

Erfahrungen mit PEARL-Schulungen im industriellen Bereich

Dr. Hans Windauer, Lüneburg

Zusammenfassung

Dieser Kurzbericht faßt die Erfahrungen aus acht PEARL-Seminaren und -Schulungskursen im industriellen Bereich zusammen.

Schlüsselwörter: PEARL Schulungskurs

Summary

This short report summarizes experiences gained in eight PEARL seminars with participants from industrial institutions.

Key words: PEARL seminar

1. Grundlage der Erfahrungen

Dieser kurze Bericht fußt auf fünf dreitägigen, einem zweitägigen und zwei eintägigen Schulungskursen, die der Autor in den Jahren 1976 bis 1979 durchgeführt hat.

Die Teilnehmer kamen aus den Bereichen

- Chemische Industrie
- Förder- und Lagertechnik
- Hersteller von DV-Anlagen
- Kernforschung
- Luft- und Raumfahrt
- Maschinenbau
- Militär
- Rundfunktechnik
- Stahlindustrie (Hüttenwesen)

Die meisten kannten Assembler oder Echtzeit-FORTRAN; weniger Teilnehmer kannten ALGOL 60 oder PASCAL. Kenntnisse in Prozeßprogrammierung waren immer vorausgesetzt.

PEARL war damals noch nicht so verbreitet und gefestigt wie heute, d.h. die Teilnehmer waren in der Regel zwar neugierig, aber auch kritisch bis skeptisch. Deshalb erforderten die Kurse immer auch viel Motivationsarbeit für PEARL.

2. Designziele von PEARL

Die Haltung vieler Teilnehmer war durch folgende Feststellungen gekennzeichnet:

"Nun programmieren wir schon zu 80% in FORTRAN anstatt in Assembler. Die restlichen 20% lohnen die Einführung von PEARL nicht."

Denn:

"Die PEARL-Anweisungen für E/A und Tasking sind doch nur syntaktischer Zucker. Beim Übergang von einem Rechnersystem zu einem anderen können die wenigen neuen Betriebssystem-Aufrufe oder RT-FORTRAN-Calls für E/A und Tasking schnell gelernt werden."

Offenbar wurden zu oft die Tasking- und E/A-Eigenschaften von PEARL, nicht jedoch seine Vorzüge im algorithmischen Bereich betont. Außerdem war der eigentliche Vorteil der Integration von E/A und Tasking, die Definition einer Schnittstelle zu Echtzeitbetriebssystemen in seinen Auswirkungen zu oft nicht bewußt.

Deshalb sollten zu Beginn einer PEARL-Schulung alle Designziele und alle wesentlichen Merkmale von PEARL überzeugend, d.h. anhand praxis-naher Beispiele erläutert werden:

Designziele von PEARL [1]:

- Flexibilität der Programme
- Selbstdokumentation der Programme
- Einfache Handhabbarkeit der Sprache
- Einfaches Erlernen der Sprache.

Die wesentlichen Merkmale von PEARL [1]:

- Umfangreiche Menge von Task-Anweisungen und Mechanismen zur Behandlung von Zeitereignissen und Einplanungen
- Synchronisation mittels Sema- und Boltvariablen
- Systemteil
- Übliche Operationen für die Behandlung von Files
- Strukturen, Bit- und Zeichenketten
- Referenzen für indirekte Adressierung
- Benutzerdefinierte Datentypen (TYPE) und benutzerdefinierte Operatoren
- Getrennte Compilierung von Moduln
- Definition eines virtuellen Echtzeit-Betriebssystems.

3. Tragfähigkeit von PEARL

Wichtig ist, darauf hinzuweisen, welche Institutionen hinter PEARL stehen, damit die Kursteilnehmer die Bedeutung und Tragfähigkeit von PEARL erkennen.

Erwähnt werden sollten zumindest die beteiligten Hersteller, der VDI/VDE, der BMFT mit seinem Fördervolumen, die Großforschungseinrichtungen sowie nun natürlich die Normung durch den DIN.

Heute kann auch schon von einigen realisierten Anwendungsprojekten berichtet werden, was sicherlich am besten überzeugt.

Bei allen durchgeführten Kursen kamen die Teilnehmer auf die Konkurrenten von PEARL zu sprechen: IRTF (Industrial Real Time FORTRAN) oder FORTRAN 75 (des VDI/VDE), CORAL 66, RTL, (Concurrent-) PASCAL und neuerdings natürlich Ada. Der Kursleiter sollte PEARL und diese Sprachen einordnen können und grob über ihren Entwicklungsstand und ihre Normungssituation Auskunft geben können.

4. Berücksichtigung verschiedener Zielgruppen

Bei den durchgeführten Kursen gab es im wesentlichen drei Gruppen: Teilnehmer, die bisher nur mit Assembler programmiert hatten, Teilnehmer, die bisher nur mit FORTRAN programmiert hatten und Teilnehmer, die sowohl Assembler als auch FORTRAN kannten.

Allen drei Gruppen neu waren die Hilfen, die PEARL für ein strukturiertes Programmieren bietet. Diese Sprachelemente mußten dementsprechend betont werden (siehe auch Punkt 5).

Die Assembler-Gruppe war immer an Effizienzfragen interessiert. Typische Fragen waren: "Was macht der Compiler daraus?", "Wie ist das Zusammenspiel zwischen Sprache, Compiler, Laufzeitsystem und dem Betriebssystem mit seinen E/A-Werken und Treibern?"

Die FORTRAN-Gruppe hatte die üblichen Schwierigkeiten bezüglich des Arbeitens mit Referenzen und hätte gerne hierauf mehr Zeit und Beispiele verwendet gesehen.

Falls also die Planung eines Kurses es erlaubt, sollten der Aufbau des Kurses, die Betonung von Schwerpunkten und die Auswahl der Beispiele auf die Vorbildung der Teilnehmer abgestimmt werden.

5. Betonung des algorithmischen Teils

Die Vorzüge von PASCAL, insbesondere seine Möglichkeiten zum Arbeiten mit problemorientierten Datentypen, werden überall gepriesen und von vielen akzeptiert, auch wenn sie sie gar nicht selbst kennengelernt haben. PEARL wird als Prozeßsprache eingeordnet mit besonderen Sprach-

mitteln für Tasking und E/A. Daß PEARL in seinem algorithmischen Teil mit Ausnahme von enumerations, variant records und global generators alle Sprachelemente von PASCAL enthält, mit Bit- und Zeichenketten u.a. sogar darüber hinausgeht, ist kaum bekannt. Umso wichtiger ist die Schulung dieser Eigenschaften von PEARL, d.h. der STRUCT- und TYPE-Elemente, zumal sie sehr viel zur Flexibilität und Selbstdokumentation der Programme beitragen, also auch ihre spätere Wartbarkeit und Modifizierbarkeit sehr stark beeinflussen.

Wichtig ist vor allem auch die Darstellung des Modulkonzepts von PEARL, das den Einsatz von PEARL (im Gegensatz zu PASCAL) bei größeren Projekten überhaupt erst erlaubt.

Bei den durchgeführten 3-tägigen Kursen erwies sich folgende Aufteilung als günstig:

- 1,5 Tage : Algorithmik
- 0,5 Tag : Ein/Ausgabe
- 0,5 Tag : Tasking
- 0,5 Tag : komplexes Anwendungsbeispiel.

6. Bedeutung der Programmiermethodik

Ein erfahrener (und eingefleischter) FORTRAN-Programmierer nahm einmal einen PEARL-Sprachreport zur Hand, der nur aus Syntax- und Semantikregeln bestand. Der FORTRAN-Programmierer wollte PEARL kennenlernen und dazu ein bestehendes FORTRAN-Programm aus dem Hochregallager-Bereich in PEARL neu schreiben. Eine Anleitung lehnte er ab. Heraus kam das gleiche Programm mit PEARL-Schlüsselwörtern statt FORTRAN-Schlüsselwörtern. Modul- und Blockstruktur, eigene Datentypen oder Strukturen waren nicht benutzt worden, obwohl das Programm dadurch kürzer und transparenter geworden wäre.

Deshalb sollte ein PEARL-Schulungskurs für Assembler- oder FORTRAN-Programmierer mit einer Anleitung in Programmiermethodik verbunden sein, bei der zumindest die Grundzüge der strukturierten Programmierung und ihre Berücksichtigung in PEARL-Programmen behandelt werden. Ein dreitägiger Kurs ist dazu natürlich zu kurz.

7. Anwendungsbeispiele

Im Bereich der Prozeßautomatisierung sind überschaubar kleine, aber trotzdem aussagekräftige Beispiele schwierig zu finden. Trotzdem sollte der Aufwand nicht gescheut werden, Beispiele zu erarbeiten, die dem Arbeitsgebiet der Kursteilnehmer entstammen. Anhand solcher Beispiele können die Eigenschaften von PEARL am eindringlichsten und auch am vorteilhaftesten demonstriert werden.

Sehr günstig wirkt sich aus, wenn das PEARL-Programm zu einem von Teilnehmern gewünschten Beispiel "live" während des Kurses entwickelt werden kann. Hierfür sollten jedoch mindestens 4 Stunden zur Verfügung stehen.

8. Literatur

[1] KFK-PDV 1 : PEARL. A proposal for a process- and experiment automation realtime language. Gesellschaft für Kernforschung mbH, Karlsruhe. April 1973.

Anschrift des Autors:

Windauer, Hans
Entwicklungsbüro Wulf Werum
Glogauer Straße 2a
2120 Lüneburg
Tel.: 04131 - 53344, - 53066