnen Dateien wird erreicht, daß beim erstmaligen Durchlauf des B-Blockes von allen Dateien der erste Datensatz eingelesen wird, während der weiteren Programmausführung wird jeweils nur von einer Datei nachgelesen, und zwar von derjenigen, von der der zuletzt verarbeitete Satz stammte.

2.4.3. C-Block

Im C-Block wird durch Auswahl über alle in den Eingabebereichen vorliegenden Sätze entschieden, welcher dieser Sätze für die nachfolgende Verarbeitung freigegeben wird.
Der C-Block ist ebenfalls in so viele Unterblöcke C1, C2, ..., Cn unterteilt, wie es sequentielle Eingabedateien gibt.

Die Auswahl und Freigabe erfolgt mit Hilfe der den vorliegenden Sätzen zugeordneten dateibezogenen Statuswörter. Dazu werden die Statuswörter fortlaufend miteinander verglichen und dasjenige ausgewählt, das von allen das - als Binärzahl gedeutet - kleinste ist. Da sich der Vergleich über das gesamte Statuswort erstreckt, ist ein solches "kleinstes" Statuswort immer eindeutig definiert. Selbst bei Gleichheit in Datei-Status und Gruppenbegriff ist das durch die vergebenen ungleichen Datei-Prioritäten gewährleistet.
Die Auswahl des "kleinsten" Statuswortes erfolgt sukzessive in den Unterblöcken C1 bis Cn. Im Unterblock C1 wird als Basis dafür lediglich der Inhalt des dateibezogenen Statuswortes L1 in das dateineutrale Statuswort LN übertragen.
In den folgenden Unterblöcken wird dann jeweils das entsprechende dateibezogene Statuswort Li (i=2,..., n) mit dem vorliegenden Statuswort LN verglichen. Gilt Li < LN, so wird Li nach LN übertragen, andernfalls bleibt LN unverändert.
Am Ende des Blockes C steht in LN das Statuswort, das den Satz repräsentiert, der als nächster zu verarbeiten ist. Das in LN stehende Statuswort enthält im Feld LND die Angabe darüber, aus welcher Datei dieser Satz stammt.

2.4.4. D-Block

Im Block D erfolgt die Zuordnungsanalyse des ausgewählten Satzes und damit verbunden die Kontrolle auf ausgelöste Gruppenwechsel.
Diese Kontrolle erfolgt durch stufenweise aufsteigenden Vergleich der Teilfelder der beiden dateineutralen Statuswörter LA und LN. LA enthält das Statuswort des zuletzt verarbeiteten Satzes, LN das im Block C ausgewählte.
Ein Gruppenwechsel in der Gruppenstufe j (j=1,2, ...,m) liegt genau dann vor, wenn LNj ≠ LAj ist.
Die im Falle eines festgestellten Gruppenwechsels in den vorhandenen Gruppenstufen durchzuführenden Gruppenverarbeitungen (und Ausgaben) sind problem-

spezifisch und aus Gründen der exakten Trennung zwischen Steuer- und Verarbeitungsblöcken in die entsprechend der Anzahl der Gruppenstufen zu schreibenden G-Blöcke G1, G2, ..., Gm ausgelagert. Der Block D steuert lediglich den Ansprung dieser G-Blöcke. Die Gruppenwechselkontrolle beginnt auf der Gruppenstufe 1 durch den Vergleich LN1 : LA1. Sind LN1 und LA1 ungleich, so wird der Indikator QL1 eingeschaltet und das der ersten Gruppenstufe zugeordnete Gruppenverarbeitungsprogramm im Block G1 abgearbeitet, wenn dieses Programm vom Programmierer durch Einschalten des Schalters QG1 freigegeben war. Sind dagegen LN1 und LA1 gleich, so liegt sowohl in der ersten als auch in allen anderen Gruppenstufen kein Wechsel vor, und es wird zum Schluß des D-Blockes gesprungen.
In analoger Weise gilt dieser Ablauf für alle weiteren Gruppenstufen, wobei nach dem eben Gesagten nur dann zur nächst höheren Stufe weitergegangen wird, wenn sich LNj und LAj noch unterscheiden. Unterscheiden sich LNj und LAj auch noch in der höchsten Gruppenstufe (j=m), dann erfolgt die Abfrage nach dem Programmende. Das Programmende ist dann erreicht, wenn LNS auf 2 steht, d. h. es existiert keine nichtabgeschlossene Datei mehr. In diesem Fall wird vom Block D zum Schlußprogramm im Block F verzweigt, andernfalls erfolgt am Ende des Blockes D der Übergang zum Block E.

2.4.5. E-Block

Im Block E erfolgt eine Analyse des im Block C ausgewählten und in LN stehenden Statuswortes, um festzustellen, aus welcher Datei der zur Verarbeitung anliegende und freigegebene Satz stammt.

Die entsprechend der Anzahl der sequentiellen Eingabedaten abzuarbeitenden Einzelverarbeitungen sind ebenfalls problemspezifisch und von Datei zu Datei unterschiedlich. Sie sind deshalb ebenso wie die Gruppenverarbeitungsteile von der Steuerung getrennt und in die vom Programmierer problembezogen zu schreibenden H-Blöcke H1, H2, ..., Hn ausgelagert.
Im Block E wird das Anspringen desjenigen H-Blockes realisiert, aus dem der zur Verarbeitung freigegebene Satz stammt.
Nach Durchlaufen des entsprechenden H-Blockes wird zum Block E zurückgesprungen. Es erfolgt die Umspeicherung LN nach LA. Der Wert des Feldes LiS für die betreffende Datei wird auf 0 gestellt, um

dadurch das Nachziehen eines neuen Satzes aus dieser Datei im nächsten Durchlauf vorzubereiten. Schließlich wird wieder zum Block B zurückverzweigt und ein neuer Durchlaufzyklus gestartet.

2.4.6. F-Block

Im Block F ist das Schlußprogramm untergebracht. Es enthält alle nach der Verarbeitung des letzten Datensatzes noch notwendigen Arbeiten. Hierzu gehören z. B.

- die Ausgabe von Endergebnissen, Schlußsätzen und Programm- und Bedienerinformationen
- das Abschließen der verwendeten Direktzugriffs- und Ausgabedateien
- die Ausgabe einer Schlußmeldung
- die Beendigung des Programmes.

2.4.7. G-Blöcke

In den G-Blöcken sind die bei Gruppenwechsel zu durchlaufenden Gruppenverarbeitungsprogramme angeordnet. Sie werden entsprechend der im Punkt 2.4.4. angegebenen Art angesprungen und geben nach ihrem Durchlaufen die Steuerung wieder an den Block D zurück.
Das Durchlaufen eines Blockes Gj (j=1,2,...,m) erfolgt jedoch nur, wenn der Programmierer vorher (in der Regel in einem der H-Blöcke des vorherigen Durchlaufs) den zugehörigen Schalter QGj eingeschaltet hatte (QGj=1). In den einzelnen G-Blöcken sind gruppenstufenabhängige Verarbeitungen und Ausgaben vorzunehmen.

2.4.8. H-Blöcke

Die H-Blöcke H1, H2, ..., Hn enthalten für jede der n sequentiellen Eingabedateien die satzbezogenen Einzelverarbeitungen und Ausgaben. Alle H-Blöcke sind problemspezifisch. Eine Gliederung in den H-Blöcken wird im allgemeinen bei einem Auftreten mehrerer Satzarten innerhalb einer Datei für deren entsprechende Abarbeitung notwendig.
In den H-Blöcken können die vom Block D bereitgestellten Indikatoren QLj (j=1,2,...,m) für Verzweigungen bzw. Sondermaßnahmen ausgewertet werden.

3. Das System -NPS-

3.1. Allgemeines

Das auf der Basis der Methode der Normierten Programmierung aufgebaute Komponentensystem -NPS- ermöglicht die Erzeugung des Ablaufsteuerungsteiles von kommerziellen Programmen ohne eigene Befehlsangaben anhand weniger und leicht handhabbarer Anweisungen.
Der Programmierer wird bei Nutzung des Systems -NPS- vollständig von der Organisation und Programmierung der Programmlogik entlastet und braucht sich nur noch mit den Problemen der eigentlichen Satz- und Gruppenverarbeitung und deren Programmierung zu befassen. Das zum Ablauf und dem richtigen Zusammenwirken der vom Programmierer geschriebenen Verarbeitungsteile notwendige steuernde Hauptprogramm wird über das System -NPS- mit Hilfe von nur drei notwendigen und vier (für bestimmte Komfortfunktionen bestimmte) wahlfreien Makro-Anweisungen erzeugt.
In diesen Makro-Anweisungen beschreibt der -NPS-Nutzer sowohl die für den Programmablauf notwendigen Angaben über die Anzahl der Gruppenstufen, die Anzahl der sequentiellen Eingabedateien, deren logische und physische Zuordnung und Struktur, die Lage der Sortier-Gruppierbegriffe in den Datensätzen als auch für bestimmte Programmfunktionen gewünschte Sondermaßnahmen durch Angabe einfacher Kennwortparameter.
Der mit Hilfe von -NPS- generierte Steuerteil des abzuarbeitenden Programmes enthält alle für die Datenzufuhr und Programmablaufsteuerung notwendigen Befehle.
Von den im System der Normierten Programmierung strukturierten Blöcken sind im Steuerteil folgende Blöcke enthalten:

- A-Block (allgemeiner Teil)
- B-Block
- C-Block
- D-Block
- E-Block
- F-Block (allgemeiner Teil).

Alle problemspezifischen Teile

- A*-Block (anwendungsbezogener Teil)
- G-Blöcke

- H-Blöcke
- F*-Block (anwendungsbezogener Teil)

werden vom Steuerteil als Unterprogramme angesprungen. Sie können vom Programmierer selbständig und vom Steuerteil unabhängig programmiert werden.

Bild 3-1 zeigt den Aufbau eines mit dem System -NPS- realisierten Programmes, das Zusammenwirken der einzelnen Programmteile und die Verantwortlichkeit ihrer Erzeugung.

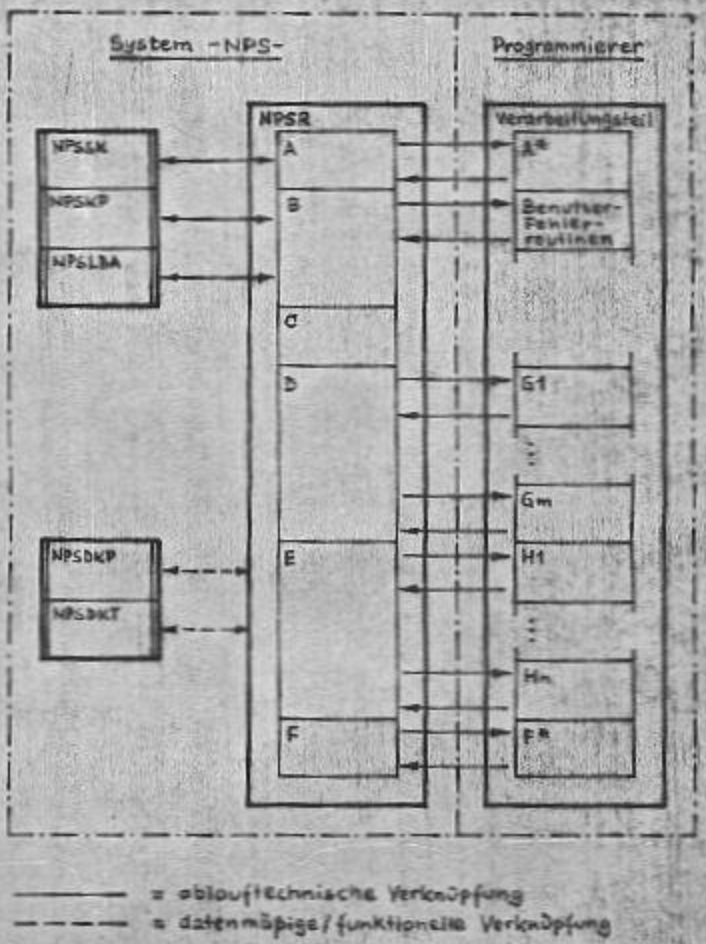


Bild 3-1. Programmablauf im System -NPS-

3.2. Einordnung im Betriebssystem DOS/ES

Die Komponenten des Systems -NPS- lassen sich im Betriebssystem DOS/ES nach ihrer Einordnung in die drei Bibliotheksebenen SL, RL und CL unterscheiden.

Die Komponenten zur Anpassung an das spezielle Problem sind Makrodefinitionen, die in der SL des Betriebssystems gespeichert sind. Mit ihrer Hilfe werden in einem mehrstufigen Generierungsprozeß die in der Modul- bzw. Phasenbibliothek gespeicherten Komponenten zur Steuerung und Verarbeitung problembezogen modifiziert bzw. durch Erzeugung externer Bezugnahmen zur Einfügung im Programmverbindungs- bzw. Ausführungslauf vorgemerkt, so daß in Verbindung mit dem vom Programmierer hinzugefügten Verarbeitungsteil die problemgerechte Programmabarbeitung gemäß Bild 3-1 gewährleistet wird.

3.3. Ausbaustufen des Systems -NPS-

Das System -NPS- steht in mehreren Ausbaustufen zur Verfügung.
Die Grundstufe (Ausbaustufe 1) enthält alle für die Funktion der Erzeugung des Steuerteiles notwendigen Komponenten sowie eine Reihe von Komponenten, die den Programmierer bei der Programmierung der Verarbeitungsblöcke unterstützen.
Bereits in der Ausbaustufe 1 ist das System -NPS- voll funktionsfähig und bietet alle im Punkt 1 genannten prinzipiellen Vorteile.
Alle weiteren Ausbaustufen des Systems -NPS- erweitern dessen funktionelle Möglichkeiten und erleichtern seine Anwendung für spezielle Probleme.
In den höheren Ausbaustufen steht dem -NPS-Nutzer ein weiterer funktioneller Komfort zur Verfügung.

Eine detaillierte Aufstellung über die Komponenten des Systems -NPS-, ihre Eingliederung in das Betriebssystem DOS/ES und den Zeitpunkt ihrer Verfügbarkeit ist dem Anhang A dieser Schrift zu entnehmen.

3.4. Einführungsbeispiel

Die Handhabung des Systems -NPS- durch einen Programmierer ist einfach, übersichtlich und leicht erlernbar.

Zur einführenden Illustrierung des äußeren Bildes einer mit -NPS- realisierten Programmierung diene der in den Bildern 3-2 bis 3-4 gegebene Beispieljob.

```
// JOB NPS     EINFUEHRUNGSBEISPIEL
weitere jobsteueranweisungen nach bedarf
* AUFRUF ZUR GENERIERUNG DES
* -NPS-VERSTAENDIGUNGS- UND
* STEUERTEILES
// EXEC ASSEMBLY
          PRINT NOGEN
          NPSGN programmname,
                gruppenstufenanzahl,
                anzahl der seq. eingabedateien
                gliederung des gruppenfeldes
                spezielle wünsche
          NPSFL dateibeschreibung datei 1
                gruppenfeldzusammenstellung
                gewünschte datensatzprüfungen
          :
          NPSFL dateibeschreibung datei n
                gruppenfeldzusammenstellung
                gewünschte datensatzprüfungen
          NPSEND
          END
/*
```

Bild 3-2. Einführungsbeispiel, Steuerteil

```
* AUFRUF ZUR UEBERSETZUNG
* DES PROBLEMSPEZIFISCHEN
* VERARBEITUNGSTEILES
// EXEC PL/I
 A..     PROCEDURE(RØØ),.
         DECLARE arbeitsbereiche für seq.
                 eingabedateien
                 EXTERNAL,.
         DECLARE spezielle bezeichner,.
         setzen von anfangswerten,...
         RETURN,.
 E1..    ENTRY(RØØ),.
         einzelverarbeitung der sätze der datei 1
         RETURN,.
  :
 En..    ENTRY(RØØ),.
         einzelverarbeitung der sätze der datei n
         RETURN,.
 G1..    ENTRY(RØØ),.
         gruppenverarbeitung für gruppenstufe 1
         RETURN,.
  :
 Gm...   ENTRY(RØØ),.
         gruppenverarbeitung für gruppenstufe m
         RETURN,.
 F..     ENTRY,.
         schlußmaßnahmen
         RETURN,.
         END,.
 /*
```

Bild 3-3. Einführungsbeispiel, Verarbeitungsteil

```
* PROGRAMMVERBINDERLAUF
 ENTRY NPS
// LBLTYP ...
// EXEC LNKEDT
* AUSFUEHRUNGSLAUF
gerätezuweisungen
kennsatzinformationen
// EXEC
steuerkarten   }
vorlaufkarten  } nach bedarf
datenkarten    }
/*
/&  ENDE DES EINFUEHRUNGSBEISPIELES
```

Bild 3-4. Einführungsbeispiel, Link- und Ausführungslauf

Anhang A: -NPS-Komponenten

Komponente	Inhalt/Funktion der Komponente	Bibliothek	verfügb. ab Ausbaust.
NPSGN	Festlegungen zum Programmverständigungsbereich	SL	1
NPSDK	Festlegungen für Drucklistenformat und -steuerung	SL	1
NPSMK	Festlegungen zum Vorlaufkartenbereich	SL	1
NPSFL	Dateidefinition und Datensatzbeschreibung der seq. Eingabedat.	SL	1
NPSEND	Aufruf der Parameterkontrollen	SL	1
NPSMT	Generierung des Magnetband - DTF	SL	1
NPSCD	Generierung des Lochkarten - DTF	SL	1

Tabelle A-1. Komponenten zur Anpassung an das Problem (Ausbaustufe 1)

Komponente	Inhalt/Funktion der Komponente	Bibliothek	verfügb. ab Ausbaust.
NPSIS	Generierung des index-seq. Platten - DTF	SL	2
NPSSD	Generierung des seq. Platten - DTF	SL	2
NPSLB	Lochband-satzbeschreibung	SL	3
NPSPT	Generierung des Lochband - DTF	SL	3
NPSU300	SCAN - R300 - Lochbandcode	SL	3
NPSL300	LTRANS - R300 - Lochbandcode	SL	3
NPSF300	FTRANS - R300 - Lochbandcode	SL	3
NPSST	Festlegungen zur H-Blockeinteilung nach Satzarten	SL	3

Tabelle A-1. Komponenten zur Anpassung an das Problem (Ausbaustufen 2 und 3)

Komponente	Inhalt/Funktion der Komponente	Bibliothek	verfügb. ab Ausbaustufe
NPSR	-NPS- Hauptsteuerroutine	RL/CL	1
NPSKP	Datensatzkontrolle	RL/CL	2
NPSSK	Steuerkarten- und Vorlaufkarteneingabe	RL/CL	2
NPSDKP	Drucklistensteuerung für Listenausgabe auf Drucker	RL/CL	2
NPSDKT	Drucklistensteuerung für Listenausgabe auf Magnetband	RL/CL	3
NPSLBA	Lochbandsatzaufbereitung mit Umcodierung	RL/CL	3

Tabelle A-2. Komponenten zur Steuerung und Verarbeitung

Normierte Vorbelegung von 3 Dateien (ohne Satzlängenkontrolle)
mit 3 Ordnungsbegriffen

Programminitialisieren, Definition der Eingabestrukturen,
Definition der Schalterschaltern für die normierte Programmierung (LØ1, LØ2, LØ3, LN,
Endebedingungen für die Dateien: Statusbyte jeweils auf 2 setzen,
Anfangswerte setzen, Eröffnen der Dateien

DO WHILE (Statusbyte LN ≠ 2) 'VLNS

sequent. Eingabe:

IF XLØ1S = Ø
THEN Satz von D1 lesen
IF XLØ1S ≠ 2
THEN XLØ1S = 1, Ordnungsbegriff von D1 → ULØ1
END IF
ENDIF

IF XLØ2S = Ø
THEN Satz von D2 lesen
IF XLØ2S ≠ 2
THEN XLØ2S = 1, OB von D2 → ULØ2
END IF
ENDIF

IF XLØ3S = Ø
THEN Satz von D3 lesen
IF XLØ3S ≠ 2
THEN XLØ3S = 1, OB von D3 → ULØ3
ENDIF
ENDIF

Satzauswahl: LN = LØ1
IF LN > LØ2 THEN LN = LØ2
IF LN > LØ3 THEN LN = LØ3
IF VLND = 1 THEN XLØ1S = Ø
IF VLND = 2 THEN XLØ2S = Ø
IF VLND = 3 THEN XLØ3S = Ø

Zwischensatz-
analyse:

```
IF   LA ) LN
THEN LA = LN   , Anfangsvorbelegung
ELSE IF LA01 ≠ LN01
        THEN CALL GV3 (Verarbeitung 3. (niedrigste) Summenstufe)
             IF LA02 ≠ LN02
                THEN CALL GV2 (Verarbeitung 2. Summenstufe)
                     IF LA03 ≠ LN03
                        THEN CALL GV1 (Verarbeitung 1. Summenstuf)
                     ENDIF
             ENDIF
        ENDIF
        LA = LN
ENDIF
```

Einzelsatz: Summieren, Verbuchen oder Einzelsatz je nach XLND

END DO;

Gruppenwechs.: GV3: (Verarbeitung niedrigste Gruppenstufe)

GV2: (" 2. ")

GV1: (" höchste ")

```
IF LN > LA02
THEN LN = LA02
ENDIF
```

```
DCL 1 ULN,  /* Struktur Gruppen Satz. Feld neu */
      2 XLNS   CHAR (1)   Datei Statistik
      2 VLN03  FIXED      1. (niedrigste) Ordnungsbegriff
      2 VLN02  "          2.      OB
      2 VLN01  "          3. (höchste) OB
      2 XLND   CHAR (1)   Datei nummer
```

andere Strukturen ULA, ULO1, ULO2, ULO3 haben gleichen Aufbau

```
DCL LN    CHAR (L'ULN')  DEF (ULN)
    LN01  CHAR (L'ULN'-1) DEF (ULN)
    LN02  CHAR (L'VLN03 + L'VLN02' + 1) DEF (ULN)
    LN03  CHAR ( L'VLN03' 1) DEF (ULN)
```