DAS BBC-PEARL-PROGRAMMIERSYSTEM UND SEINE ANWENDUNGEN IN DER NETZLEITTECHNIK

Dipl.-Math. G. Koch
BBC Abt. SI/NP2 6802 Ladenburg

1. Das BBC-PEARL-Programmiersystem

Seit 1966 beschäftigt sich die Fa. BBC mit dem Einsatz von prozedurorientierten Realzeit-Programmiersprachen zur Rationalisierung der Programmentwicklung für Automatisierungsprojekte (s.Tab. 1). Nach den ersten positiven Erfahrungen mit dem von BBC entwickelten PAS1-Programmiersystem unter Verwendung eines auf Prozeßrechneranforderungen erweiterten PL/I-Subset engagierte sich BBC von Anfang an bei der Entwicklung von PEARL. Die bereits im Jahre 1973 nach Fertigstellung der Definition von PEARL 73 in Angriff genommene Pilotimplementierung eines PEARL-Subsets für die PDP11-Rechnerfamilie wurde durch das PDV-Projekt der Bundesregierung im Rahmen des 2. DV-Förderungsprogramms unterstützt. Diese früheste PEARL-Implementierung besaß folgende Merkmale:

- kleiner PEARL-Subset
- kleines aber reaktionsschnelles PEARL-Betriebssystem
- PEARL-Compiler mit wirkungsvollen Möglichkeiten der Übersetzungs-Beeinflussung (Compile-time-statements)
- Test- und Bedienungshilfsmittel

Erste Einsatzerfahrungen bei der Abwicklung von industriellen Automatisierungs-Projekten bestätigten die Erwartungen:

- Erhebliche Rationalisierung von Programmentwurf und Programmtest gegenüber Assembler-Programmierung
- Unterstützung der strukturierten Programmierung

Nach der Durchführung mehrerer Projekte kamen von Seiten der Anwender folgende zusätzliche Forderungen:

Die diesem Bericht zugrunde liegenden Arbeiten wurden mit Mitteln des Bundesministers für Forschung und Technologie im Rahmen des 1. und 2. DV-Förderungsprogramms gefördert. Die Verantwortung für den Inhalt liegt jedoch allein beim Autor.

- Vergrößerung des Subsets
- Erweiterung des Betriebssystems unter Beibehaltung der kurzen Reaktionszeiten
- Bereitstellung von weiteren Hilfsmitteln für größere Projekte

Durch Auswertung des Erfahrungsrückflusses aus über 100 Anlagen in verschiedenen Branchen (s.Bild 1) entwickelte sich das zum heutigen Zeitpunkt bei BBC im industriellen Einsatz befindliche PEARL-Programmiersystem:

- Verfügbar für die PDP11-Rechnerfamilie von der LSI/11 bis zur PDP 11/70
- abgerundeter PEARL-Subset, der deutlich über BASIS-PEARL hinausgeht
- ausgereiftes, bedienungsfreundliches PEARL-Programmiersystem
- auch als core-only Version verfügbar
- Wartung und Pflege durch BBC

In der Zukunft sieht BBC die Bereitstellung und Pflege von effizienten PEARL-Programmiersystemen als Aufgabe der Rechnerlieferanten. In diesem Sinne wurde die Fa. Digital Equipment durch BBC angeregt, eine PEARL-Implementierung bereitzustellen.

In Bild 2 sind die wesentlichsten Komponenten des BBC-PEARL-Programmiersystems dargestellt. Alle Teile sind so optimal aufeinander abgestimmt, daß sämtliche Fehlermeldungen sowie Bedien- und Testhandlungen sich unmittelbar auf die PEARL-Quellenprogramme beziehen.

Folgende Leitlinien lagen dem Entwurf des Programmvorbereitungskomplexes zugrunde:

- Erkennung aller Formalfehler in einem PEARL-Programmpaket
- Bereitstellung von Listen und Statistiken zur Unterstützung bei Fehlersuche und bei Programmänderungen

- Vollständige Referenzen zwischen Quellenprogramm und Speicherplatz (Befehle oder Daten)

Dabei kommt dem Compiler- und Prelinker-Listings zentrale Bedeutung zu.

Das Compiler-Listing eines PEARL-Moduls besteht aus folgenden Teilen:

Quellenprogramm mit:
- Blockstrukturtiefe
- Statementnr. und Ablage-Adresse

Fehlermeldungen:
- Statementnr.
- Fehlerstart
- Namen der fehlerbehafteten Variablen

Crossreferenz:
- alle Vereinbarungen alphabetisch sortiert
- Adresse für Ablage
- Attribute der Größen
- Benutzungsliste

Statistische Daten

Beim Prelinken eines PEARL-Anwenderpakets entstehen folgende Listen:

Liste der Module:
- Module-Name
- Compilierdatum
- Fehlerzustand

Liste der Task's und Prozeduren:
- Name
- Module-Name
- Speicherbedarf

Module-Statistik:
- Module-Name
- Speicherbedarf
- Zuordnungsdaten

Task-Statistik
- Taskname
- Priorität
- Speicherbedarf
- Module-Name

Fehler-Diagnostik
- Name der globalen Variablen
- Referenz in Module
- Deklariert in Module
- Fehlerart

Global-Cross-Reference
- Name der globalen Variablen
- Modul, in welchem die Variable deklariert ist
- vollständige Attributliste
- Liste der Module, die Spezifikationen enthalten

Die Komponenten des Laufzeitsystems sind nach gleichen Grundsätzen entwickelt worden, d.h. das Quellenprogramm ist die Bezugsbasis:

- für Test- und Bedienhandlungen
- für Fehlermeldungen des Betriebssystems

2. Beispiel für den Einsatz von PEARL bei EVU-Projekten: Zentrale Netzleitstelle für die Stadtwerke Mannheim (SMA)

Auch für Projekte aus dem EVU-Bereich wurde seit 1974/75 das BBC-PEARL-Programmiersystem eingesetzt. Das Projekt der zentralen Netzleitstelle für die Stadtwerke Mannheim wird als Beispiel aus 74 Anlagen herausgegriffen, da neben den üblichen EVU-Aufgaben auch die des überlagerten Netzschutzes mit einer Reaktionszeit unter 300 msec zu erfüllen sind. Dieser Reserveschutz greift bei Versagen des Abzweigschutzes ein, um die Auslösung der Sammelschieneneinspeisung zu verhindern.

Die Projektaufgabe besteht in der Realisierung der zentralen Netzführung für das E-Netz des Stadtgebietes von Mannheim mit folgenden Zielen:

- Erhöhung der Verfügbarkeit und Betriebssicherheit des Netzes
- Erhöhung der Wirtschaftlichkeit durch Nutzung von Netzreserven
- Entlastung des Betriebspersonals von Routineaufgaben

Dafür wurden in der zentralen Netzleitwarte folgende Aufgaben realisiert:

- Darstellung des aktuellen Netzzustandes in Form von Übersichts-, Teil- und Anlagenbildern auf Farbmonitoren:
  - graphische Darstellung des Netzes
  - Zustand der Objekte durch Farbgebung
  - aktuelle Daten (Spannung, Strom, Leistung, Temperatur)
  - Markierung von irregulären Zuständen (Farb-
  - hinterlegung, Blinken)

- Auslösung von Schalthandlungen durch Identifikation des Objektes im Bild und Kommandogabe über Funktionstastatur

- signifikante Meldung von Prozeßereignissen
  - akustisches Signal (durch Gong)
  - Anzeige der betroffenen Stationen im Netzübersichtsbild
  - automatische Bildaufschaltung mit blinkenden Objekten
  - Warn- u. Störmeldungen auf a/n-Monitor und Protokolldrucker

- automatische Reaktion auf Prozeßereignisse
  - überlagerter Netzschutz bei Versagen des Primärschutzes

- Handnachführung von Netzzustandsänderungen über Bild- und Funktionstastatur
  - Netztrennstellen
  - Erdungspunkte

- Nachführung von Netzstrukturänderungen durch Ergänzung/Änderung der Datenbank
  - neue Umspannwerke, Schaltanlagen, Übergabestationen, Netzstationen
  - neue Meßwerte

- Netzübersichtsbild als Landkarte mit 4 Sammelmeldungen je Station

- 16 beliebig wählbare Meßwerte auf Meßwertschreiber aufzeichnen

- Abruf von Protokollen
  - Netztrennstellenübersicht
  - anstehende Warnmeldung usw.

- Gewährleistung sehr hoher Verfügbarkeit der Netzleitstelle

Alle diese Aufgaben wurden mit der in Bild 3 dargestellten Doppelrechnerkonfiguration gelöst. Die Tabelle 2 gibt eine prägnante Übersicht über die charakteristischen Daten des E-Netzes und der Automatisierungskomponenten.

Die gesamte Software der zentralen Netzleitwarte wurde mit dem BBC-PEARL-Programmiersystem realisiert. Sie umfaßt:

- ca. 60.000 PEARL-Statements
- ca. 0,5 M Byte für Programme
- ca. 5 M Byte Daten

Das Softwarepaket besteht aus 90 PEARL-Modulen und enthält:

- 39 Tasks
- 200 Prozeduren (auf Modulebene)
- 650 globale Namen

Zur Bearbeitung der Aufgaben sind folgende Datenspeicher notwendig:

- 150 K Worte Kernspeicher
  (davon ca. 20 K Worte für Betriebssystem)
- 7,5 M Byte Nachschubspeicher für Daten und Programme

Die Datenbasis des SMA-Systems zeigt das Bild 4. In den Bildern 5 und 6 sind am Beispiel der Aufgabe des überlagerten Netzschutzes die Bearbeitungsschritte, der Datenfluß und die globale Ablaufsteuerung dargestellt.

3. Erfahrungen mit dem Programmiersystem

In den gesamten, in Phasen aufgeteilten strukturierten Abwicklungsprozeß ist die Benutzung von PEARL und des PEARL-Programmiersystems eingebettet (s.Tabelle 3).

Der Rationalisierungseffekt bei Einsatz der Sprache beginnt bereits bei der Erarbeitung der Spezifikationen durch Benutzung der Datentypen und tritt voll in Erscheinung beim Entwurfsprozeß des Systems. Hierbei sind die PEARL-Konzepte für Tasking, Synchronisation und Dations sowie das der Blockstruktur (lokale und globale Daten) und des Parametermechanismus besonders tragfähig, um die zentrale Ablaufsteuerung darzustellen und Rückwirkungsfreiheit und Modularität der Programmbausteine zu sichern. PEARL-Elemente werden im Fortgang des Entwurfsprozesses als Pseudo-Code in Struktogrammen verwendet, so daß ein lückenloser Übergang bei der Codierung der PEARL-Module gewährleistet ist.

Die Qualität des BBC-PEARL-Programmiersystems prägt die sich anschließende Programmvorbereitungsphase. Beim Compilieren der PEARL-Module und dem Prelinken werden alle Formalfehler erkannt und prägnant ausgewiesen, so daß für Test und Inbetriebnahme das Aufspüren der inhaltlichen Fehler verbleibt. Dafür stehen quellenprogrammorientierte Test- und Bedienhilfsmittel zur Verfügung, d.h. ein Rückgriff auf die Assemblersprache ist nicht erforderlich.

Die vom Compiler und Prelinker erzeugten Statistiken erlauben, Optimierungen hinsichtlich Speicherraum und/oder Ausführungszeit zielsicher durchzuführen.

Änderungen und Erweiterungen des Programmpaketes waren durch die strukturierte Entwurfstechnik unter Ausschöpfung der Compiler- und Prelinkerlistings unproblematisch durchführbar.

Die Erfahrungen aus den abgewickelten Projekten zeigen, daß erst die Synthese aus einer strukturierten Projektabwicklung und einem der Sprache PEARL gerechtwerdenden PEARL-Programmiersystem, (d.h. eine echte PEARL-Umgebung), den optimalen Rationalisierungseffekt für Softwareerstellung und -pflege erreichen lassen. Diese Erkenntnis ist seither die Leitlinie der Projektabwicklung für die BBC-Netzführungssysteme.

1966: Beginn der Entwicklung von Sprache, Compiler und Betriebssystem für das Prozeß-Automatisierungs-System "PAS 1", basierend auf einem PL/I-Subset. Zielrechner war die "H 316".

Ab 1969: Einsatz dieses Software-Systems. PAS 1 war damals die erste, echte höhere Prozeßsprache, die in der BRD entwickelt wurde.

1970: Mitbegründung des PEARL-Arbeitskreises

1973: Definition eines Subsets aus dem Gesamtumfang der höheren Prozeßsprache PEARL 73. Entwicklung des zugehörigen Compilers, eines PEARL-Betriebssystems und eines Bedien- und Testsystems auf Sprachen-Niveau für die Prozeßrechnerfamilie "PDP 11":

- kleiner Subset
- kleines reaktionsschnelles Betriebssystem
- effizienter Compiler mit Compilezeit-Beeinflussung
- Testhilfsmittel, Bedienung

1974/75: erste Einsatzerfahrungen

- PEARL unterstützt die strukturierte Programmierung
- Vergrößerung des Subsets erforderlich
- Erweiterung des Betriebssystems ohne Verschlechterung der Reaktionszeiten
- Entwicklung von Hilfsmitteln für größere Projekte

1975 - heute: Erweiterung des BBC-PEARL-Programmiersystems

- PDP 11/03 .... PDP 11/70
- abgerundetes Subset
- ausgereiftes Gesamtsystem
- erprobt in mehr als 100 Projekten
- mit Platte oder reines Kernspeichersystem
- Wartung u. Pflege im industriellen Einsatz

Tabelle 1: Zeitlicher Ablauf des Einsatzes höherer Prozeßsprachen bei BBC

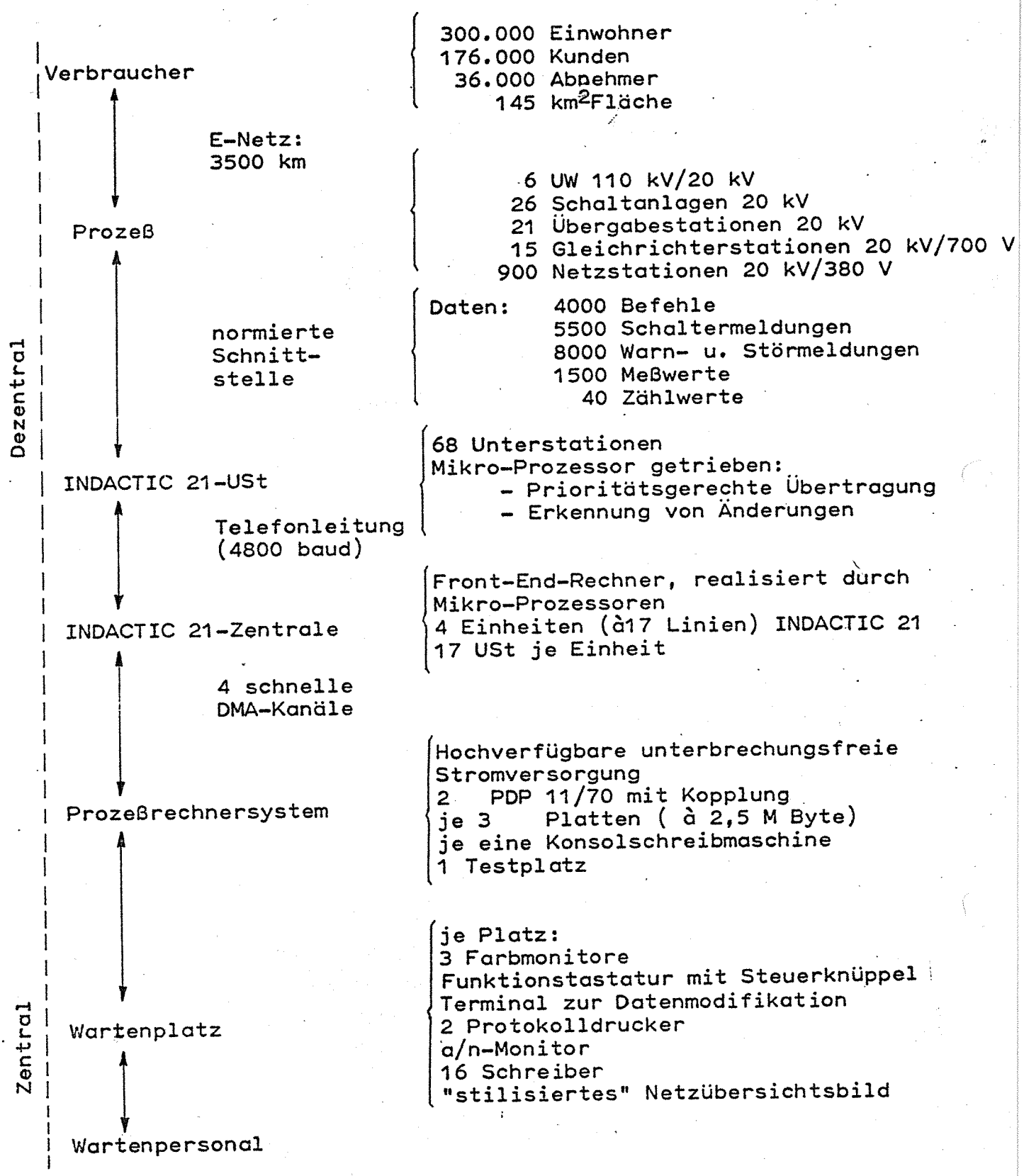

Tabelle 2: Charakteristische Daten des SMA-Systems

Projektphasen	PEARL-System	Einfluß
Aufgaben-definition Spezifikation Systementwurf grob Systementwurf fein Codierung	PEARL-Sprache	- Datentypen - PEARL-Konzepte: Tasking, Synchronisation, Dations - lokale, globale Daten - PEARL als Pseudocode in Struktogrammen - lückenloser Übergang auf PEARL-Module

Übersetzung
 	
Test
 	
Inbetriebnahme
 	
Pflege
 	
Erweiterungen/ Änderungen
	PEARL-Programmiersystem	- zügige Fehlerbeseitigung je Module
		- Prelinker sichert Freiheit von Formalfehlern
		- Problemlose Fehlersuche durch Test u. Bedienhilfs-mittel auf Quellenebene
		- Optimierung zielsicher:
		  - Statisktiken d. Prelinken
		  - Compilezeit-Statement
		- unproblematisch durch
		  - strukturierten Entwurf
		  - aktuelle Cpl-u. Prelinker-Listen

Tabelle 3: Einfluß von PEARL bei den Projektphasen

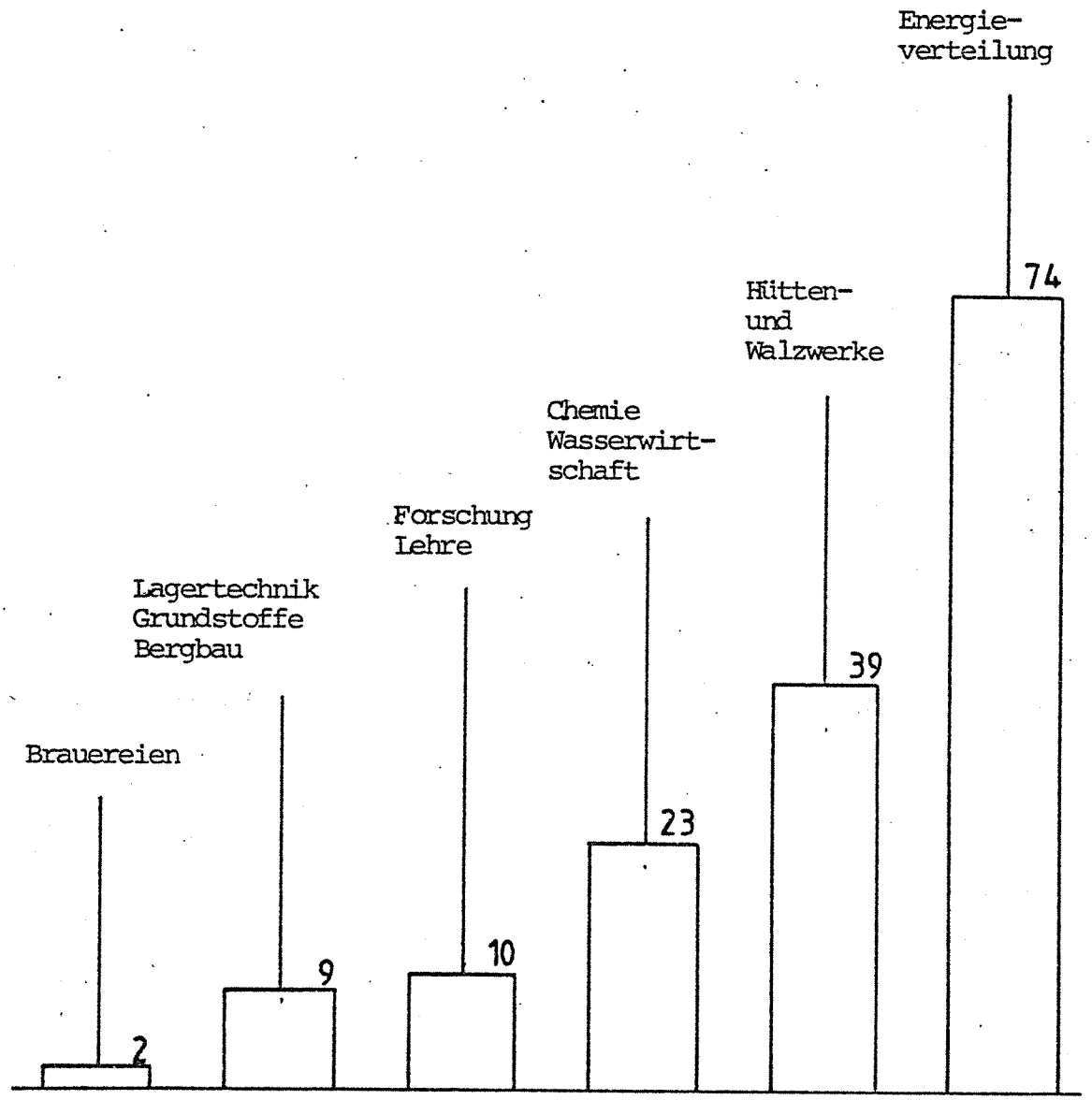

Bild 1: In Auftrag und Betrieb genommene PEARL-Rechner kumuliert.
Stand: Dezember 1979

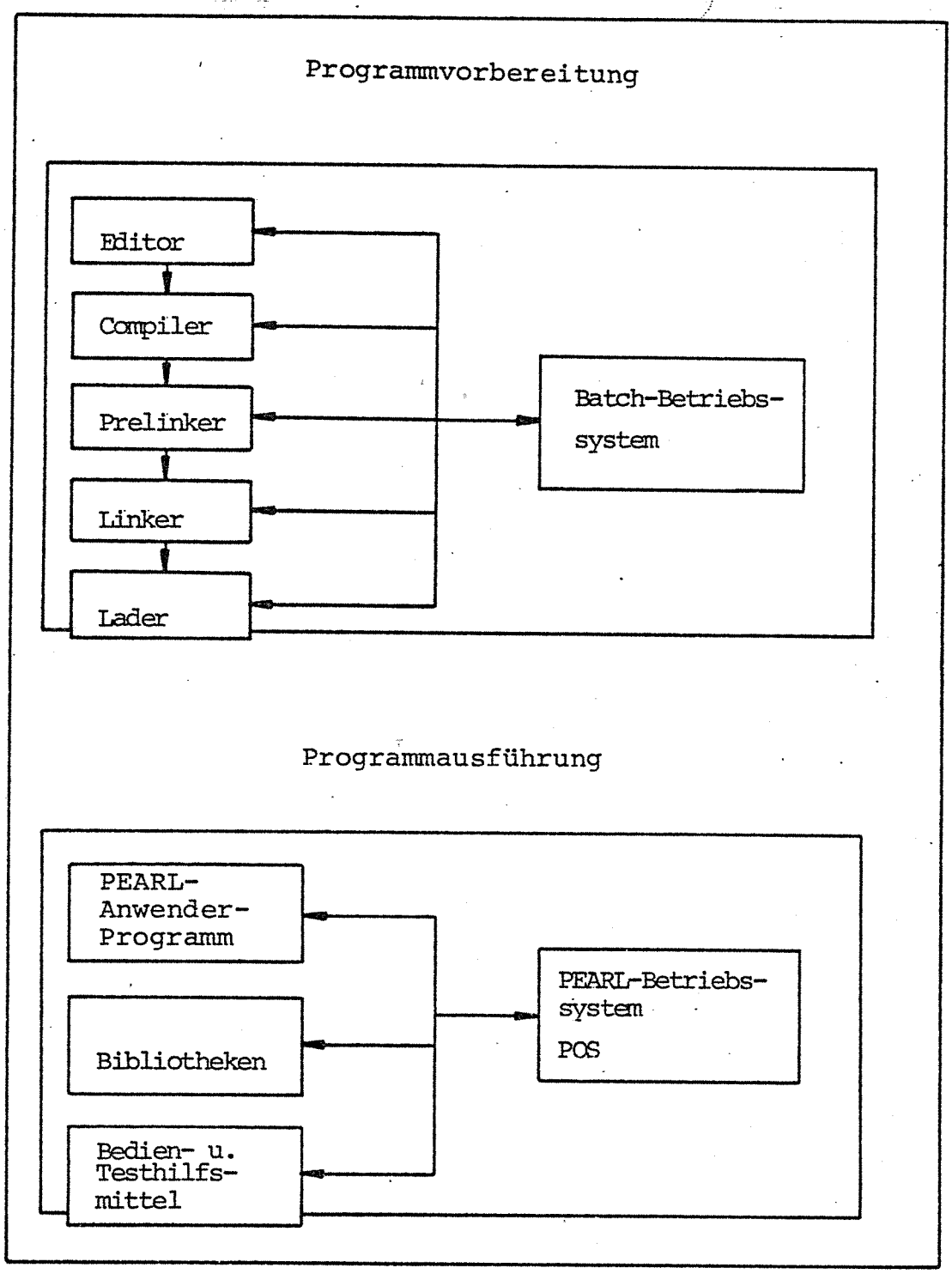

Bild 2: Komponenten des BBC-PEARL-Programmiersystems

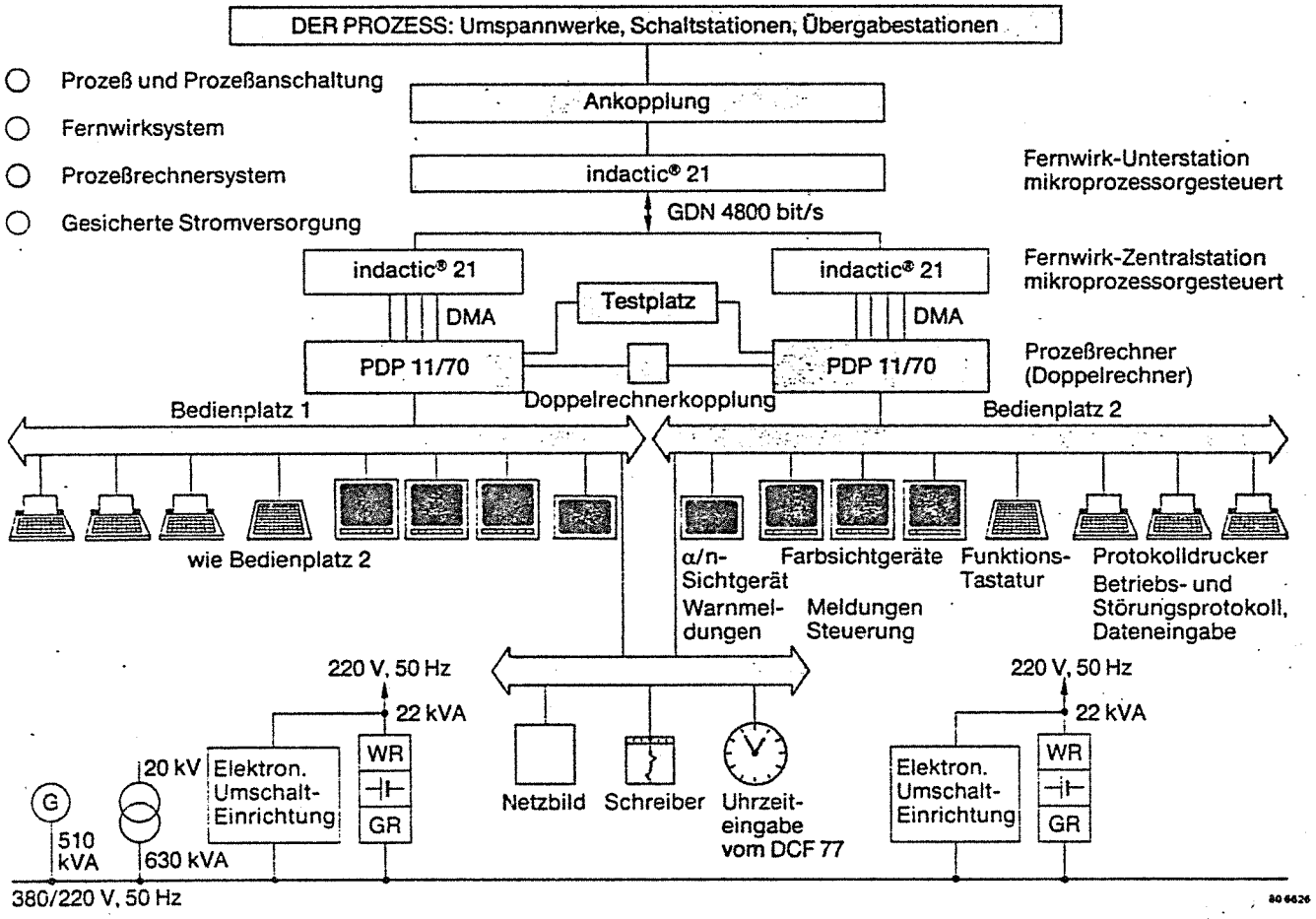

Bild 3: Blockschaltbild des Netzleitsystems für SMA

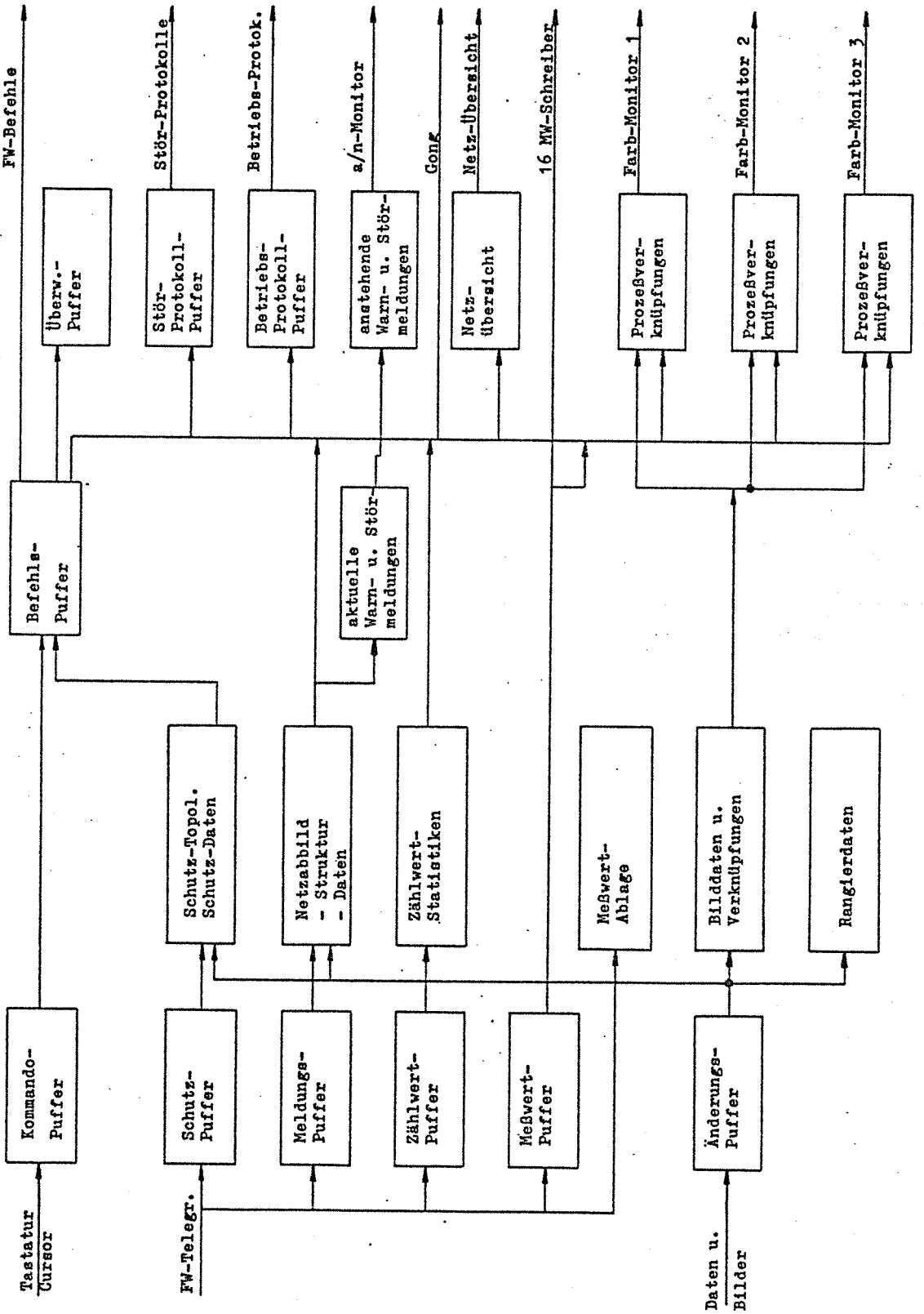

Bild 4: Datenbasis und Datenfluß des SMA-Systems

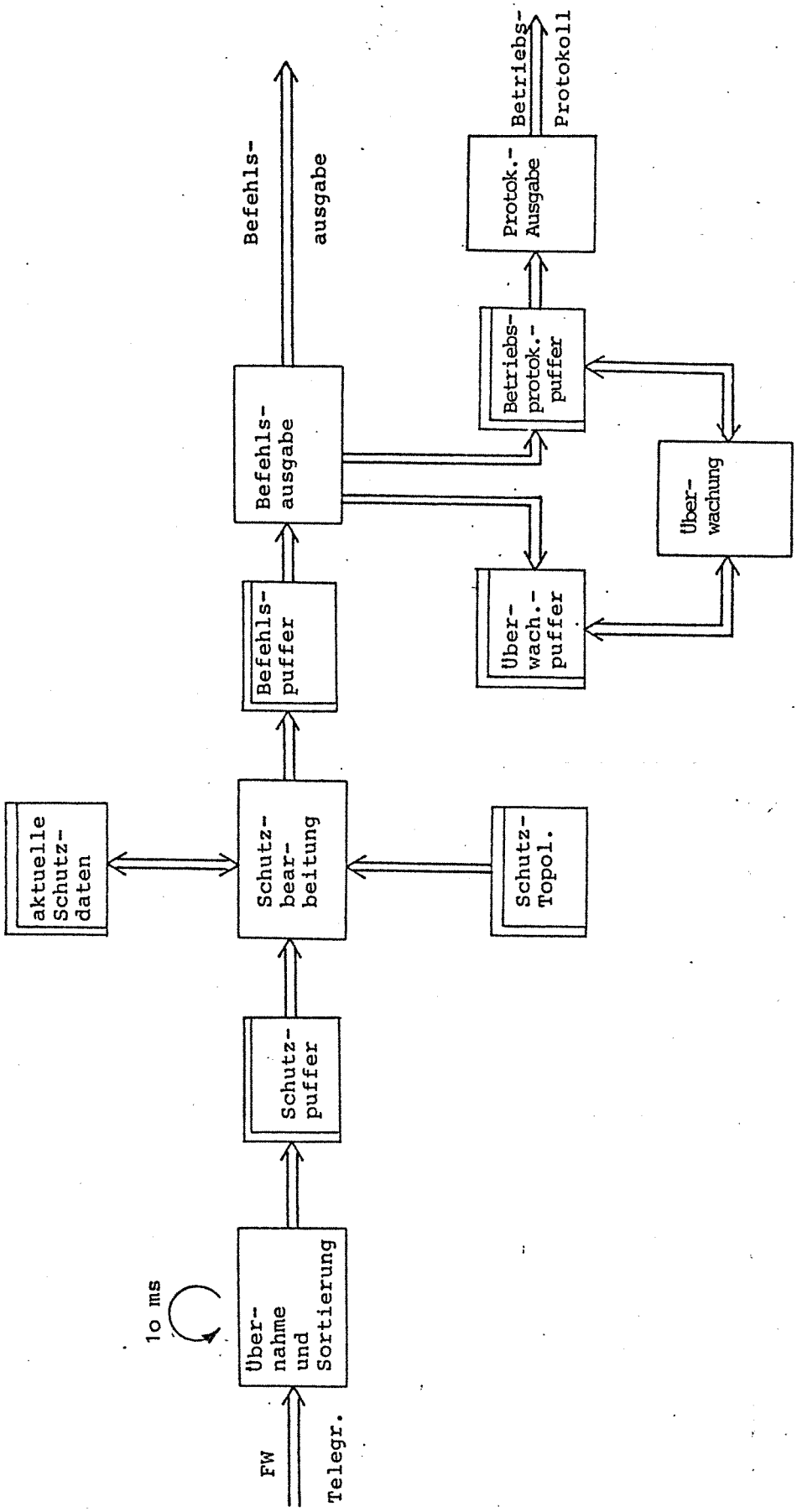

Bild 5: Bearbeitungsschritte und Datenfluß
für die Netzschutzaufgabe bei SMA

```
STARTM:   TASK; /*STARTTASK DES SYSTEMS */
          .
          .
          EVERY 10 MSEC ACTIVATE (TELEIN);
           /* EINPLANUNG DER ZYKLISCHEN ÜBERNAHME DER FW-TELEGRAMME*/
          .
          .
          END;
```

```
TELEIN:   TASK;  /* UEBERNAHME UND SORTIERUNG DER FERNWIRK-
                    TELEGRAMME */
          .
          .
          ACTIVATE  (SCHUTZ);    /* FALLS ANREGUNGS- ODER RICHTUNGS-
                                    KRITERIEN EINLAUFEN */
          .
          .
          END;
```

```
SCHUTZ:   TASK;  /* NETZSCHUTZBEARBEITUNG */
          .
          .
          ACTIVIATE (BEFAUS);  /* FALLS NETZSCHUTZFALL ERKANNT
                                  WIRD */
          .
          .
          END;
```

```
BEFAUS:   TASK;  /* AUSGABE VON SCHALTBEFEHLEN ZUR FERNWIRK-
                    PERIPHERIE */
          .
          .
          END;
```

Bild 6: Globale Steuerung für die Netzschutzaufgabe bei SMA