

AUFBAU UND ERPROBUNG EINER MODULAR

STRUKTURIERTEN PORTABLEN

BASIC-PEARL-ANWENDER-PROGRAMMBIBLIOTHEK

E. Welfonder, Th. Röhrich und H. Sternad

Universität Stuttgart

Abteilung Stromerzeugung und Automatisie-
rungstechnik (IVD)

1. Einführung

Die Implementierung der Prozeßrechnersprache PEARL (Process and Experimental Automation Realtime Language) ist inzwischen so weit fortgeschritten, daß eine Anwendung von PEARL /1,2/ auf breiter Front möglich geworden ist. So bieten die deutschen Rechnerhersteller ausreichend ausgetestete PEARL-Systeme an und auch führende ausländische Rechnerhersteller haben bereits PEARL implementiert. Zudem ist im Dezember 1979 der PEARL-Verein als Interessengemeinschaft von Rechnerherstellern und PEARL-Anwendern gegründet worden /3/.

Der wesentliche Vorteil der Prozeßrechnersprache PEARL besteht - neben der Problemorientiertheit und der modularen Strukturierbarkeit von PEARL - in der Programm-Portabilität, d.h. der Problemteil eines PEARL-Anwender-Programmes ist ohne Modifikation auf verschiedenen Prozeßrechnern lauffähig, während im Systemteil des PEARL-Anwender-Programmes ausschließlich die Namen der speziellen hardwareseitig angeschlossenen Prozeß- und Geräteperipherie eingesetzt werden müssen. Diese Portabilität wird von allen PEARL-Implementatoren bezüglich Basic-PEARL /1/, das eine wesentliche Untermenge von Full-PEARL /2/ darstellt, erfüllt. Aufgrund der Programm-Portabilität ergeben sich die folgenden Möglichkeiten: zum einen können einmal in PEARL erstellte Anwenderprogramme leicht von einem Prozeßrechnertyp auf einen anderen Rechnertyp - auch unterschiedlichen Fabrikats - übertragen werden. Eine derartige Übertragbarkeit ist bei der bislang üblich gewesenen Assembler-Programmierung nicht möglich, weshalb beim Übergang auf einen neuen Rechnertyp die für den bisherigen Prozeßrechner - häufig mühsam - erstellten Anwenderprogramme nicht weiterverwendet werden können. Zum anderen erweist es sich aufgrund der Programmportabilität als sinnvoll, eine modular strukturierte PEARL-Anwender-Programmbibliothek aufzubauen, die eine Vielzahl in sich abgeschlossener Programmbausteine für häufig wiederkehrende Teilaufgaben der Prozeßleittechnik beinhaltet. Mittels einer derartigen Programmbibliothek kann der Anwendungsfachmann sein spezielles Anwenderprogramm durch Verwendung fertiger Programmbausteine in ähnlich einfacher Weise

aufbauen, wie dies bei der Erstellung wissenschaftlicher Programme durch den Zugriff auf Bibliotheksprogramme seit langem üblich ist.

Eine derartige PEARL-Anwender-Programmbibliothek ist von der Abteilung Stromerzeugung und Automatisierungstechnik (IVD) der Universität Stuttgart auf der Grundlage von Basic-PEARL erstellt worden. Dabei konnte die Abteilung auf PEARL-Erfahrungen aufbauen, die sie seit 1973 bei der PEARL-Implementierung im Rahmen der Arbeitsgemeinschaft PFK (PEARL für Kleinrechner) sowie der ASME (Arbeitsgemeinschaft Stuttgart-München-Erlangen) /4,5/, sowie bei der Erstellung von PEARL-Testprogrammen /6,7/ gewonnen hat. (Das erstellte Testprogrammpaket ist von der Mehrzahl der deutschen und ausländischen Rechnerhersteller zum Austesten ihrer PEARL-Implementationen herangezogen worden.) Ferner verfügt die Abteilung Stromerzeugung und Automatisierungstechnik aufgrund diverser Prozeßrechneranwendungen an energie- und verfahrenstechnischen Prozessen über das für den Aufbau einer derartigen Prozeßrechner-Programmbibliothek erforderliche Prozeß-Know-how.

2. Aufbau der Programmbibliothek

Der generelle Aufbau der Basic-PEARL-Anwender-Programmbibliothek ist in Bild 1 veranschaulicht. Die vom Prozeß kommenden analogen und digitalen Rohwerte werden mittels geeigneter Programmbausteine aufbereitet und als Fertigwerte in der Prozeßdatenbank abgelegt. Mittels der Fertigwerte können unter Verwendung vorhandener Programmbausteine die verschiedensten Aufgaben der Prozeßdatenverarbeitung ausgeführt werden, wie z.B.:

- Prozeßdaten-Reduktion mittels statischer bzw. dynamischer Modellbildung
- Prozeßüberwachung
- Prozeßan- und Abfahrsteuerungen
- Prozeßregelung und
- Prozeßoptimierung.

Die Datenausgabe kann über entsprechende Programmbausteine im open-loop-Betrieb in Form von Protokollen bzw. anhand von Monitor-Bildern erfolgen sowie im closed-loop-Betrieb durch die Ausgabe von Stell-

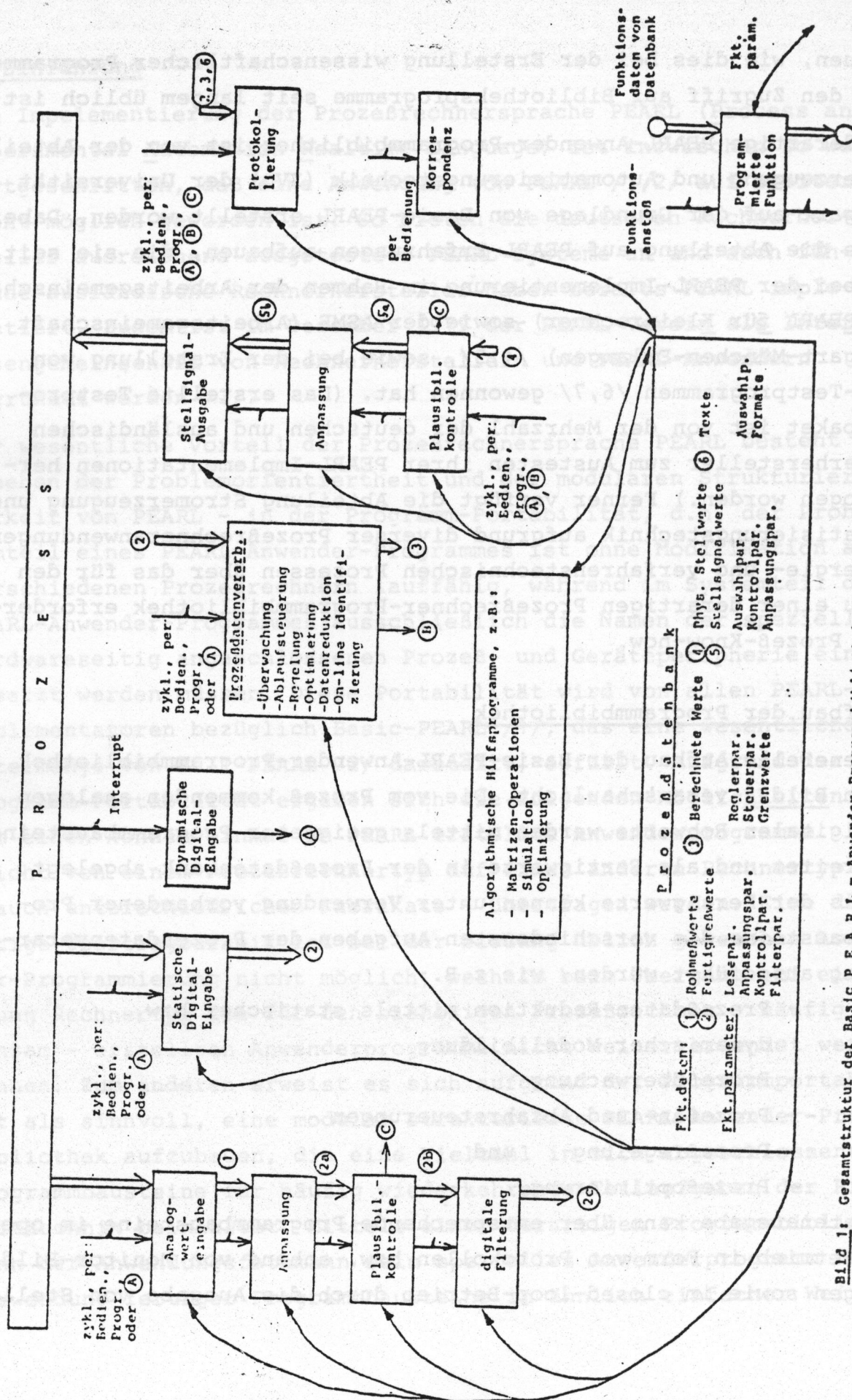


Bild 1: Gesamtstruktur der Basic PEARL - Anwender-Programmbibliothek

signalen.

Eine Übersicht über die erstellten Programmbausteine ist im Anhang zusammengestellt.

3. Eigenschaften der Programmbibliothek

Um die Programmbibliothek möglichst vielseitig einsetzen zu können, erfolgte ein modularer und hierarchischer Aufbau der Programmbibliothek auf der Grundlage leicht miteinander verknüpfbarer Programmbausteine. Aufgrund dieser Zielsetzung ergaben sich die folgenden Anforderungen an die Programmbibliothek:

- Flexibilität

Da die Anwendungsmöglichkeiten der einzelnen Bausteine erfahrungsgemäß mit ihrem Funktionsumfang abnehmen, wurden bevorzugt kleinere und mittlere Programmbausteine auf Prozedur- und Taskebene erstellt.

- Schnittstellenfestlegung

Um die Programmbausteine leicht miteinander verknüpfen zu können, erfolgte eine detaillierte Schnittstellenfestlegung für die einzelnen Programmbausteine.

- Dokumentation

Damit die Programmbibliothek zudem leicht handhabbar ist, sind Aufgaben und Schnittstellen der Programmbausteine ausführlich dokumentiert. Siehe Beispiel Bild 2.

- Test der Programmbausteine

Die einzelnen Programmbausteine sind anhand von Simulationsbeispielen sowie praktischen Erprobungen an ausgeführten Prozessen - siehe Abschnitt 4 - ausführlich getestet worden.

- Erweiterbarkeit

Seitens des Anwenders können auf einfache Weise weitere spezielle Bausteine in die Programmbibliothek eingefügt werden.

4. Erprobung

Zum on-line-Test der Basic-PEARL-Anwender-Programmbibliothek ist im Universitätsbereich Stuttgart-Pfaffenwald ein PEARL-Erprobungszentrum errichtet worden, bei dem - mit PEARL-Systemen ausgerüstete - Prozeßrechner verschiedener Hersteller über ein von Hartmann&Braun implementiertes PDV-Bussystem /14/ mit verschiedenartigen Prozessen

```

NLDF: PROCEDURE( ALTWERT( )  FLOAT IDENT,
                 NEUWERT( )  FLOAT IDENT,
                 TOLERANZ( ) FLOAT IDENT,
                 1           INV FIXED ) GLOBAL;

/*.....*/
/*.....*/
/*.....*/
/*      NICHTLINEARES DIGITALES FILTER      */
/*.....*/
/*.....*/
/* -ZWECK: DIE PROZEDUR NLDF ( NICHTLINEARE DIGITALE */
/*   FILTERUNG) GLAETZT EIN EINGANGSSIGNAL MITTELS  */
/*   EINES NICHTLINEAREN DIGITALEN FILTERS.        */
/*.....*/
/* -AUFRUF: CALL NLDF(ALTWERT,NEUWERT,TOLERANZ,1)  */
/*.....*/
/* -UEBERGABEPARAMETER:                          */
/*   ALTWERT  FLOAT  AUSGANGSGROESSE ZUM ZEITPUNKT  */
/*               (K-1) (EINGABEGROESSE)            */
/*   NEUWERT  FLOAT  EINGANGSGROESSE DES FILTERS ZUM  */
/*               ZEITPUNKT (K),                     */
/*               WIRD ALS GEFILTERTE GROESSE UEBER-  */
/*               GEBEN (EIN-/AUSGABEGROESSE)        */
/*   TOLERANZ FLOAT  UNEMPFINDLICHKEITSSZONE        */
/*               DES DIGITALEN FILTERS (EINGABEGROESSE) */
/*               TOLERANZ = 0.5*STOERAMPLITUDE      */
/*   1        FIXED INDEX DES MESSWERTES           */
/*               (EINGABEGROESSE)                   */
/*.....*/
/* -PLATZBEDARF: 301 * 16 BIT AUF SIEMENS 330      */
/*.....*/
/* -METHODE:                                         */
/*   DER MESSWERT (NEUWERT) WIRD WIE FOLGT GLAETZT: VER- */
/*   GLEICH DES NEUWERTES MIT DEM ALTEN GELAETZTEN MESS- */
/*   WERT (ALTWERT) ANHAND EINER MITGEFUEHRTEN UNEMP- */
/*   FINDLICHKEITSSZONE C (TOLERANZ).               */
/*.....*/
/*   NEUWERT := ALTWERT FALLS AENDERUNG KLEINER TOLERANZ */
/*               := NEUWERT - TOLERANZ                  */
/*               FALLS POSITIVE AENDERUNG > TOLERANZ    */
/*               := NEUWERT + TOLERANZ                  */
/*               FALLS NEGATIVE AENDERUNG > TOLERANZ    */
/*.....*/
/* -VERFASSER:                                       */
/*   H. STERNAD,M. ALT, IVO, UNIVERSITAET STUTTGART,  */
/*   ABT. STROMERZEUGUNG UND AUTOMATISIERUNGSTECHNIK  */
/*.....*/
/* -VERSION: 04.08.1978                             */
/*.....*/
/* -LITERATUR:                                       */
/*   WELFONDER E. UND A. LAMPART: NICHTLINEARES NACHLAUF- */
/*   FILTER ZUR UNTERDRUECKUNG DES NETZFREQUENZRAUSCHENS */
/*   BEI DER PRIMARREGELUNG.                          */
/*   VGB- KONFERENZ "FORSCHUNG IN DER KRAFTWERKSTECHNIK */
/*   1977", ESSEN                                     */
/*.....*/
/*.....*/

IF ABS(NEUWERT(1)-ALTWERT(1)) LE TOLERANZ(1)
THEN NEUWERT(1) := ALTWERT(1);
ELSE
  IF NEUWERT(1) GT (ALTWERT(1) + TOLERANZ(1))
  THEN NEUWERT(1) := NEUWERT(1) - TOLERANZ(1);
  ELSE NEUWERT(1) := NEUWERT(1) + TOLERANZ(1);
  FIN;
FIN;
ALTWERT(1) := NEUWERT(1); /* NULLPUNKTNACHFUEHRUNG */
RETURN;
END; /* PROCEDURE NLDF */

```

Bild 2: Basic PEARL Anwender Programmbibliotheks-
prozedur NLDF (nichtlineares digitales
Filter)

Literatur

zusammenwirken, siehe Bild 3 sowie /15/. Das PEARL-Erprobungszentrum bietet zudem die Möglichkeit, aus einzelnen Bausteinen der Programmbibliothek aufgebaute PEARL-Anwenderprogramme für die angeschlossenen Prozesse auf verschiedenen Prozeßrechnern laufen zu lassen und somit die Funktionsfähigkeit, Portabilität, Effizienz sowie Bedienkomfort der einzelnen PEARL-Systeme zu erproben. Über erste ausführliche Erprobungen wird in /16/ berichtet.

5. Zusammenfassung

Durch die Anwendung vorgefertigter und getesteter Programmbausteine aus der oben beschriebenen Basic-PEARL-Anwender-Programmbibliothek kann der Aufwand für die Prozeß-Software-Erstellung, besonders in den Projektphasen Programmsystementwurf, Codierung und Einzeltest unter Umständen erheblich gesenkt werden. Aufgrund der klaren Strukturierung der vorhandenen Programmbausteine kann ein Anwender-Prozeßautomatisierungsprogramm eine große Zahl dieser Bausteine enthalten, im Grenzfall sogar vollständig aus solchen Bausteinen aufgebaut sein.

GMR-Fachtagung "Prozeßrechner 1977"

/6/ Alt, M.: Programmpaket zum Testen von Basic-PEARL-Implementationen. GMR-Bericht KfV-PDV 155, 1978.

/7/ Alt, M.: Programmpaket zum Testen von Basic-PEARL-Implementationen. PDV-Bericht KfV-PDV 155, 1978.

/8/ Mast, T.: Programmiersprache PEARL zur internationalen Normung. Bericht des ISO eingereicht: Typ 21, 1979, Nr. 4, S. 10.

/9/ Dettmer, R. und M.: Einzigartige Ermöglichung viel steiler Leistungsgradienten durch struktur-optimal geregelte Kraftverteilung - Konzept in Heizkraftwerk der Universität Stuttgart. Brunnen-Wärme-Kraft 1977, Nr. 1.

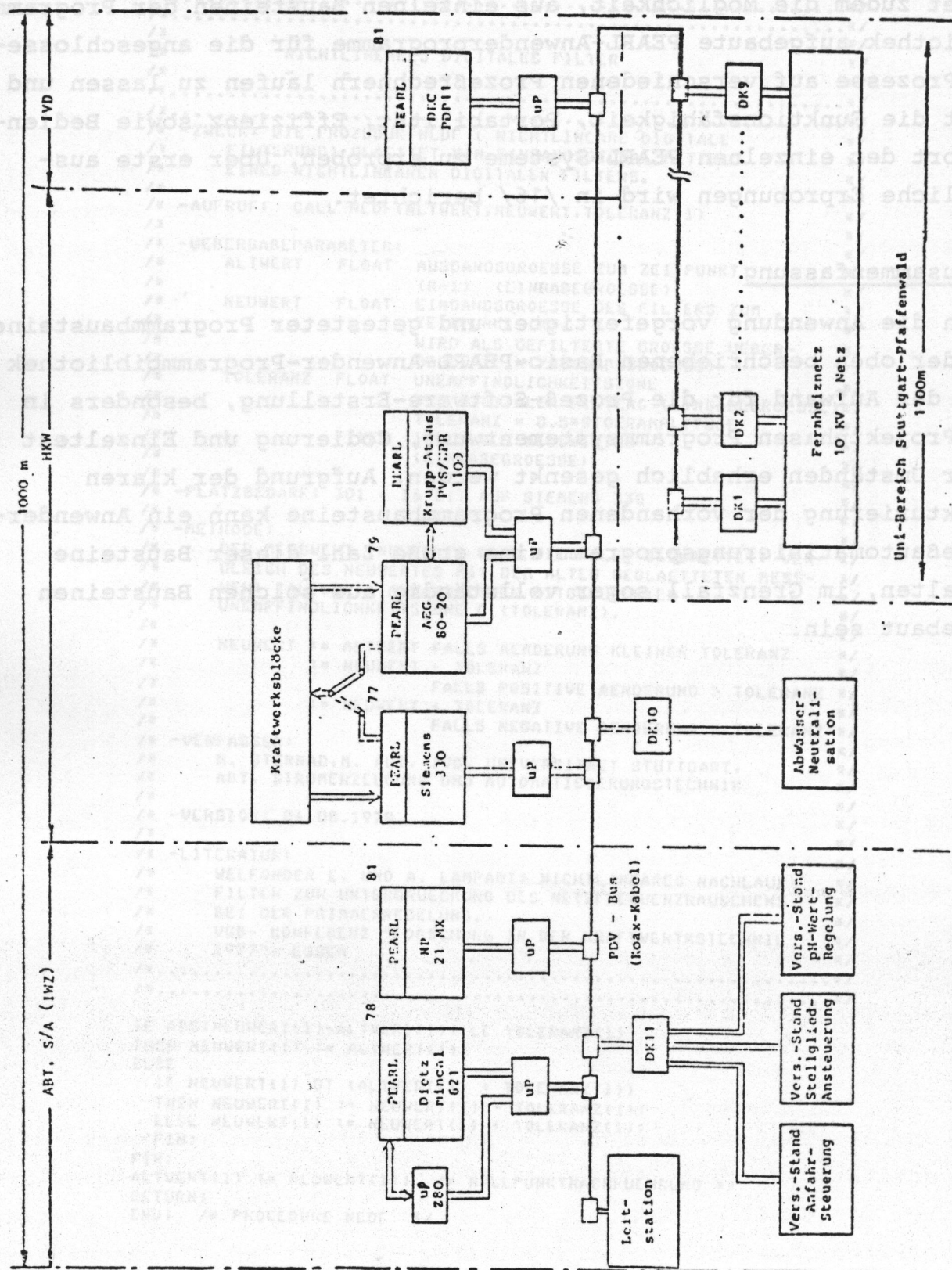


Bild 3: PEARL Erprobungszentrum
up = u-Processor Buskoppler

Literatur:

- /1/ Basic PEARL Sprachbeschreibung. PVD-Bericht KfK-PDV 121, Kernforschungszentrum Karlsruhe GmbH, 1977
- /2/ Full PEARL Language Description, PDV-Bericht KfK-PDV 130, Kernforschungszentrum Karlsruhe GmbH, 1977.
- /3/ Pressenotiz über die Gründung des PEARL- Vereins e.V. PEARL- Rundschau, Heft 1, Band 1, Seite 31.
- /4/ Kluttig, R., M. Alt: Beschreibung des Macroübersetzers STAGE 2 als Hilfsmittel zur Realisierung der Prozeß-rechnersprache PEARL auf dem Prozeßrechner Dietz 'mincal 621'. PDV-Entwicklungsnotiz PDV-E 63.
- /5/ Welfonder, E., M. Alt und J. Bühler: Implementierung der Prozeßrechnersprache PEARL für den Prozeßrechner AEG 60-10 und Anwendung an einem industriellen Prozeß. GMR-Fachtagung "Prozeßrechner 1977".
- /6/ Alt, M.: Programmpaket zum Testen von Basis-PEARL-Implementationen. rt 26 (1978), H. 8, S. 272-273.
- /7/ Alt, M.: Programmpaket zum Testen von Basic PEARL-Implementationen. PDV-Bericht KfK-PDV 155, 1978.
- /8/ Martin, T.: Programmiersprache PEARL zur internationalen Normung bei der ISO eingereicht. rtp 21, 1979, H. 4, S. 10
- /9/ Dettinger, R. und E. Welfonder: Ermöglichung viel steilerer Leistungsgradienten durch struktur-optimal geregelte Kraftwerksblöcke - erprobt im Heizkraftwerk der Universität Stuttgart. Brennstoff-Wärme-Kraft 1977, Nr. 1

- /10/ Welfonder, E., T. Beyerle und M. Alt: Kontinuierliche Abwasserneutralisation mittels eines Kleinprozeßrechners. rtp, Heft 4 + 5, 1978.
- /11/ Welfonder E. und A. Lampart: Nichtlineares Nachlauf-
filter zur Unterdrückung des Netzfrequenzrauschens
bei der Primärregelung. VGB- Konferenz "Forschung
in der Kraftwerkstechnik 1977", Essen.
- /12/ Welfonder, E. und F. Heilemann: Experimentelle Ermittlung
der Frequenz- und Spannungsabhängigkeit von Verbraucherteil-
netzen. VDE-Kongreß '78, Hannover.
- /13/ Welfonder E.: Regeldynamisches Zusammenwirken von
Kraftwerksblöcken und Verbrauchern im Netzverbund-
betrieb. Elektrizitätswirtschaft, Sept. 1980.
- /14/ Serielles Bussystem für industrielle Anwendungen unter
Echtzeitbedingungen (PDV-Bus) - Entwurf für einen Stan-
dardisierungsvorschlag - erarbeitet im Arbeitskreis zur
PDV-Bus Implementierung (API). PDV-Bericht KfK-PDV 150,
1978.
- /15/ Kluttig R. und E. Welfonder: Universelles PDV- Bus-
Interface. Elektronik Heft 25, 1979, Seite 73ff.
- /16/ Alt M.: Erprobung der Prozessrechnersprache PEARL
anhand der optimalen Mehrgrößenregelung von Kraft-
werksblöcken im HKW der Universität Stuttgart.
(erscheint in Kürze).

IVD Stuttgert

BPP Programmbausteine 11.3.80

Anhang:Uebersicht ueber die verfuegbaren ProgrammbausteineAusfuehrende Stelle:

Abteilung Stromerzeugung und Automatisierungstechnik (IVD)
 (Lt.: Priv.- Doz. Dr. Ing. E. Welfonder)
 Universitaet Stuttgart
 7 Stuttgart-80, Pfaffenwaldring 9

Sachbearbeiter: Dipl. Ing. H. Sternad Tel. 0711/7846208
 Dipl. Inform. Th. Roehrich 0711/7846203

Stand: 10.8.1980

1. Prozessschnittstelle1.1 Analoge Messsignale

- | | |
|---------------------|---|
| - ADERF (Modul) | Analog Daten Erfassung |
| - DIALOG (Task) | Dialogtask fuer die Parameterlisten-
erstellung. |
| - HAUPT (Task) | Steuertask fuer die Analogwert-
fassung und Protokollierung. |
| - HOLEN (Task) | Erfassungstask |
| - SPLOT (Task) | Steuertask fuer die Analogwert-
Protokollierung. |
| - AWLES (Prozedur) | Analogwert lesen |
| - AWAUFB (Prozedur) | Analogwert aufbereiten |
| - NLDF (Prozedur) | Nichtlineares digitales Filter |
| - AFLES (Prozedur) | Analogfeld lesen |
| - AFAUFB (Prozedur) | Analogfeld aufbereiten |
| - NLDF (Prozedur) | Nichtlineares digitales Filter fuer Feld |

1.2 Statische Binaersignale

- | | |
|------------------|---|
| - DIGERF (Modul) | Digitale Daten Erfassung |
| - DERFT (Task) | Task fuer die Erfassung der digitalen
Messgroessen |

2. Prozessdatenverarbeitung2.1 Datenreduktion

- | | |
|------------------|--|
| GLDFI (Prozedur) | Gleitender Durchschnitt von Fixed Groessen |
|------------------|--|

IVD Stuttgart

BPP Programmbausteine 11.8.80

2.2 Regelung

- | | | |
|---------|------------|--|
| - PIDU | (Prozedur) | PID- Stellungsalgorithmus |
| - PIDDU | (Prozedur) | PID- Inkrementalalgorithmus |
| - PIDUF | (Prozedur) | PID- Algorithmus mit Filterung der Eingangsgroesse |

3. Bedienerchnittstelle

3.1 Graphische Ausgabe

- | | | |
|----------|------------|---|
| - PLOT | (Prozedur) | Aussage von Kurvenpunkten auf Drucker |
| - PLOTZW | (Prozedur) | dto. fuer mehrdeutige Funktionen |
| - PLOTIL | (Prozedur) | Ploten von Hoehenlinien von Funktionen zweier Variablen |
| - KOPF | (Prozedur) | Listenkopf fuer PLOT und PLOTZW |

3.2 Bedienung

- | | | |
|---------|------------|--|
| - LEX | (Prozedur) | Lexikalische Analyse von einfachen in Grenzen frei wahlbaren Bediensprache |
| - PSYN | (Prozedur) | Partielle Syntax- Analyse von einfachen in Grenzen frei wahlbaren Bediensprachen |
| - PSYN2 | (Prozedur) | wie PSYN, jedoch Kurzformen der Schlüssellwoerter moeglich. |
| - KWF | (Prozedur) | Konvertierung von 6 Char Wort in fixed Zahl. |

3.3 Protokolle

- | | | |
|--------|----------|--|
| - PROT | (Module) | Modul zur Erstellung eines Warn- Stoer- und Schalterprotokolles sowie von Stoerablauf- Messwertverfolgungs- und Anlagenzustands- protokollen |
| - DT | (Task) | Drucktask fuer einen Protokolldrucker |
| - SAP | (Task) | Task zur Erstellung von Stoerablauf- protokollen. |

4. Datenverwaltung

4.1 Puffersysteme

Seitenverwaltendes Meldungspuffersystem:

- | | | |
|--------|------------|---|
| - IMP | (Prozedur) | Initialisieren des Meldepuffersystems |
| - EMMS | (Prozedur) | Eintragen einer Meldung in Meldepuffer- Seite |
| - LMMP | (Prozedur) | Loeschen einer Meldung in Meldepuffer- System |
| - WMPS | (Prozedur) | Wechseln der Meldepuffer- Seite |
| - HMPS | (Prozedur) | Hole Meldung aus Puffer- Seite |
| - HSMP | (Prozedur) | Hole Seite aus Meldungspuffer System |

IVD Stuttgart

BPP Programmbausteine 11.8.80

4.2 Textaufbereitung

- MTA	(Prozedur)	Meldungstext Aufbereitung
- MTE	(Prozedur)	Meldungstext Einspeicherung
- WOTAG	(Prozedur)	Aufbereitung von Wochentagstexten
- DATUM	(Prozedur)	Datumsaufbereitung

5. Mathematische Tasks und Prozeduren

5.1 Matrizen und Vektoren

- VEKTAD	(Prozedur)	Vektor Addition
- VEKTSU	(Prozedur)	Vektor Subtraktion
- VEKTMU	(Prozedur)	Vektor Multiplikation
- MATMU	(Prozedur)	Matrizen Multiplikation
- RMATMU	(Prozedur)	Matrizen Multiplikation lineare Form
- MINV	(Prozedur)	Matrix invertieren nach Gauss-Jordan
- LINV	(Prozedur)	Linear gespeicherte Matrix invertieren
- GMTA	(Prozedur)	Zweidimensionale Generalmatrix transponieren
- LMTA	(Prozedur)	dto. lineare Form
- GTPRO	(Prozedur)	Generalmatrix transponieren
- LTPRO	(Prozedur)	Generalmatrix transponieren lineare Form
- GTPRO2	(Prozedur)	Generalmatrix transponieren
- LJAPU	(Prozedur)	Loesung der Liapunov- Gleichung
- PRMAT	(Prozedur)	Ausdrucken einer Matrix
- RPRMAT	(Prozedur)	Ausdrucken einer Matrix reduzierte Form
- CONVM	(Prozedur)	Vektor in Matrix konvertieren
- CONMV	(Prozedur)	Matrix in Vektor konvertieren

5.2 Simulation

- DISKSIM	(Prozedur)	Diskrete Simulation
- AEQDIS	(Prozedur)	Diskretisierung mit äquidistanten Zeitintervallen
- GZUF	(Prozedur)	Zufallsgenerator fuer gleichverteilte Zufallszahlen.
- NZUF	(Prozedur)	Zufallszahlengenerator fuer (0,1) normalverteilte Zufallszahlen.

fuer Modellfolgerregelkreise:

- SWDM	(Prozedur)	Berechnung der Steuermatrizen fuer Modellfolgerregelkreise
- DISF	(Prozedur)	Diskrete Simulation des Fuehrungssystems
- DISP	(Prozedur)	Diskrete Simulation des Ausgangsvektors des gefuehrten Systems.

5.3 Optimierungsverfahren

- SMFP	(Prozedur)	Optimierung nach Fletcher und Powell
- SMP0	(Prozedur)	Optimierung nach Powell

IVD Stuttgart

BPP Programmbausteine 11.8.80

- SMKG (Prozedur) Optimierung mit konjugierten Gradienten

5.4 Berechnung von Zustandsreglern

- DOPTF (Prozedur) Berechnung eines quadratisch optimalen Zustandsreglers fuer spezielle Fuehrungsgrösse (Modellfolgesystem).
- SWDM (Prozedur) Berechnung der Stoersgrössenaufschaltung sowie der stationaeren Anteile fuer Modellfolgerregelungen.
- DISF (Prozedur) simuliert den disk. Verlauf des Ausgangs- und Steuervektors des Fuehrungssystems eines Modellfolgerregelkreises.
- DISP (Prozedur) simuliert sereregelte oder unregelte Systeme bzw. Modellfolgerregelkreise.
- DISOPT (Prozedur) Berechnung eines quadratisch optimalen Zustandsreglers

5.5 Zustandsbeobachter

- VLBOB (Prozedur) Vollstaendiger Lueneberger Beobachter

6. Prozessabhaengige Bausteine

6.1 Abwasserneutralisation

- PHREG (Modul) Kontinuierliche Abwasserneutralisation