

## Die Bedeutung von PEARL in der Prozessinformatik

Dr. Dieter Pflügel, St. Georgen

Zusammenfassung. Um einem Studenten der Informatik die Probleme der Prozeßtechnik nahe-zubringen, benötigt man eine Prozeßbeschreibungs-sprache. Es zeigt sich, daß PEARL die hierfür geeignetste Sprache ist. Mit PEARL kann die Hardware beschrieben werden, als auch die Lösungsalgorithmen für die Logik der Programme und deren zeitliches Verhalten.

Summary. To teach the problems of process control to students of computer science, we need a process description language. It is shown, that PEARL fulfills all the requirements. PEARL can be used to describe the hardware, the algorithms of programming logic and the time behaviour.

### 1. Die Bedeutung von höheren Programmiersprachen für den Einsatz von Datenverarbeitungsanlagen

Die Ausbreitung der Datenverarbeitung ist unaufhaltsam. Es gibt kaum noch ein Gebiet, in denen man den Computer nicht in irgendeiner Form wiederfindet. Insbesondere durch die Entwicklung der Mikroprozessoren ist dieser Innovationsschub in der Datenverarbeitung fast unbegrenzt. Parallel zur Verbreiterung des Einsatzspektrums von DV-Anlagen kann man eine Weiterentwicklung bzw. Verbesserung der Programmiersprachen, die zur Realisierung von DV-Problemen herangezogen werden, beobachten. Man kann sogar sagen, daß die Weiterentwicklung der Programmiersprachen eine notwendige Voraussetzung für das Eindringen des Computers in immer weitere Anwendungsbereiche ist. Bis in die Mitte der siebziger Jahre hatten sich auf dem Groß-EDV-Sektor ganz bestimmte Sprachen etabliert, wie FORTRAN, PL/1, ALGOL, APL, COBOL, BASIC u.a.m. Die sogenannten maschinennahen Sprachen, auch Assemblersprachen genannt, wurden dabei immer mehr in den Hintergrund abgedrängt. Ein Hauptgrund war die starke Spezialisierung auf einen bestimmten Maschinentyp. Portabilitätsprobleme spielten eine immer größere Rolle. Jede der oben genannten problemorientierten Sprachen hatte sich ihren Markt erobert, wobei Überschneidungen der Einsatzgebiete natürlich möglich waren, wie z.B. bei FORTRAN, ALGOL, COBOL und PL/1. Mit dem verbesserten Werkzeug 'Programmiersprache' erweiterte sich damit auch

automatisch der Problemkreis, der mit diesen Sprachen rationell gelöst werden konnte. Es steht sicher außer Frage, daß ohne FORTRAN und COBOL die Datenverarbeitung ihren heutigen Stand sicher noch nicht erreicht hätte. Mit anderen Worten, je komfortabler eine Sprache ist, um so breiter wird das Einsatzgebiet der Datenverarbeitungsanlagen sein. Eine Grenze der Komplexität der Sprache wird allerdings oft durch die DVA-Systeme selbst diktiert, die z.B. nicht beliebig große Speicher für den Übersetzungsvorgang zur Verfügung stellen können. Ein Ausweg sucht man dann zwar in der sogenannten Cross-Software, die aber nicht immer befriedigende Ergebnisse liefert. Daß höhere Sprachen die Einsatzbereitschaft von DVA-Systemen erhöhen, konnte man ganz extrem beim Einsatz der Mikroprozessoren beobachten. Die anfänglich benutzten primitiven Sprachen wie Hexacodeverschlüsselung bis zum Assembler, grenzten die Einsatzgebiete der Mikros stark ein. Entsprechendes galt für das Programmierpotential. Nur Spezialisten konnte die Kleinstrechner programmieren. Heute gibt es kaum noch ein Entwicklungssystem, auf dem nicht in den höheren Sprachen wie BASIC, PASCAL, oder FORTRAN programmiert werden kann. Damit ist der Kreis der potentiellen Anwender ganz erheblich angestiegen.

Man kann zusammenfassend sagen: erst höhere Programmiersprachen erlauben eine volle Ausschöpfung der Möglichkeiten des Einsatzes von Datenverarbeitungsanlagen.

## 2. Auswahl höherer Prozeßsprachen

Bis jetzt wurde von der Datenverarbeitung im allgemeinen gesprochen. Da das Anwendungsspektrum sehr breit ist, ist es sinnvoll, wenn man eine Klassenbildung nach Aufgabengebieten durchführt. Man unterscheidet ganz grob, bekanntlich zwischen der technischwissenschaftlichen, der kommerziellen und der prozeßtechnischen Datenverarbeitung. Eine entsprechende Unterteilung ergibt sich dann auch für die problemorientierten höheren Sprachen. Die beiden erstgenannten Problemklassen werden z.B. durch die Sprachen FORTRAN, ALGOL, PL/1 b.z.w. COBOL, PL/1 versorgt. Für die Prozeßdatenverarbeitung ist die Auswahl an höheren Programmiersprachen bis jetzt noch recht mager. Das Programmieren in maschinennahen Sprachen scheidet aus Kostengründen aus. Das Softwareprodukt wird viel zu teuer und unsicher, eine Wartung ist fast unmöglich. Neben den verschiedenen Echtzeit-FORTRAN-Dialekten hat sich als neu konzipierte Sprache in letzter Zeit PEARL als echte Prozeßsprache hervorgetan. Das Sprachkonzept von PEARL beinhaltet im wesentlichen drei Komponenten. Einmal kann man mit Anweisungen, wie sie von anderen höheren Programmiersprachen bekannt sind, den logischen Ablauf einer Prozeßaufgabe formulieren. Der zeitliche Ablauf b.z.w. Synchronisationen von Teilalgorithmen erfolgt mit speziell entworfenen Sprachelementen. In dem sogenannten Systemteil eines PEARL-Programms kann im Unterschied zu allen anderen Programmiersprachen, die Struktur und die Eigenschaften der zu verwendenden Hardware beschrieben werden. Damit hat der Anwender die Möglichkeit, eine Prozeßaufgabe optimal an die vorhandene Peripherie anzupassen. Ein weiterer Gesichtspunkt von PEARL ist die Möglichkeit, Sicherheitsprobleme der Automationstechnik zu berücksichtigen. Durch die oben aufgezählten Eigenschaften des Sprachkonzeptes ist PEARL sehr gut geeignet, prozeßtechnische Aufgaben im Detail und problemlos zu beschreiben. Man kann also sagen: von den vorhandenen Prozeßsprachen ist PEARL die am weitesten entwickelte und am besten geeignete Sprache.

## 3. Die drei Anwendergruppen für die Prozeßtechnik

Die große Vielfalt der Einsatzmöglichkeiten von Datenverarbeitungsanlagen in der Prozeßtechnik bringt es mit sich, daß die Anforderungen an die Kenntnisse zur Lösungsfindung solcher Probleme sehr hoch sind b.z.w. das Wissensspektrum sehr breit sein muß. Hier benötigt man Kenntnisse über die Struktur der DVA's, einen fundierten Überblick über Programmiermethoden und Verfahren, als auch das notwendige Wissen über den Ablauf und das Verhalten des technischen Prozesses selbst.

Diese Tatsache findet einen starken Niederschlag in der Ausbildung von entsprechenden Ingenieuren und Programmierern. Hierbei sind im wesentlichen drei Zielgruppen zu unterscheiden. Die erste Gruppe bringt bereits sehr gute Programmierkenntnisse mit, besitzt aber nur geringe Kenntnisse über den technischen Teil der Prozeßtechnik, das sind die Hardwarekomponenten sowie die Physik des Prozesses. Gruppe zwei ist zu Gruppe eins genau diametral angesiedelt. Programmierkenntnisse sind praktisch kaum vorhanden. Zwischen beiden Gruppen liegt dann die Gruppe drei, die über mäßige Erfahrungen im Programmieren von DVA verfügt. Die erste Gruppe setzt sich im wesentlichen aus Informatikern verschiedener Richtungen zusammen. Hinzu kommen Programmierer aus der Industrie, die sich ihre Programmierkenntnisse an Projekten, z.B. aus der kommerziellen DV angeeignet haben. Zur Gruppe zwei gehört der große Anwenderkreis der Prozeßtechniker der verschiedensten Branchen. Das sind Physiker aus dem Bereich der Kernphysik, Maschinenbauer, die sich mit der Steuerung von Werkzeugmaschinen beschäftigen, das sind Chemiker der chemischen Industrie oder Verfahrenstechniker u.a.m.. Gruppe drei rekrutiert sich aus eins und zwei, wobei die jeweiligen Kenntnisse unterschiedlich tief sind. Sie stellt gewissermaßen den Übergang der ersten zur zweiten Gruppe dar.

4. Gruppe eins als Zielgruppe für die Ausbildung in der Prozeßinformatik

Für die Bearbeitung von prozeßtechnischen Aufgabenstellungen benötigt man das Potential aus den Gruppen eins und zwei. Für die Ausbildung zum Prozeßinformatiker ergeben sich deshalb zwei grundverschiedene Vorgehensweisen. Für Gruppe eins müssen Kenntnisse über die technische Seite eines Prozesses vermittelt werden, sowie die Prozeßhardwarekomponenten der DVA erläutert werden. Der Gruppe zwei müssen dagegen Programmiertechni-

## 4. Gruppe eins als Zielgruppe für die Ausbildung in der Prozeßinformatik

Für die Bearbeitung von prozeßtechnischen Aufgabenstellungen benötigt man das Potential aus den Gruppen eins und zwei. Für die Ausbildung zum Prozeßinformatiker ergeben sich deshalb zwei grundverschiedene Vorgehensweisen. Für Gruppe eins müssen Kenntnisse über die technische Seite eines Prozesses vermittelt werden, sowie die Prozeßhardwarekomponenten der DVA erläutert werden. Der Gruppe zwei müssen dagegen Programmiertechni-

ken beigebracht werden. Dazu gehört auch das Programmieren in geeigneten Programmiersprachen. Die weiteren Ausführungen werden sich jetzt mit Gruppe eins beschäftigen und hier dann wieder speziell die Verhältnisse an der Fachhochschule Furtwangen erläutern.

#### 5. Prozeßinformatik an der FH Furtwangen

Wie wir gesehen haben, ist die Prozeßtechnik nur ein Teilgebiet aus dem heute sehr breit angelegten Arbeitsfeld der Datenverarbeitung. Mit Einführung der Studienrichtung Informatik an den Universitäten und Technischen Hochschulen Mitte der sechziger Jahre wurde zwar der großen Nachfrage nach Fachkräften für die Lösung von Aufgaben aus der Datenverarbeitung Rechnung getragen, es zeigte sich aber sehr schnell, daß der theoretische Teil der Ausbildung den praktischen Teil bei weitem übertraf. Das hatte zur Folge, daß ein Informatiker dieser Prägung doch erhebliche Schwierigkeiten bei der Bewältigung programmiertechnischer Aufgaben hatte.

Die Fachhochschule Furtwangen war seinerzeit die erste Fachhochschule, die den Engpaß der praxisbezogenen Ausbildung im Bereich der Informatik erkannte. Daraus entstand 1971 der Fachbereich Allgemeine Informatik (AI), mit den Schwerpunkten: mathematische Grundlagenfächer wie Algebra(10) ( die in Klammern angegebenen Zahlen geben die Stundenzahl für Vorlesung b.z.w. Praktikum an ), Analysis(12), Wahrscheinlichkeitsrechnung und Statistik(6), numerische Methoden(4+2), Operations Research (4) und den DV-spezifischen Fächern wie Programmiersprachen(12+6), Übersetzerbau(4+2), Betriebssysteme(4+2), Datenorganisation(4+2) mit den jeweiligen Praktikas. Die Hardwarekenntnisse werden mit folgenden Fächern vermittelt: Elektrotechnik(2), Digitaltechnik(2), Elektronikpraktikum(4) und Struktur von DV-Anlagen. Ab Wintersemester 1980/81 wurde die Vorlesung Prozeßinformatik als zweistündige Vorlesung mit einem zweistündigen Praktikum in die Pflichtvorlesungen im 7 Semester mit aufgenommen. Bis dahin wurde die Prozeßtechnik als Wahlvorlesung geführt. Mit der Übernahme der Prozeßinformatik in den Pflichtvorlesungskatalog bestand jetzt allerdings die Notwendigkeit, die Vorlesung vollständig neu zu gestalten und umzustellen. Die Zielgruppe hatte sich weitgehendst in Richtung Gruppe eins verschoben, also sehr gute Ausbildung in den

Datenverarbeitungsfächern, nur mangelnde Kenntnisse bezüglich der Hardware und der technischen Seite der Prozeßtechnik.

#### 6. PEARL als didaktische Stütze für die Prozeßinformatik

Wie eingangs schon gesagt wurde, überdeckt die Prozeßtechnik ein sehr breites Spektrum von Anwendungsgebieten. Bei einer zweistündigen Vorlesung bleibt hier nur ein kleiner Spielraum für die Vermittlung von Kenntnissen zur Lösung prozeßtechnischer Probleme übrig. Es war deshalb notwendig, einen didaktischen Weg zu finden, mit dem man die oben geschilderten Schwierigkeiten überbrücken konnte. Es muß versucht werden, sozusagen in der Sprache des Informatikers zu sprechen. Am einfachsten erfolgt dies mittels geeigneter Symbolsprachen, mit denen man das Prozeßgeschehen dann beschreiben kann. Im ersten Ansatz wurde deshalb als Prozeßbeschreibungssprache die in Freiburg entwickelte Sprache BASEX, das ist BASIC für Echtzeitanwendung, versuchsweise genommen. Da die Sprache auf BASIC beruht, war das Erlernen der Sprachelemente, mit denen man den logischen Ablauf eines Algorithmus beschreibt, innerhalb sehr kurzer Zeit möglich. Problematisch wurde es allerdings beim Aufstellen b.z.w. der Realisierung von zeitlichen Abläufen. BASEX gestattet es z.B. nicht, zyklische Zeitaufträge abzusetzen. Das Semaphorkonzept zur Synchronisation von Teilalgorithmen ist nicht realisiert worden. Wegen der BASIC-Eigenschaft der Sprache waren die so entstandenen Programme unstrukturiert und für einen Außenstehenden kaum nachvollziehbar. Als Beschreibungssprache in der Vorlesung kam somit BASEX nicht in Frage. Es wurden deshalb verschiedene andere Sprachen auf ihre Brauchbarkeit hin untersucht. Als Ergebnis dieser Untersuchung zeigte sich, daß PEARL den aufgestellten Forderungen am besten entsprach. Die Forderungen lauteten:

1. Blockorientierte Sprache
2. Moderne Sprachelemente für die algorithmische Beschreibung des logischen Ablaufes eines Programms
3. Sprachelemente für die Beschreibung der zeitlichen Steuerung von Programmteilen
4. Sprachelemente für die Beschreibung des Hardwareaufbaus des Rechensystems und der Peripherie

5. Unterstützung der Idee der strukturierten Programmierung
6. Eignung als Systembeschreibungssprache im Unterricht
7. Möglichkeiten der Zerteilung des Gesamtsystems in getrennt übersetzbare Programstücke

Die einzelnen Forderungen lassen sich aus den folgenden Bedingungen ableiten.

Schon im ersten Semester werden die Studenten mit der höheren Programmiersprache PASCAL vertraut gemacht. PASCAL wird dann später im Übersetzerbau als Systemsprache verwendet. Damit ist das Erlernen einer weiteren Sprache, die verwandte Sprachelemente wie PASCAL enthält, problemlos. Dies betrifft Punkt 1 und 2 des Anforderungskatalogs. Punkt 3 ist besonders wichtig, da jetzt auf Problemebene das zeitliche Verhalten eines Prozesses formuliert werden kann. Querverbindungen z.B. zu Betriebssystemen können aufgezeigt werden. Punkt 4 folgt aus der Tatsache, daß die Hardwarekenntnisse nicht sehr detailliert sind. Damit ist es möglich, das Verständnis für diese Seite der Prozeßtechnik zu erhöhen. Punkt 5 ermöglicht einen 'sauberen' Programmierstil, dem nicht zu unterschätzende Aufgaben zufallen. Mit 6 ist es möglich benötigte Algorithmen im Unterricht allgemein zu formulieren, die dann später im Praktikum weiter verwendet werden können. Punkt 7 ist besonders wichtig, da zur Durchführung des Prozeßpraktikum von den Studenten verlangt wird, daß sie unter anderem Modifizierungen bestehender Aufgabenlösungen vornehmen sollen, b.z.w. einzelne Module neu schreiben müssen. Mit der Wahl von PEARL als Systembeschreibungssprache im Unterricht und als Programmiersprache im Praktikum kann man folgende Schwerpunkt formulieren, in denen PEARL ein besonders gutes didaktisches Hilfsmittel darstellt.

1. Beschreibung des Hardwareaufbaus eines Prozeßsystems. Dazu gehört einmal die Festlegung der Verbindungen der einzelnen Hardwarekomponenten untereinander sowie die Bestimmung der Informationsflüsse in Richtung und Größe. Zum anderen gehört dazu die Auslegung der Hardware wie z.B. Bitbreite eines Analogdigitalwandlers, Festlegung der benötigten Rechengenauigkeiten, Angleichung der Daten von der externen Darstellung an die interne Darstellung im

Rechner u.s.w. Die Vergabe von symbolischen Namen für die zu verarbeitenden Daten wie Interrupts, Analogdaten, Einzelbit- und Mehrbitgrößen u.a. macht sich hier besonders positiv bemerkbar.

2. Zerlegung der gesamten Prozeßaufgabe in Teilaufgaben, die teilweise parallel ablaufen können. Hier macht sich das Taskkonzept von PEARL besonders bezahlt. Der Student wird damit gezwungen, den Prozeß genau zu analysieren um z.B. Unabhängigkeiten zu erkennen. Mit der Möglichkeit der Prioritätenvergabe kann dann die Wichtigkeit dieser Teilaufgaben bei Gleichzeitigkeit diskutiert werden.
3. Aufstellung der Lösungsalgorithmen in allgemeingültiger Form. Die in PEARL bekannten Programmstrukturen wie if-then-else, for-while-Schleife und case-Verteiler sowie das Prozedurkonzept genügt im allgemeinen zur problemorientierten Darstellung der Aufgabenlösungen. Entsprechendes gilt für die vorhandenen Datenstrukturen.
4. Aufstellung von Zeitplänen für die Aktivierung der einzelnen Teilalgorithmen und Formulierung mit Hilfe der PEARL-Task-Steuerelemente.
5. Durchführung des Praktikums. Das Prozeßpraktikum läuft zeitlich parallel zur Vorlesung ab. Diese Konstruktion bringt im allgemeinen immer Schwierigkeiten mit sich, da die benötigten Programmiersprachen noch nicht bekannt sind. Es zeigt sich nun, daß mit PEARL dieses Problem weitgehendst behoben werden kann. Die wichtigsten Sprachelemente werden zu Beginn der Vorlesung kurz vorgestellt und dann im Laufe des Semesters je nach Bedarf erweitert.

#### 7. Verwendetes Prozeßrechensystem

Mit der Umstellung von BASEX auf PEARL wird auch ein anderes Rechensystem verwendet. Zur Umstellung durchgeführt, die vor allem für das Praktikum notwendig ist. Als neues System wird eine PDP 11/34 mit folgendem Ausbau verwendet: 256 KB Arbeitsspeicher, eine Magnetbandstation mit 800-1600 bpi, zwei RK07 - Platten mit je 25 MB, sieben Terminals, ein ADC (12 Bit) mit vorgeschaltetem Multiplexer für 16 Eingänge, 4 DAC, eine Digitale-I/O

mit 16 Bit, interruptfähig, ein Drucker ELA 180. Das PEARL-System läuft unter dem Betriebssystem RSX11M.

#### 8. Schlußbemerkung

Für die Unterrichtung des Faches Prozeßinformatik ist es unbedingt notwendig, eine Prozeßbeschreibungssprache zu benutzen, die die algorithmische Beschreibung von Prozeßabläufen in der dem Informatiker gewohnten Weise erlaubt. Sie muß es weiterhin ermöglichen, Hardwarekonfigurationen im Detail zu beschreiben. Diese Sprache sollte möglichst

auch als Programmiersprache für ein Praktikum verwendbar sein. Die Verwendung von PEARL erscheint zur Zeit als die einzig praktikable Lösung. Es ist zu hoffen, daß PEARL in Zukunft auch auf kleineren Systemen implementiert wird.

Dr.Pflügel, Dieter  
7742 St.Georgen  
Am Sommerrain 4  
Tel. 07724 7275