

Bundesministerium für Forschung und Technologie

Forschungsbericht

Kernforschung

Überlegungen zur Definition einer
experimentorientierten Programmiersprache

von

P. Elzer

P. Holleczeck

Physikalisches Institut III
der Universität Erlangen-Nürnberg

34 Seiten
-- Abbildungen
1 Tabelle
45 Literaturstellen
3 Anhänge

Mai 1974

Übersicht:

Ausgehend von den Erfahrungen beim Einsatz von on-line-
Rechnern im physikalischen Experiment wurde ein Vorschlag
für eine höhere Programmiersprache für Realzeitanwendungen
erarbeitet.

Der Anwendungsbereich des fertigen Sprachentwurfs erstreckt
sich auf alle Gebiete der Echtzeit-Datenverarbeitung, da
die Definition der Sprache in einer Arbeitsgruppe aus Ver-
tretern verschiedener Firmen und Institute geschah, die
mannigfaltige einschlägige Erfahrungen besaßen.

Nachdem der eigentliche Sprachvorschlag bereits in verschie-
denen Veröffentlichungen vorgestellt wurde ([1], [2], [3]),
wird in diesem Bericht vorwiegend auf die politischen und
organisatorischen Aspekte des Vorhabens eingegangen.

Summary:

Starting from the experience with the use of on-line computers
in nuclear physics experiments a proposal for a higher level
language for realtime applications was worked out.

The application area of the complete proposal however
covers all fields of realtime data processing, since the
language was defined by a working group consisting of
representatives of different companies and institutes
who had a broad scope of experience on the subject.

Because the language proposal itself was already described
in various publications ([1] , [2], [3]), this report
is mainly concerned with political and organizational
aspects of the project.

I n h a l t

0	Einleitung	3
1	Die Entstehung von PEARL	4
1.1	Allgemeines zur Entwicklung von Realzeitprogrammiersprachen	4
1.2	Die Sprache PEARL	5
1.3	Der PEARL-Kreis	6
1.4	Der PDV-Subsetarbeitskreis SAK	7
2	Grundgedanken der PEARL-Entwicklung	7
2.1	Auswahlprinzipien	7
2.2	Wesentliche Elemente	10
3	Die Rolle von PEARL in der internationalen Standardisierung	13
3.1	Die internationalen Standardisie- rungsbemühungen	13
3.2	Der PURDUE-Workshop	14
3.3	Der europäische Beitrag	16
3.4	Die Rolle von PEARL	20
4	Versuch einer Bewertung der PEARL- Entwicklung	21
4.1	Die globale Zielsetzung	21
4.2	Die Bedeutung der Gruppenarbeit	21
4.3	Der Gewinn an "know-how"	24
4.4	Der Modellcharakter der PEARL-Entwicklung	25
4.5	Schlußbemerkung	26

5 Bibliographie	28
5.1 Algorithmische Sprachen	29
5.2 Real-Time-Sprachen	30
5.3 Standardisierung	33
5.4 Übersichten	34

Anhang A

E. Eichenauer, V. Haase, P. Holleczek, K. Kreuter
G. Müller: PEAFI, eine prozeß- und experiment-
orientierte Programmiersprache; Angewandte Infor-
matik, 9 (1973), S. 363-372

Anhang B

Anwendungsbeispiel

Anhang C

Auszug aus dem Protokoll einer Sitzung des PEARL-Kreises

Ø Einleitung

Der vorliegende Bericht ist zu einem Teil der Abschlußbericht für das vom Bundesministerium für Forschung und Technologie (BMFT) unter der Bezeichnung "NDV 13" geförderten Vorhaben:

"Überlegungen zur Definition einer experimentorientierten Programmiersprache".

Zum anderen ist er ein Zwischenbericht für das unter der Bezeichnung "NDV 15" geförderte Vorhaben:

"Implementation von PEARL".

Das erste Vorhaben hatte ursprünglich eine Laufzeit bis zum 31.12.1972. Es führte ziemlich frühzeitig zu einer vorläufigen Festlegung [1] der Eigenschaften der Sprache

PEARL (= Process and Experiment Automation Realtime Language).

Als sich die Notwendigkeit einer Testimplementation herausstellte, wurde das erste Vorhaben vorzeitig am 30.6.1971 beendet und in das zweite übergeführt. Diese organisatorische Zäsur stellte jedoch keine Unterbrechung der sachlichen Arbeit dar.

Nach dem Erscheinen des PEARL-Vorschlags [2] kann nun der "theoretische Teil" der Arbeit an PEARL als abgeschlossen betrachtet werden. Es wurde deshalb der jetzige Zeitpunkt gewählt, um einmal diese Aspekte der Arbeit im Zusammenhang darzustellen, d.h. die Entstehung der Programmiersprache, ihren Stellenwert im Felde des "software-engineering" und ihre Vermaschung mit internationalen Bemühungen um eine standardisierte höhere Programmiersprache für Realzeitanwendungen.

Eine Schilderung der mehr praktischen Aspekte, der Implementationsvorarbeiten und der während der Implementierung von PEARL gewonnenen Erfahrungen soll dem Schlußbericht des Vorhabens NDV 15 vorbehalten bleiben.

Anschließend sei noch eine kurze Bemerkung zur Form des vorliegenden Berichtes erlaubt:

Bei der Entwicklung und Verbreitung von PEARL ist bereits eine große Zahl von Veröffentlichungen, Berichten und Vorträgen entstanden. Es erschien deshalb wenig sinnvoll, alle in diesen Einzeldarstellungen schon enthaltenen Details zu wiederholen. Da zudem die Entwicklung von PEARL eine Gruppenleistung ist, wurde für diesen Bericht eine Form gewählt, die diese Tatsache zur Geltung bringt:

Die neueste einführende Darstellung von PEARL [3] und ein kurzes Programmbeispiel bilden den Anhangteil. Im Bericht selbst werden lediglich skelettartig die wichtigsten Grundlinien von PEARL skizziert. In der Bibliographie (Kap. 5) werden weitere Gesamt- und Einzeldarstellungen über PEARL nachgewiesen.

1. Die Entstehung von PEARL.

Anlässlich des Beginns der Prüfung von PEARL auf Normwürdigkeit und Normfähigkeit durch einen Arbeitskreis des Arbeitsausschuß 5 des Fachnormenausschuß Informationsverarbeitung im Deutschen Normenausschuß (DNA/FNI-AA5) wurde eine kurze Darstellung der Situation verfaßt und dem FNI überreicht [4] . Aus dieser Arbeit sind die folgenden einführenden Kapitel entnommen: (Beginn des Zitats)

1.1. "Allgemeines zur Entwicklung von Realzeitprogrammiersprachen"

Seit Mitte der 60-er Jahre versucht man, eine höhere Programmiersprache - ähnlich FORTRAN [22] oder ALGOL [23] - für Anwendungen in der Realzeitprogrammierung zu schaffen. Dabei handelt es sich zunächst vornehmlich um ad-hoc-Lösungen für spezielle Anwendungen.

Die Entwürfe reichen von reinen Sprachvorschlägen theoretischer Natur (z.B. verschiedene Sprachentwürfe mit dem Arbeitstitel "RTL" (Realtime language) [5]^x, [6] bis zu kompletten lauffähigen Programmiersystemen, die im Handel erhältlich sind (z.B. INDAC-8 von Digital Equipment Corp. [7]). Die in der Praxis eingesetzten Sprachen sind zumeist FORTRAN-Erweiterungen mit unterschiedlichem Umfang und wechselndem Erfolg [8] .

Erst gegen Ende der 60-er Jahre (1969-70) tauchten - vor allem in Europa - Sprachentwürfe auf, die vom Ansatz her für allgemeine Anwendungen bestimmt waren. Hierunter fallen u.a.:

RTL/1 [9], RTL/2 [10] (beide England)

LAI (= language d'applications industriels) [11], und

PROCOL (= process control language) [12] (beide Frankreich)

PAS1 (= Prozeßautomatisierungssprache [13] und auch

PEARL (beide Deutschland).

1.2 Die Sprache PEARL

PEARL - Process and Experiment Automation Realtime Language - ist eine Realzeitsprache, deren Entwicklung auf drei unabhängig voneinander entstandenen Vorstudien basiert ([14], [15], [16]). Ein erster Rohentwurf der Sprache PEARL datiert vom Herbst 1970:

"PEARL - the concept of a process and experiment automation language" [1]. Nach äußerst umfangreichen Detailausarbeitungen wurde der fertige Sprachentwurf als Arbeitsunterlage für den Implementator im Frühjahr 1973 als "PDV-Bericht" veröffentlicht [2]. Eine allgemeine Einführung ist in Vorbereitung. Sie wird in der Zeitschrift "angewandte informatik" veröffentlicht werden. (xx)

^{x)} Die Nummern der Zitate wurden gegenüber dem Original geändert, um innerhalb des Schlußberichts eine einheitliche Numerierung durchhalten zu können.

xx) Diese Veröffentlichung ist inzwischen erschienen [3]

1.3. Der PEARL-Kreis

Entwickelt wurde PEARL von einer deutschen Arbeitsgruppe, dem sog. "PEARL-Kreis". Die Zusammensetzung dieser Gruppe aus Herstellerfirmen, Anwendern, Softwarehäusern und Instituten erwies sich als gut ausbalanciert. Sie gewährleistete eine angemessene Berücksichtigung der verschiedensten Anwendungs- und Erfahrungsbereiche bei der Definition der Sprache.

Gegründet wurde der PEARL-Kreis im Herbst 1969 im Rahmen der "Studiengruppe Nuklearelektronik", einer vom BMW finanziell geförderten Arbeitsgruppe zur Verfolgung der Entwicklung hochwertiger Meßelektronik. Auch die Arbeit des "PEARL-Kreises" wurde vom BMW (und später vom BMT) finanziell unterstützt, vor allem durch am Physikalischen Institut III der Universität Erlangen-Nürnberg laufende Forschungsvorhaben. (x)

Der PEARL-Kreis wuchs sehr schnell und hatte Mitte 1970 bereits ca. 25 Mitglieder.

Er verstand sich stets als sachliches, allen interessierten Fachleuten offenes Gremium, eine Haltung, die ihren sichtbaren Ausdruck u.a. besonders deutlich in den Beziehungen des PEARL-Kreises zum VDI/VDE fand und findet:

Die Zusammenarbeit mit dem VDI/VDE, besonders mit dem UA "Programmiertechnik", war von Anfang an ausgezeichnet, was sich besonders auch dadurch ausdrückte, daß der Obmann dieses Ausschusses (Herr Dr. Heller) ständiger Gast im PEARL-Kreis war.

Seit 1973 bildet der PEARL-Kreis innerhalb der VDI/VDE "Gesellschaft für Meß- und Regelungstechnik" den Ausschuß "Echtzeitprogrammiersprachen" (Obmann: Herr Elzer, Univ. Erlangen-Nürnberg in der Sektion "Technik der Prozeßrechner" (Sektionsleiter: Dr. Ankel, BASF) dieser Gesellschaft.

x) Anm. zum Zitat: Dies sind die Vorhaben NDV13 und teilweise NDV 15 die der vorliegende Bericht zum Gegenstand

1.4. Der PDV-Subsetarbeitskreis SAK

Besondere Förderung erfährt die Arbeit an PEARL seit Anfang 1972 durch das "Projekt PDV" (= Prozeßdatenverarbeitung) im Rahmen des 2. DV-Programmes der Bundesregierung:

Seit Ende 1972 sind mehrere durch PDV geförderte PEARL-Implementationen in Arbeit. Dadurch soll eine intensive praxisbezogene Erprobung der Sprache und ihr genügend breit gestreuter Einsatz gewährleistet werden.

Zur Koordination der diesbezüglichen Arbeiten an PEARL wurde der "Subset-Arbeitskreis" SAK unter Vorsitz der Projektleitung PDV in der GfK gegründet. Neben den von PDV geförderten PEARL-Implementatoren gehört diesem SAK noch ein Vertreter der VDI/VDE-Gesellschaft für Meß- und Regelungstechnik an."
(Ende des Zitats).

2. Grundgedanken der PEARL-Entwicklung

2.1. Auswahlprinzipien

Ganz grundsätzlich kann eine Programmiersprache als Teilaspekt des "software engineering" betrachtet werden, d.h. als Bestandteil der Programmiermethode, die zur Lösung eines bestimmten Problems angewandt wird. Nach welchen Gesichtspunkten ist nun eine solche Programmiermethode auszuwählen?

Das Kernproblem ist die Entscheidung, ob eine höhere Programmiersprache verwendet werden soll und kann und wenn ja, welche Fähigkeiten eine solche Sprache haben soll.

Bisher wurden Realzeitprobleme praktisch ausschließlich in Assembler programmiert, da kein eindeutig besseres Verfahren zur Verfügung stand. Das war mit erheblichen Nachteilen verbunden.

- ⊙ Programme langwierig zu erstellen
- ⊙ schwierig zu testen
- ⊙ schwierig zu warten
- ⊙ schwierig zu ändern
- ⊙ schwierig zu übertragen.

Das Ergebnis ist, daß Programme für Realzeitanwendungen teuer sind. Hinzu kommt, daß die Kosten für die "hardware"beständig sinken, wodurch der Kostenanteil für die Programmierung ein immer größeres Übergewicht bekommt. Außerdem besteht nach wie vor ein Mangel an qualifizierten Programmierern für die angesprochenen Problemkreise, und die Kommunikation zwischen dem Entwerfer und dem Programmierer eines Systems ist, wie immer zwischen Spezialisten verschiedener Fachgebiete, schwierig. Eine solche Situation erfordert Abhilfe. Bisher existierte aber keine geeignete Alternative zur Assemblerprogrammierung, da die herkömmlichen Programmiersprachen das Gebiet der Realzeitprogrammierung ausklammerten. Das ergab für die damit erstellten Programme

- ⊙ mangelnde Problemangepaßtheit

woraus

- ⊙ mangelnde Effizienz

der Lösung resultierte.

Eine höhere Programmiersprache für Realzeit Zwecke gestattet nun eine wesentlich

- ⊙ verbesserte Problemangepaßtheit

und damit

- ⊙ verbesserte Effizienz

der Lösung gegenüber herkömmlichen Programmiersprachen.

Kriterien für die Auswahl einer Programmiermethode					
KRITERIUM	AUSWIRKUNG	ASS.	PAKET	SPRACHE	
1	Effizienz des fertigen Systems	+	(+)	++	
2		+	(+)	+	
3		+	+	(-)	
4		-	++	+	
5	Effizienz bei Erstellung eines Systems	-	++*	+	
6		-	++*	+	
7		-	-	+	
8		-	-	+	
9		-	-	(+)	
10		-	(-)	+	
11		--	+	+	
12		--	++*	+	

* beim Anwender

Tab. 1: Auswahlkriterien

Ein Beurteilungsschema (Fig. 1) soll in kurzer Form verdeutlichen, wie groß die Vorteile der höheren Programmiersprache (prozedurorientierte Programmiersprache) gegenüber den beiden anderen zur Auswahl stehenden Techniken, der Assemblerprogrammierung und dem Anwenderpaket, sind.

Weitgehende Überlegungen, insbesondere die Abgrenzung gegenüber den "niederen Sprachen", wie sie für die Systemprogrammierung mehr und mehr Verwendung finden, sind im Einleitungsteil der einführenden Darstellung von PEARL [3] enthalten, die diesem Bericht als Anhang A beigelegt ist.

Aus all den angeführten Gründen wurde PEARL als benutzerorientierte Programmiersprache der mittleren Ebene angelegt.

Die hauptsächlichlichen Vorteile einer derartigen Sprache sind:

- Leichte Erlernbarkeit und damit
Verwendbarkeit auch für den
Nicht-DV-Spezialisten
- Große Flexibilität
- Hohe Fehlersicherheit
- Hoher Dokumentationswert

2.2. Wesentliche Elemente

Eine ausführliche Beschreibung der Sprachelemente von PEARL erübrigt sich an dieser Stelle. Eine einführende Beschreibung liegt im Anhang vor und für den Implementator ist der ausführliche Sprachvorschlag [2] bestimmt.

Für den eiligen Leser sollen jedoch in einigen kurzen Tabellen die wesentlichen Bestandteile von PEARL hier zusammengefaßt und zu den einzelnen Beurteilungskriterien in Beziehung gebracht werden.

PEARL - Entwurf

SYSTEMBESCHREIBUNG
TASKSTEUERUNG
ERWEITERTER ALGORITHMISCHER TEIL
STANDARD E/A
PROZESS E/A
MODULSTRUKTUR

PEARL-SYSTEMBESCHREIBUNG

- | | |
|---|---------------------------|
| • Beschreibung der Anlagenkonfiguration auf Sprachebene | --- Dokumentationswert |
| • Beschreibung der Verknüpfung zwischen hardware und software auf Sprachebene | --- Rechnerunabhängigkeit |
| • Bereitstellung von Information für die Systemgenerierung auf Programmebene | --- Problemgerechtheit |
| • Bereitstellung symbolischer Namen für die Externgeräte | --- Sicherheit |

PEARL-TASKSTEUERUNG

- | | |
|--|---------------------------|
| • Explizite Steuerung von Tasks durch Anweisungen | --- Rechnerunabhängigkeit |
| • Implizite Steuerung von Tasks durch Programmstruktur | --- Sicherheit |
| • Beeinflussung der Priorität von Tasks | --- Systemgerechtheit |
| • Synchronisation von Tasks | --- Problemgerechtheit |
| • Ankopplung von Tasks an externe Ereignisse | --- Systemgerechtheit |

PEARL - ERWEITERTE
ALGORITHMISCHE

- ⊙ Bit- und Stringbearbeitung — Rechnerunabhängigkeit
- ⊙ Strukturen — Programmieraufwand
- ⊙ Adressen — Rechnerunabhängigkeit
- ⊙ Typ- und Operatorvereinbarungen — Programmieraufwand
- ⊙ Prüfbarkeit zur Übersetzungszeit — Sicherheit

PEARL-STANDARD I//

- ⊙ Formatvariable — Programmieraufwand
- ⊙ Einheitlichkeit — Erlernbarkeit
- ⊙ Symmetrie — Sicherheit
- ⊙ Geräteunabhängigkeit — Änderungsaufwand
- ⊙ File-handling-System — Problemgerechtigkeit

PEARL-PROZESS E/A

- ⊙ Ansteuerung nicht standardmäßiger Peripheriegeräte — Systemgerechtigkeit
- ⊙ Ein-/Ausgabe mit Buchung — Problemgerechtigkeit
- ⊙ Graphische Ein-/Ausgabe — Rechnerunabhängigkeit

PEARL-MODULSTRUKTUR

- ⊙ Möglichkeit zur getrennten Übersetzung von Programmteilen — Änderungsaufwand
- ⊙ Möglichkeit zum stückweisen Aufbau eines Programmsystems — Testaufwand
- ⊙ Möglichkeiten zur Kommunikation zwischen Programmen über globale Daten — Sicherheit

3. Die Rolle von PEARL in der internationalen Standardisierung

3.1. Die internationalen Standardisierungsbemühungen

Aus der Einsicht heraus, daß eine Programmiersprache nur dann lebensfähig ist, wenn sie auf genügend vielen Anlagen implementiert wird, wurde von Anfang der PEARL-Entwicklung an versucht, eine Zusammenarbeit auf möglichst breiter Basis zu erreichen. Insbesondere wurde schon frühzeitig Anschluß an die internationalen Standardisierungsbestrebungen gesucht und zwar in doppelter Absicht: Einmal sollte durch rechtzeitige Trendverfolgung eine Entwicklung vermieden werden, die am zu erwartenden internationalen Standard vorbeiführen könnte. Zum anderen wurde versucht, die in der PEARL-Entwicklung gewonnenen Erkenntnisse weitgehend in den Standardisierungsprozeß einzubringen, um die eingesetzten Mittel möglichst effektiv auch langfristig wirksam werden zu lassen.

Man kann ohne Übertreibung sagen, daß die PEARL-gruppe in der Zwischenzeit eine weitgehend aktive und treibende Rolle bei den internationalen Standardisierungsbestrebungen auf dem Gebiet der Realzeitsprachen spielt.

Interessant ist ferner, daß, obwohl die Notwendigkeit einer weltweit standardisierten Programmiersprache für Realzeitanwendungen in der Fachwelt praktisch unbestritten ist, sich die offiziellen Standardisierungsgremien erst in jüngster Zeit damit zu beschäftigen begannen. So wurde erst im April 1973 auf einer Tagung in Budapest das Arbeitsgebiet des "Subcommittee 9 of Technical Committee 97 der International Standards Organization" (ISO-TC-97-SC9, "Programming languages for numerical control") erweitert und lautet jetzt: "The standardization of programming languages for the numerical control of machines and for the control of industrial processes". So konzentrierte sich

die Arbeit weitgehend auf im Vorfeld der Normung tätige freiwillige und halboffizielle Gruppen.

Eine detaillierte "Momentaufnahme" der derzeitigen Situation auf dem Standardisierungsgebiet wird in [4] gegeben, woraus die folgenden Abschnitte entnommen sind:

"Als in dieser Beziehung aktivstes Gremium ist der "Purdue Workshop on Standardization of Industrial Computer Languages" (x) (im folgenden nur "Workshop" genannt) hervorzuheben, welcher seit April 1969 in halbjährlichem Turnus tagt. Initiator, Organisator und ständiger Vorsitzender ist Prof. Dr. T.J. Williams von der Purdue University, Lafayette, Indiana, USA."

Deshalb soll im folgenden näher auf dieses Gremium eingegangen werden (wieder unter Verwendung von [4]).

3.2. Der PURDUE-Workshop. (Beginn des Zitats)

"Ursprünglich ging man bei diesem Workshop davon aus, daß es lediglich gelte, die verschiedenen in den USA existierenden Prozeß-FORTRAN's [z.B. 27] aneinander anzupassen und zu standardisieren und eine einheitliche Terminologie auf dem Gebiet der Prozeßrechnerprogrammierung festzulegen.

Es stellte sich jedoch bald heraus, daß das Aufgabengebiet weit umfangreicher war, als man angenommen hatte, und es wurden fünf Ausschüsse gebildet:

1. Short term procedural language committee
(FORTRAN-Standardisierung)
2. Glossary committee (Terminologie)
3. Functional requirements committee
(Festlegung anwendungsbezogener technischer Anforderungen an Prozeßrechnersysteme)
4. Problem-oriented language committee
(Arbeiten an einer (oder mehreren) direkt problemorientierten Programmiersprache(n))

x) jetzt: "International Purdue Workshop on Industrial Computer Systems".

5. Long term procedural language committee (LTPL-C)
(Festlegung der Nachfolgesprache für Prozeß-FORTRAN).

Das LTPL-Komitee (LTPL-C)

Für die Arbeit des "long term procedural language committee" (= LTPL-C) gab es allerdings keine existierende Sprache, die eine derart hervorstechende Favoritenrolle zu spielen in der Lage gewesen wäre, wie dies beim Prozeß-FORTRAN der Fall war.

Ursprünglich wurden folgende Sprachen betrachtet:
JOVIAL 28 , SPL/76, ALCOL 23 , PL/I 24 ,
"Case-RTL" 17 , LISP, SNOBOL, GPSS, SIMSCRIPT.

Dazu kamen ab Herbst 1970 noch die in Europa entwickelten Sprachvorschläge, darunter auch PEARL. Das LTPL-C entwickelte sich schnell zum größten und aktivsten Ausschuß der Workshop, da bei den meisten beteiligten Institutionen auf diesem Arbeitsgebiet vitale Interessen berührt wurden und da zwischen einem Teil der US-amerikanischen und den übrigen Teilnehmern erhebliche Anschauungsunterschiede auftraten.

Die Diskussion vom PL/I und die Philosophie des "melt of best features"

Die Diskussionen entzündeten sich hauptsächlich daran, daß sich eine US-amerikanische Teilgruppe des LTPL-C Komitees recht frühzeitig auf die ausschließliche Verwendung und Anpassung von PL/I für Prozeßzwecke konzentriert und sich zu diesem Zweck als Unterausschuß X3J1.4 dem PL/I Normkomitee X3J1 der ANSI angeschlossen hatte.

Nach Ansicht der überwältigenden Mehrheit der anderen Workshop-Teilnehmer aus Europa, Japan und den USA gibt es für diese Entscheidung keine sachlich zwingenden Gründe. Vielmehr wird sie auf gewisse Eigeninteressen zurückgeführt.

Demgegenüber haben die Experten, insbesondere auf der Herbstsitzung 1972 des Workshops, festgestellt und eindeutig gefordert, daß die zukünftige LTPL-Sprache als "melt of best features" bereits existierender Kandidatensprachen zu erarbeiten sei [18] .

3.3 Der europäische Beitrag

Diese Ansichten wurden wortführend von den Europäern vorgetragen und verfochten. Es war jedoch sehr schwierig, den Standpunkt der Europäer angemessen zur Geltung zu bringen und die von ihnen gemachten technischen Erfahrungen in den Standardisierungsprozeß einzubringen.

Eine große Rolle spielte dabei das Fehlen einer gemeinsamen europäischen Organisation, die in Europa etwa die Rolle spielen könnte, wie es die ANSI (= American National Standards Institute) oder das NBS (= National Bureau of Standards) in den USA tun. Das Gewicht der nationalen Organisationen reicht meist nicht aus, um beispielsweise einer von der ANSI unterstützten Aktivität entgegenzuwirken, wenn dies nach Meinung europäischer Fachleute erforderlich sein sollte.

In Europa entstanden zwar im Laufe des Jahres 1972 einige Gruppen, die sich mit Prozeß-FORTRAN befassen, allerdings bisher auf nationaler Ebene (BRD, GB). Außerdem sind z.Zt. im Rahmen der EG-Behörden und verschiedener anderer europäischer Organisationen Bestrebungen im Gange, eben diese oben angesprochenen Zustände zu verbessern und Koordination und Kooperation auf europäischer Ebene zu verstärken. Solche Dinge erfordern aber naturgemäß sehr viel Zeit, bevor sie zum Tragen kommen.

"Das Komitee LTPL-E"

Um nun wenigstens auf dem aktuellen Gebiet einer zukünftigen Realzeitprogrammiersprache durch gegenseitige Abstimmung und gemeinsames Handeln im Bedarfsfall die Durchsetzungschancen der europäischen Vorstellungen zu verbessern, wurde anlässlich der dritten Sitzung des Workshop im Herbst 1970 eine europäische Sektion von LTPL-C gegründet (kurz LTPL-E genannt). Ihre erste Sitzung fand im Januar 1971 in einem Entwicklungslabor der Firma Shell in Den Haag statt.

Der erste Vorsitzende war Ivor Gray von ICI, England, der Leiter jener Gruppe, die den Sprachvorschlag RIL/2 entwickelt hatte. Aus Gründen der Arbeitsbelastung legte Herr Gray im Herbst 1971 dieses Amt nieder. Nachfolger als Vorsitzender im LTPL-E Komitee wurde Dipl.-Phys. P. Elzer von der Universität Erlangen-Nürnberg, der derzeitige Obmann des Ausschusses "Echtzeitprogrammiersprachen" in der VDI/VDE-Gesellschaft für Meß- und Regelungstechnik.

Zur Zeit umfaßt LTPL-E folgende Mitgliedsorganisationen:

AEG-Telefunken	D
British Steel Corporation	GB
Brown Boveri & Cie	D
C.A.P.	F
C.E.R.C.I.	F
Elektronik System Gesellschaft	D
FOXBORO	NL
GEC-Elliott	GB
Gesellschaft für Kernforschung Karlsruhe	D
ICI	GB
IDAS	D
IRIA	F

Magyar Tudomanyos Akademia	H
National Computing Centre	GB
PAN DATA	NL
Post Office Research Department	GB
Royal Radar Establishment	GB
SCS	D
SESA	F
SHELL	NL
SIEMENS	D
SINTEF	N
S.T.E.k.I.A.	F
University of Bradford	GB
University of Erlangen	D
University d'Orsay	F
University of Stuttgart	D
University of York	GB

Erfolge von LTPL-E

Durch gemeinsames Auftreten konnte LTPL-E auch bereits einige Erfolge in technischer und organisatorischer Hinsicht erzielen:

1. Es wurden "functional requirements" für eine zukünftige Realzeitsprache entwickelt und vom Workshop zum Beschluß erhoben [19]. Diese Liste technischer Anforderungen wird ständig weiterentwickelt.
2. Für das LTPL-C wurden gegenüber dem Workshop abgeänderte Statuten durchgesetzt, die eine schriftliche Abstimmung ermöglichen. Das ist besonders für nicht US-amerikanische Teilnehmer von großer Bedeutung [20].
3. PL/I wurde seiner einseitigen Favoritenstellung entkleidet und auf seine berechnete Kandidatenrolle reduziert [18].
4. In LTPL-E wurde ein detaillierter Vergleich einiger "candidate languages" begonnen. Dadurch sollen die Grundlagen für die Festlegung einer späteren Standardsprache geschaffen werden. Diese Arbeiten schreiten sehr zügig voran. Sie gewährleisten in vollem Umfang ein sachgerechtes und von jeder einseitigen Tendenz befreites Vorgehen im Sinne der Philosophie des "melt of best features".

Vergleichssprachen sind derzeit:

ALGOL 68 [25] , CORAL 66 [26] , PAS 1 [13] , PEARL [2] ,
PL/I [24] und PROCOL [12] .

In diesem Zusammenhang erhielt das LTPL-Komitee einen neuen, europäischen, Vorsitzenden - Herrn Johannes Reh, ESG, München. Gestützt auf den Konsens der Komitee-Mitglieder in Europa, Japan und den USA soll er nunmehr

- die Komitee-Arbeiten im Sinne der Philosophie des "melt of best features" fördern und
- die Zusammenarbeit mit allen weiteren, einschlägig interessierten und fachlich kompetenten Gremien anstreben und vorantreiben." (Ende des Zitats)

In jüngster Zeit ist zudem noch eine Zusammenarbeit mit der ESONE-CAMAC-Gruppe zustande gekommen. Damit ist auch die Basis für eine Arbeit gelegt, die Aspekte der standardisierten Hardware gleichrangig mit denen der Software behandelt, was für die Zukunft eine sehr fruchtbringende Kombination zu werden verspricht.

3.4 Die Rolle von PEARL

Zur Zeit sind also die Perspektiven auf dem Gebiet der europäischen und weltweiten Zusammenarbeit recht gut, jedoch sollten die dabei zu berücksichtigenden Zeitfaktoren nicht unterschätzt werden. Das führt zu der Notwendigkeit, die Zeit bis zur Verfügbarkeit eines internationalen Standards zu überbrücken.

Im Verlauf des im LTPL-E-Komitees begonnenen Sprachvergleichs hat sich sehr bald herausgestellt, daß PEARL von den derzeitigen Kandidatensprachen technisch am weitesten entwickelt ist und praktisch alle bisher veröffentlichten "functional requirements" [19] erfüllt.

Die Arbeiten an PEARL stehen auch in keinerlei Gegensatz zur Philosophie des "melt of best features". Vielmehr zielen sie auf erprobungs- und implementierungsorientierte Erkenntnisse ab, die der gesamten Fachwelt und natürlich insbesondere der beteiligten Industrie zu gute kommen.

Eine Beobachtung der gegenwärtigen Tendenz zur internationalen standardisierten Prozeßrechnersprache läßt den Schluß zu, daß wohl wesentliche Elemente von PEARL auch wesentliche Elemente dieser "LTPL" sein werden. Damit hätte sich die Arbeit an PEARL in mehrfacher Hinsicht gelohnt: Zum Einen sind die auf diesem Gebiet tätigen deutschen Firmen und Institute durch diese Arbeit in die vorderste Front des entsprechenden Fachgebietes gerückt. Zum Anderen wird aktiv Einfluß auf die internationale Standardisierung ausgeübt, was ein entsprechendes Ergebnis der eingebrachten Investitionen verspricht.

4. Versuch einer Bewertung der PEARL-Entwicklung

4.1 Die globale Zielsetzung

Wie man aus der Aufzählung anderer Sprachentwürfe in Abschnitt 1 ersehen kann, wurde bei der Entwicklung von PEARL nur an wenigen - dafür aber wesentlichen - Stellen absolutes technisches Neuland betreten. Das galt jedenfalls, wenn man jeweils einzelne Sprachbestandteile betrachtet, die in dieser oder jener Form schon in einem anderen Sprachentwurf vorhanden waren. Einige der in Abschnitt 2.2 beschriebenen Spracheigenschaften sind jedoch auch absolut neu (z.B. Systemteil, die Gesamtstruktur des Tasking und einige Teile der I/O). Ganz anders wird das Bild jedoch, wenn man PEARL in seiner Gesamtheit betrachtet: Es wurde zum ersten Mal versucht, den Stand der Technik auf dem Gebiet der Beschreibung von Prozessen und dem des Benutzerinterface von Realzeitbetriebssystemen geschlossen zusammenzufassen und in die Form einer bewußt benutzerorientierten Programmiersprache zu gießen.

Dabei sollte sich nicht nur eine Vereinfachung für den Benutzer dadurch ergeben, daß die Programmierung der allermeisten Probleme vereinfacht wird, sondern es sollte außerdem ein gewisser Druck auf die Ersteller von Betriebssystemen ausgeübt werden können in der Richtung, alle bereits zu vereinheitlichenden benutzerrelevanten Eigenschaften von Betriebssystemen auch in einheitlicher Form zu gestalten.

4.2 Die Bedeutung der Gruppenarbeit

Als wesentlichster Punkt der ganzen PEARL-Arbeit darf aber angesehen werden, daß das "PEARL-Projekt", wenn man es einmal so nennen soll, die einzige der vielen in Abschnitt I genannten Sprachentwicklungen ist, bei der es gelang, die Entwicklungsgruppe von der Grundidee bis zur Implementierung zusammenzuhalten.

Dazu kommt, daß es sich nicht um eine herkömmliche Entwicklungsgruppe innerhalb einer Firma oder einer festgefügteten Institution handelt, sondern um eine lose Vereinigung von Fachleuten verschiedenster Herkunft, die allein durch den sachlichen Zwang der zu leistenden Arbeit zusammengehalten wurden.

Ein solches teamwork großen Ausmaßes hat seine Vor- und Nachteile.

Der entscheidende Vorteil soll gleich zu Anfang genannt werden.:

Eine Programmiersprache, die nicht implementiert wird, ist tot geboren. Deshalb ist es notwendig, die vitalen wirtschaftlichen Interessen der Herstellerfirmen, die ja die hauptsächlichsten Implementatoren sind, von vornherein an das Projekt zu binden, um sozusagen die Lebensfähigkeit des Produktes zu "erzwingen". Dadurch wird aber auch gleichzeitig die Praxisbezogenheit des Entwurfs gewährleistet, da rein theoretische und vielleicht schwer zu verwirklichende Vorschläge mit wirtschaftlichen Nachteilen für den verbunden sind, der den Vorschlag macht. Insofern muß eine Gruppe, die eine Programmiersprache entwickelt, unbedingt Vertreter der maßgeblichen Herstellerfirmen enthalten.

Andererseits wird eine reine Gruppierung der Herstellerindustrie allein jedoch kaum in der Lage sein, ein Produkt wie PEARL zu verwirklichen, da vor allem ein gewisses (oft mißverstandenes und überbetontes) Konkurrenzdenken bewirkt, daß notwendige Diskussionen über Normierungsfragen - und die Entwicklung einer Programmiersprache gehört zu diesen - oft in unverbindlichen Willenserklärungen steckenbleiben. Deshalb ist die Beteiligung maßgebender Benutzer (= Kunden) notwendig. Im Fall PEARL waren es die großen Kernforschungsinstitute. In anderen Fällen mögen es andere Institutionen sein. Die Benutzer können auch der möglichen Gefahr der "Betriebsblindheit" bei den

Herstellern entgegenwirken und verhindern, daß die Qualität des Produktes "Compiler" nur an den minimalen Herstellungskosten gemessen wird - was ja bekanntlich nicht unbedingt eine Optimierung der Gesamtkosten des endgültigen Produktes "Programm" bedeuten muß.

Eine andere Gruppe von möglichen Interessenten auf dem Gebiet der Programmiersprachen kann heute nicht mehr vernachlässigt werden: die Softwarehäuser. Diese Firmen werden in Zukunft immer mehr an Bedeutung gewinnen und wahrscheinlich in dem Maße immer weitere Bereiche der Benutzerprogrammierung übernehmen, in dem die Herstellung der immer komplexer werdenden Grundsoftware die Herstellerfirmen mehr und mehr belastet. Deshalb sollten auch die Softwarehäuser - zumindest diejenigen, die sich nicht ausschließlich mit der Produktion direkt benutzerbezogener und deshalb allzu spezieller Software befassen, bei der Entwicklung einer Programmiersprache mit einem vorhersehbar breiten Anwendungsbereich beteiligt sein. Dadurch wird auch der Gefahr begegnet, daß die Herstellerfirmen die Definition des Produktes zu sehr auf die Eigenschaften ihrer gerade aktuellen Rechner ausrichten.

Bei diesen Erwägungen über den Aufbau einer Arbeitsgruppe muß man sich allerdings auch ständig einer Gefahr bewußt sein: der Gruppengröße. Auf der einen Seite ist zur Entwicklung und vor allem zur Durchsetzung eines Produktes vom Aufwand einer Programmiersprache eine gewisse "kritische Masse" notwendig und es ist auch nicht möglich, potentielle Interessenten, die einen wesentlichen Beitrag liefern könnten, auszuschließen. Auf der anderen Seite wird dadurch zwangsläufig schnell die arbeitsfähige Größe einer Gruppe (8-10 Teilnehmer) überschritten. Daß diese Tatsache zu gewissen Verzögerungen und Reibungsverlusten bei der PEARL-Arbeit geführt hat, soll nicht verschwiegen werden. Jedoch konnten durch die Aufteilung des Plenums in arbeitsfähige Untergruppen und die Ausbildung eines "harten Kernes" von Autoren mit einer starken persönlichen und firmenpolitischen Motivation die Arbeitsfähigkeit und die Kontinuität

der Gruppe doch im Endeffekt gesichert werden. Doch sollten gewisse Erfahrungen, die bei der PEARL-Arbeit gemacht wurden, bei ähnlichen Projekten in Zukunft berücksichtigt werden, besonders um nicht immer auf ein weit überdurchschnittliches Maß an persönlichem Einsatz einiger weniger Mitglieder angewiesen zu sein, wodurch eine solche Struktur leicht instabil werden kann.

4.3 Der Gewinn an "know-how"

Ein weiterer Aspekt des team-work sollte auch nicht unterbewertet werden: die mögliche Konzentration von know-how durch Zusammenarbeit der verschiedensten am Projekt beteiligten Fachleute. Es wird immer wieder eingewandt, daß eine Sprache, die von einem kleinen, geschlossenen, homogenen Team an einer Stelle entwickelt wird, besser sei, als der "committed designed elephant", wie sie bei derartig großen beschließenden Strukturen wie dem PEARL-Kreis zustandekämen. Doch darf man bei dem Versuch der Einführung eines derartigen Produktes nicht die menschliche Seite vergessen, die eine wesentlich größere Rolle spielt, als dies bei technischen Produkten gemeinhin angenommen wird. Und so sollte die Tatsache nicht außer acht gelassen werden, daß es für die endgültige Durchführung des Gemeinschaftsprojekts sehr nützlich sein kann, wenn Mitarbeiter von in Frage kommenden Firmen auch durch ein persönlich-technisches Engagement gebunden sind.

Diese Schilderung soll nun nicht den Eindruck erwecken, als ob alle angesprochenen Probleme bei der PEARL-Arbeit in optimaler Weise gelöst wurden. Man würde heute - nach Beendigung der Arbeit - manches etwas anders machen. Aber die Probleme wurden gelöst und die PEARL-Gruppe ist heute - weltweit gesehen - eine der ganz wenigen "überlebenden" Gruppen dieser Art, d.h. ihr Zusammenhalt konnte lange genug erhalten werden, um das entscheidende Stadium der gemeinsamen praktischen Verwirklichung zu erreichen, ohne daß das ganze Projekt eine reine akademische Fingerübung geblieben wäre.

4.4 Der Modellcharakter der PEARL-Entwicklung

Die Entwicklung von PEARL kann somit sehr wohl als ein Modellfall für ähnliche gelagerte Entwicklungen - nicht nur auf dem Computersektor - angesehen werden.

Zunächst einmal kann das PEARL-Projekt als Modell dafür angesehen werden, daß es möglich ist, Firmen in Fällen zur Zusammenarbeit zu bringen, wo dies aus Gründen der Verbesserung der internationalen Konkurrenzsituation nötig ist. Die Zusammenarbeit kann sehr wohl durch staatliche Initiative bewirkt werden, jedoch nicht durch reines Dekret oder Vorschlag, sondern durch Formulierung eines konkreten Projekts, das gewisse Eigeninteressen der Firmen anspricht. Nur muß die Zusammenfassung der verschiedenen Interessenten rechtzeitig erfolgen, bevor sich innerhalb der Firmen und Institutionen vorgefaßte Meinungen verfestigt haben, die dann nur zu reinen Meinungskämpfen führen.

Dann kann es als Modell für eine Zusammenarbeit von Instituten und Firmen angesehen werden, bei der für beide Teile wertvolle Erfahrungen gewonnen werden konnten. Vor allem gelang es beim PEARL-Projekt, die Gefahr zu vermeiden, daß eine der beiden Seiten als reiner Zulieferer für die andere Seite auftritt.

Die Institute (hier besonders das 3. Phys. Inst. der Univ. Erlangen) konnten die ihnen gemäß Rolle eines Katalysators erfüllen. Es darf wohl als sicher angenommen werden, daß das PEARL-Projekt ohne Institutsmitarbeit überhaupt nicht zustande gekommen wäre. Es wurde vermieden, daß Institute nur als Quelle billiger Arbeitskraft für die Industrie eingesetzt wurden, was meist zu größeren inneren Schwierigkeiten besonders bei Universitätsinstituten führt, da es deren Charakter als eigenständigen Forschungsstellen eindeutig widerspricht. Es wurde aber auch genügend Grundsatzentwicklung bei den beteiligten Firmen geleistet, um bei den dort tätigen Fachleuten den Eindruck zu vermeiden, nur für die Routinearbeit verschlissen zu werden. Zudem konnte vielwertvolles

know-how der Firmen für die Allgemeinheit nutzbar gemacht werden.

Zum dritten kann die PEARL-Arbeit als Modell für ähnliche Arbeiten an Universitätsinstituten gelten. Es soll hier entschieden der weithin verbreiteten Ansicht widersprochen werden, daß an Universitäten nur zweckfreie Forschungsarbeiten betrieben werden sollten. Im Sinne der Einheit von Forschung und Lehre an der Universität muß es Universitätspersonal möglich sein, know-how unter halbindustriellen Bedingungen zu gewinnen, das in eine praxisbezogene Lehrtätigkeit einfließen kann. Das soll nun in keiner Weise die Bedeutung der reinen Grundlagenforschung schmälern, jedoch muß immer wieder auf die Gefahr hingewiesen werden, die eine forschungsmäßige "Inzucht" an den Universitäten für die Qualität der technisch-wissenschaftlichen Ausbildung haben muß.

Als letztes kann die PEARL-Arbeit als ein Modell dafür angesehen werden, wie es gelingen kann, mit einem relativ geringen Anteil an Staatsgeldern ein Projekt mittlerer Größe auszusteuern. Das Vorhaben NDV13 bestand aus Stellen für 2 wissenschaftliche Mitarbeiter und einen technischen Mitarbeiter. Notwendige Sekretariatsarbeit wurde zunächst aus einem anderen Bundesprojekt mit getragen. Mit diesem sehr kleinen Stab wurde eine Entwicklungsgruppe zusammengehalten, die zusätzlich noch 11 fast hauptamtliche Mitarbeiter aus anderen Institutionen (vor allem Industrie) und bis zu 20 gelegentliche Teilnehmer hatte (siehe Protokoll im Anhang C).

4.5 Schlußbemerkung

Am Abschluß sollen noch ganz kurz einige wesentliche Gedanken stehen:

Zuerst sollten auch einige bei der PEARL-Arbeit gemachte Negativerfahrungen mit verwertet werden. In der Hauptsache erwies sich das Fehlen einer zentralen Aussteuerungsmöglichkeit der Gruppe über staatliche Zuschüsse als sehr hemmend. Es war bekannt, daß auch einige der teilnehmenden

Firmen staatliche Förderungsmittel bereits während der Definitionsphase von PEARL erhielten, jedoch war keine direkte Einflußnahme seitens der Hauptinitiatoren möglich. Man sollte in Zukunft versuchen, bei ähnlichen Projekten einen verantwortlichen Projektträger zu finden, der dann auch gewisse Vollmachten erhält. Das mag zunächst Schwierigkeiten machen, jedoch sollten sich Lösungsmöglichkeiten durch Diskussion unter den jeweiligen Beteiligten finden lassen. Es ist abzuwarten, wie die Entwicklung des PEARL-Subsets verläuft, bei der diese Erfahrung ja bereits durch eine gewisse zentrale Funktion des Projekts PDV berücksichtigt wird.

Insgesamt jedoch kann die PEARL-Arbeit als erfolgreich betrachtet und positiv gewertet werden. Insbesondere hat sich gezeigt, daß bei geeigneter Kräftezusammenfassung ("Klotzen, nicht Kleckern") europäische Fachleute durchaus in der Lage sind, etwas der großen amerikanischen Konkurrenz gleichwertiges zu schaffen und dies auch - gegeben eine entsprechende Unterstützung durch Staat und Industrie - international zu vertreten wissen. Doch das ist ein anderer Punkt, der noch zu sehr von der allgemeinen wirtschaftlichen Lage abhängt, um jetzt abschließend beurteilt zu werden. Jedenfalls war das PEARL-Projekt eine entscheidende Ermutigung für eine ganze Anzahl deutscher Teilnehmer am internationalen fachlichen Kräftenessen. Und dieser Erfolg, der etwas das sonst übliche Starren auf den "Großen Bruder" abbauen helfen kann, sollte nicht unterschätzt werden.

5. Bibliographie

Vorbemerkung:

Die nachstehende Bibliographie geht über die im Text angeführten Zitate (gekennzeichnet durch Referenznummern) beträchtlich hinaus, erhebt aber dennoch keinen Anspruch auf Vollständigkeit. Besonders im Abschnitt über PEARL konnten nur die z.Zt. am leichtesten zugänglichen Arbeiten aufgeführt werden, da es z.T. bis Redaktionsschluß nicht gelang, von allen PEARL-Autoren eine vollständige Liste ihrer einschlägigen Spezialarbeiten zu erhalten, z.T. der Vertraulichkeitsgrad und die Zugänglichkeit vieler Arbeiten noch geklärt werden muß.

Eine wesentlich erweiterte Bibliographie über Prozeßsprachen allgemein und speziell PEARL ist aber im Rahmen der Folgeprojekte in Arbeit und wird baldmöglichst veröffentlicht werden.

5.1. Algorithmische Sprachen

- [23] Naur et al.: Revised Report on the Algorithmic Language ALGOL 60;
Num. Math. 4 (1963) 420-453
- [22] USA Standard FORTRAN, United States of America Standards Institute
USAS X3.9 - 1966, New York, (Mar. 1966)
- Zemanek: Rolle und Bedeutung formeller Sprachen;
E u. M 83(1966), 463
- [24] IBM PL/1 - language-specifications;
IBM C28/65-71
- Beech et al.: Concrete Syntax of PL/1
IBM United-Kingdom-Lab. TN 3001
- [25] Wijngarden, Mailloux, Peck, Coster: Report on the Algorithmic Language ALGOL68;
Num. Math. 14 (1969) 79-218

5.2. Real-Time-Sprachen

5.2.1 allgemeine Überlegungen

- Opler: Requirements for Real-Time Languages;
Comm. ACM 9 (1966), 196
 - E.W. Dijkstra: Cooperating Sequential Processes;
ed. Genuys; Programming Languages (Academic Press,
London 1968)
 - IEEE-Transactions: Industrial & electronics & control
instrumentations 16 (1968) No. 2 (Sonderheft über
Prozeßkontrollsprachen)
 - P. Elzer: Facilities for a real-time language;
1st European Seminar on Realtime programming,
Harwell, 1971, ed. I.C. Pyle, I.N. Hooton and
R.I. MacDonald (Transcripta Books, London, 1972),
p. 22
- [8] D.R. Frost: Experiences with languages for real-time
programming; 2nd European Seminar on Realtime-
programming, Erlangen, 1972, ed. I.C. Pyle,
P. Elzer (Transcripta Books, London 1974), p.9-14
- [16] P. Elzer: Möglichkeiten zur Entwicklung einer Programmier-
sprache für kernphysikalische Experimente; Arbeits-
papier, Phys. Inst. Erlangen, Juli 1968
- [15] H. Mittendorf: Konzept einer Mittelsprache für Prozeß-
anwendungen; SIEMENS A.G., Karlsruhe, personal
communication

5.2.2. Beschreibungen spezieller Sprachen (außer PEAPL)

- [28] M.H. Perstein: The JOVIAL (J3) Grammar and Lexicon;
System Development Corp., Santa Monica, Calif.,
TM-555/002/04, (May, 1966)
- [14] V. Haase: EXOS - Entwurf einer experimentorientierten
Programmiersprache; GfK-Karlsruhe, externer
Bericht, Aug. 1967
- [5] 'RTL', A language for Real-time-systems;
The Computer Bulletin, (Dec. 1967) 202-212
- [27] D.R. Frost: Fortran for Process Control;
Instrumentation Technology 16 (1969) 4
- [7] INDAC-8 software;
Digital Equipment Corporation; Maynard, Mass.
- [17] J.D. Schoeffler: RTL - a Real Time Language for Computer
Control Software; Systems Research Center,
Case Western Reserve University, Cleveland, Ohio 44106
- [26] P.M. Woodward, P.R. Wetherall, B. Gorman: Official
definition of CORAL 06; HMSO, London, (1970)
- [11] CERCI-SESA: Industrial Programming Language: LAI;
Minutes of the Fourth Workshop on Standardization
of Industrial Computer Languages, Purdue University,
Lafayette, Indiana, Nov. 9-12 (1970)
- [10] J.G.P. Barnes: Some Proposals for RTL/2;
Minutes of the Fourth Workshop on Standardization of
Industrial Computer Languages, Purdue University,
Lafayette, Indiana, Nov. 9-12 (1970)
- [13] G. Koch: Die Software des Systems DP1000;
BBC-Nachrichten, Bd. 53 (1971), 278-284
- H. Zima: A programming language for real-time systems;
Scripta Books, Computing with REAL-TIME-SYSTEMS,
Vol.1, (1972)
- [12] M. Ritout et al.: PROCOL, A programming system adapted for
process control; 5th IFAC World Congress, Paris,
June 1972, ed. Intern. Federation of Automatic
Control (Instrument Society of America, New York)
- [9] I. Gray et al.: RTL/1, a language for real-time programming;
ICI, Pangbourne, personal communication
- [6] J. Gertler: High-level programming for process control;
The Computer Journal, Vol. 13, Nr. 1, (Feb. 1970)

5.2.3. Beschreibungen des Sprachentwurfes PEARL bzw. spezieller Eigenschaften

- [1] / J. Brandes, S. Eichtentopf, P. Elzer, V. Haase, H. Mittendorf:
PEARL, The Concept of a Process- and Experiment-
oriented Programming Language; elektronische
datenverarbeitung, 10 (1970) 429-442
- / P. Elzer, H. Mittendorf: PEARL - eine Prozeßsprache;
Anwenderkreis I Siemens-Prozeßrechner, Jahres-
tagung 1971, HMI Berlin; ed. P. Abend, R. Bublitz
(HMI-B120, Berlin, 1972)
- / P. Elzer: PEARL - eine prozeß- und experimentorientierte
Programmiersprache; Anwenderkreis I, Siemens-
Prozeßrechner, Jahrestagung 1972; FHS-Südwestfalen-
Paderborn; ed. J. Becker (Paderborn, 1973)
- / J. Brandes, B. Eichenauer, S. Eichtentopf, P. Elzer, *speaker*
L. Frevert, V. Haase, P. Holleczeck, K. Kreuter, B. Krüger,
G. Müller, K. Pfeiffer, P. Rieder, B. Schürlein,
K.-H. Timmesfeld:
A realtime programming language and its application
for measuring processes;
IFAC, 5th World Congress, Paris, June 1972 *(Instrument Society of America)*
ed. Intern. Federation of Automatic Control
(Instrument Society of America, New York)
- [21] B. Eichenauer, P. Elzer, V. Haase, P. Holleczeck, K. Kreuter,
G. Müller: Die Prozeß- und Experimentprogrammiersprache
PEARL; Entwicklung, Aufbau und Anwendungsbeispiel;
German chapter of the ACM, Jahreshauptversammlung,
Freiburg / Br., 3/1973
- [2] / K.-H. Timmesfeld, B. Schürlein, P. Rieder, K. Pfeiffer,
G. Müller, K. Kreuter, P. Holleczeck, V. Haase, L. Frevert,
P. Elzer, S. Eichtentopf, B. Eichenauer, J. Brandes:
PEARL - A Proposal for a process- and experiment
automation realtime language; Ges. f. Kernforschung
mbH, Karlsruhe, Report KFK-PDV 1, April 1973
- [3] B. Eichenauer, V. Haase, P. Holleczeck, K. Kreuter, G. Müller:
PEARL - eine prozeß- und experimentorientierte
Programmiersprache; Angewandte Informatik 9/73
- P. Rieder: Das Task- und Timing-Konzept von PEARL;
2. Jahrestagung der GI, Karlsruhe 10/72
ed. by Gesellschaft f. Informatik, (Karlsruhe 1972)
- K. Pelz: Die vollständige PEARL-Syntax in Baumform mit
Querverweisen; Gesellschaft f. Kernforschung mbH,
Karlsruhe, Report PDV-E 14 (Sept. 1973)
- B. Eichenauer: Die Prozeßprogrammiersprache PEARL-System-
beschreibung und E/A-Konzept; 2. Jahrestagung der
GI, Karlsruhe 10/72; ed. by Gesellschaft für Infor-
matik, Karlsruhe 1972)

5.3 Standardisierung

- [19] LTPL-Committee: Functional requirements for language features for a procedural language for industrial computers; Minutes of the 6th Workshop, Purdue University, Lafayette, Indiana, Okt. (1971) 77

- [20] LTPL-C Voting Rules: Minutes of the 7th Workshop, Purdue University, Lafayette, Indiana, Apr. (1972) 146

- [18] Report of the long term procedural language committee; Minutes of the 8th Workshop, Purdue University, Lafayette, Indiana, Okt. (1972) 105-106

- [4] P. Elzer, J. Reh: Kurze Darstellung der Situation der internationalen Standardisierung auf dem Gebiet der Realzeitprogrammiersprachen und der Rolle von PEARL; Ges. f. Kernforschung mbH, Karlsruhe, Projekt PDV, Sept. (1973)

5.4. Übersichten

- IFIP Fachwörterbuch der Informationsverarbeitung;
Amsterdam, 1968
- Berger, Seibt, Strunz: Bibliographie zum Thema
Programmiersprachen; elektronische datenverarbeitung
11 (1969) Heft 5 und 7
- Workshop on Standardization of Industrial Computer
Languages: Preliminary Glossary (Minutes, part 2)
Purdue University, Lafayette, Indiana, Oct. 1969
- Dictionary of Industrial Digital Computer Terminology
(prepared by: Glossary Committee of Purdue Workshop)
Instrument Society of America, Pittsburgh, 1972)