

PEARL im Vergleich mit anderen Programmiersprachen

Teil A: Rückblick auf die Entwicklung von PEARL

Von P. Elzer

Rückblick auf die Entwicklung von PEARL

Vorbemerkungen

Programmiersprachen entstehen nicht im luftleeren Raum. Sie haben ihre Vorgänger und Vorbilder. Sie werden von den Anforderungen ihres Einsatzgebietes und vom allgemeinen Stand der Technik zur Zeit ihrer Entstehung geprägt. Sie schreiben gewissermaßen den jeweils gültigen Wissensstand bezüglich Konstruktionselementen von Programmen fest. Sind diese noch nicht genügend formalisierbar oder reichen traditionelle Beschreibungsformen nicht aus, so muß tiefergehende Grundlagenarbeit investiert werden. Sie sind nicht zuletzt aber auch sozialen, ökonomischen und politischen Einflüssen unterworfen. So sind gerade geltende Lehrmeinungen, die Kräfteverhältnisse des Marktes oder nationale Interessen stets auch vermengt gewesen in Entstehung oder Durchsetzung vieler Programmiersprachen.

Eine umfassende Betrachtung von Programmiersprachen müßte daher auch alle diese Faktoren mit in Betracht ziehen. Daß dies im Rahmen der Zeitbegrenzung der folgenden Vorträge nicht möglich ist, liegt auf der Hand. Es sollte jedoch nicht versäumt werden, solche Gesichtspunkte zumindest einmal ins Bewußtsein rufen.

Die Programmiersprachszene zu Beginn der Entwicklung von PEARL

Die Entwicklung von PEARL begann 1969 im Rahmen der "Studiengruppe Nuklearelektronik", gefördert vom BMBW. Da äußere Einflüsse zu Beginn einer Entwicklung am prägendsten wirken, sind die zu dieser Zeit wesentlichen Programmiersprachen von besonderem Interesse.

- FORTRAN (= formula translator), eine Entwicklung der IBM in den 50-er Jahren; praktisch die erste der heutigen höheren Programmiersprachen; zum damaligen Zeitpunkt erste Ansätze in Richtung "Real-time-FORTRAN".

- COBOL (= Common Business Language), eine vom US-Verteidigungsministerium finanzierte Entwicklung zur Erstellung kaufmännischer und verwaltungstechnischer Programme; ihr Schwerpunkt liegt auf nichtnumerischer Datenverarbeitung und Datenhaltung.

- ALGOL 60 (= Algorithmic Language 1960), eine 1960 veröffentlichte Sprache, die von einer Arbeitsgruppe der IFIP erarbeitet wurde; versucht, die bei FORTRAN erkannten Schwächen bezüglich Datentypen und Programmstruktur zu beseitigen; die zu seiner Beschreibung entwickelte Notation (Backus-Naur-Form (BNF)) erweist sich als bahnbrechend für die Entwicklung praktisch aller folgenden Programmiersprachen.

Diese drei Sprachen waren praktisch das 'tägliche Handwerkzeug' der Programmierung zu jener Zeit - und sind es eigentlich noch heute. Dazu kamen einige andere Sprachen, die aber jeweils "gemeindebildend" wirkten und trotz hohen theoretischen Wertes wenig Eingang in die tägliche Praxis fanden:

- APL (= A Programming Language), eine speziell für die extrem kompakte, mathematisch orientierte Darstellung von Formeln entwickelte Sprache, die außerdem voll auf interaktive Arbeitsweise ausgerichtet ist; wegen des großen notwendigen Zeichenvorrats und der "unüblichen" Notation damals noch kaum verbreitet.
- LISP (= List Programming Language), eine Sprache, in der sich besonders Textstrukturen gut darstellen lassen.

Alle diese Sprachen - und noch einige andere mehr - erwiesen sich als nicht ausreichend zur Erfüllung der Anforderungen der Realzeitprogrammierung. Also wurden zusätzliche Sprachen in die Betrachtung einbezogen, die zumindest vom Ansatz her den Anspruch erhoben, für dieses Anwendungsgebiet geeignet zu sein:

- CORAL 66, eine Entwicklung des "Royal Radar and Signal Establishment" in England zur Programmierung von Computersystemen des Verteidigungsbereiches; bewußter Verzicht auf ein unterliegendes Betriebssystem, daher keine Zeitsteuerungs- und Parallelismuselemente, keine E/A; dafür direkte Anschlüsse an Maschineneigenschaften und Assembler.
- PL/I (= Programming Language One), eine Entwicklung der IBM; gedacht als Universalsprache für ALLE Anwendungen; Obermenge von COBOL, FORTRAN, ALGOL; ergänzt um Dateiverwaltung, Ankopplung an Interrupts, Fehlerreaktionen, rudimentäre Tasksteuerung.
- ALGOL 68, wiederum eine Entwicklung der IFIP; ebenfalls Anspruch einer Universalsprache; enthält erstmals wirklich strenge Typbindung, totales Referenzkonzept und Elemente zur Spracherweiterung; "orthogonaler Entwurf"; zur Beschreibung wird die neue Technik der zweistufigen Grammatiken entwickelt (Van Wijngarden); außerordentlich schulbildend.
- PAS 1 (= Prozeßautomatisierungssprache 1), eine Entwicklung von Brown Boverie & Cie.; PL/I-Dialekt mit Ergänzungen zur Beschreibung der Hardwareumgebung eines Programmes und seines kompletten Zeitverhaltens.

Die Anforderungen an eine Programmiersprache für Realzeitanwendungen

Es war zu Anfang nicht unbedingt das Ziel des PEARL-Kreises, eine komplette neue Programmiersprache zu entwickeln. Es wurde geprüft, ob existierende Programmiersprachen dahingehend ergänzt werden konnten, um folgende Forderungen zu erfüllen:

- Komplette Beschreibung des Zeitverhaltens nebenläufiger (paralleler) Programme
- Synchronisierung
- Ein-/Ausgabe von binären, numerischen und graphischen Größen
- Komplette Behandlung von Dateien ohne Verlassen des laufenden Programmes
- Getrennte Übersetzung von Programmoduln

- Behandlung von Bit- und Zeichenketten im Programm
- Schutz von Konstanten
- Arbeiten mit Zeigern
- Beschreibung der Hardwareumgebung des Programmes

Vorschläge zur Lösung dieser Probleme wurden 1970 in einem ersten "Konzept" veröffentlicht.

Die Detailentwicklung

Diese erste Veröffentlichung erweckte starkes Interesse und bewirkte eine schnelle Vergrößerung des PEARL-Kreises, in dem dann praktisch alle einschlägigen Hersteller und Softwarehäuser der Bundesrepublik integriert waren. Anfang 1972 wurde die Förderung der PEARL-Entwicklung im "Projekt PDV" des BMFT übergeleitet. Ende 1972 lag dann eine erste komplette Sprachbeschreibung vor, die nach mühevoller Redaktionsarbeit 1973 als PDV-Bericht "KfK-PDV-1" veröffentlicht wurde.

Bei der Detailarbeit stellte sich auch heraus, daß eine Gemeinsamkeit aller Beteiligten und die Erfüllung der oben skizzierten technischen Anforderungen nur auf der Basis einer Neuentwicklung zu erreichen waren.

Die Normung von PEARL

Bei den 1975 anlaufenden Arbeiten zur Umsetzung der ersten PEARL-Beschreibung in eine normfähige Form stellte sich dann heraus, daß noch erheblicher Aufwand zur Erzielung einer entsprechenden formalen Qualität der beschreibenden Dokumente investiert werden mußte. So mußten z.B. neue Beschreibungshilfsmittel für die Darstellung der Nebenläufigkeit auf der Basis der Petrinetze entwickelt werden. Ein geschlossenes Ein-/Ausgabemodell war für die Zusammenfassung von konventioneller und Prozeßein-/ausgabe nötig. Erfahrungen mit Implementationen von Subsets von ALGOL 68 beeinflussten das Referenzkonzept, etc. Auch internationale Kooperationen blieben nicht ohne Einfluß.

So spiegelte die fertige Beschreibung von "Full-PEARL" im PDV-Bericht "KfK-PDV-130" 1977 durchaus die rasche technische Entwicklung auf diesem Fachgebiet wieder.

Die derzeitige Programmiersprachenszene

Der Bedarf nach einer höheren Programmiersprache für Realzeitanwendungen wurde etwa gleichzeitig in vielen Ländern festgestellt. Das führte dazu, daß an vielen Stellen unabhängig voneinander Entwicklungen gestartet wurden, die heute konkurrierend nebeneinanderstehen. Dazu gehören auch die, die im folgenden näher betrachtet werden:

- PASCAL, als "Protest" gegen den Wildwuchs der "Universalsprachen" PL/I und ALGOL 68 entwickelt, die kaum mehr lernbar waren und deren Compiler finanziell untragbar wurden; ursprünglich von Wirth in Zürich zu Lehrzwecken eingesetzt, um den Studenten Verständnis für die Strukturierungsprinzipien von Programmen beizubringen; legt auch großen Wert auf Strukturierung von Daten.
 - RTL/2 (= Real-time Language 2), ursprünglich eine Entwicklung des englischen Chemiekonzerns ICI, wird heute durch ein Softwarehaus vertrieben; baut auf Erkenntnissen von ALGOL 68 auf, legt aber großen Wert auf Einfachheit und Programmstrukturierung;
- da sie keine Realzeitelemente etwa im Sinne von PEARL enthält, kommt sie ohne unterliegendes Betriebssystem aus.
- LTR (= Language Temps Réel), eine Entwicklung der französischen Marine; heute in Frankreich Standardsprache für Anwendungen im Verteidigungsbereich, Grundphilosophie ähnlich bei PEARL, aber mit abweichender Strategie beim unterliegenden Betriebssystem.
 - RTF (= Real-time-FORTRAN), eine ständig wachsende Sammlung von Prozeduren mit standardisierten Aufrufen zur Programmierung von Realzeitanwendungen in FORTRAN.
 - Real-time-BASIC, Ergänzung der wegen ihres interaktiven Charakters vor allem für Labor- und Kleinanwendungen sehr bewährten Sprache BASIC durch (teilweise) genormte Prozeduraufrufe für Realzeitsteuerung.
 - Ada (dies ist ein Name!), neueste Entwicklung des US-Verteidigungsministeriums zur Programmierung von "embedded computer systems"; steht in der Tradition von PASCAL und Systemimplementierungssprachen wie LIS, RTL/2, etc. und Experimentalsprachen wie CLUE, ALPHARD, etc.