

Aufbau und Durchführung einer Anwenderschulung Systematisches Programmieren mit PEARL Teil 1

Dipl.-Inform. Herbert Brinkkötter, Dr.-Ing. Klaus Rebensburg, Berlin (West)

Zusammenfassung:

In diesem Beitrag wird über den Aufbau einer an der TU-Berlin entwickelten PEARL - Schulung, die dazugehörigen Schulungsunterlagen, sowie über die bei der Durchführung gesammelten Erfahrungen berichtet. Konzeption dieser einwöchigen Schulung, Lernziele, praktischer Übungsteil und die Dokumentation werden vorgestellt.

Verschiedene Stufen der Entwicklung und Erprobung führten zur Heraushebung einer als unbedingt notwendig erachteten Menge von Lernzielen und Inhalten bei vorgegebener Zielgruppe und vorgegebenen zeitlichen Rahmenbedingungen.

Summary:

This paper describes the contents of a PEARL - Course, documentation and experiences. Concept of this One-Week-Course, objectives, practical part and documentation is presented.

Different phases of development and practice led to the emphasis of absolutely necessary objectives and contents in the context of the given target group and time frame.

1. Historie

Am 1. 7. 1978 begann ein bei der Prozeßrechnerverbundzentrale (PRZ) der TU - Berlin durchgeführtes und vom BMFT gefördertes Forschungsprojekt. Im Rahmen dieses Vorhabens wird die Programmiersprache PEARL (Process and Experiment Automation Realtime Language) auf verschiedenen Rechnern implementiert und bei mehreren Instituten eingeführt.

Durch die Einführung dieser höheren Programmiersprache versprechen sich die Anwender eine Effizienzsteigerung und eine Kostenreduzierung bei der Lösung von wissenschaftlichen Problemen in unterschiedlichen Fachgebieten. Für die TU - Anwender bedeutet das u.a. eine Konzentration der Kräfte auf die inhaltlichen wissenschaftlichen und technischen Problemstellungen.

Auf diesem Hintergrund entstand schon 1979 für die Schulung der Anwender im PRZ ein

Skript "Systematisches Programmieren mit PEARL", das für 14-tägige Schulungen verwandt wurde. Der sehr heterogene Anwenderkreis in diesen Kursen - außer Informatik- und Elektrotechnik - Studenten waren die in Industrieanwendungen und Forschung tätigen Institute zu berücksichtigen - führte zu einer Umorientierung des Schulungskonzepts. Dieses wurde u.a. extern Anfang 1981 bei einer einwöchigen Kompaktschulung beim Institut für Rundfunktechnik in München erprobt. Der PEARL - Verein erwarb Rechte an der dort verwendeten Schulungsunterlage und beauftragte die TU - Berlin, daraus eine für die Schulungen des PEARL - Vereins verbindliche Lehr- und Lernkonzeption zu erstellen.

Diese wurde, inclusive der dazugehörigen Unterlagen termingerecht fertiggestellt und ist seit Juli 1981 Grundlage der vom PEARL - Verein in Zusammenarbeit mit dem VDI - Bildungswerk veranstalteten PEARL-Kurse.

Konzeption und Inhalte wurden von den Herren H. Brinkkötter, K. Nagel, H. Nebel, K. Rebensburg von der TU - Berlin unter Mitwirkung der Herren K. Pelz, Uni Erlangen, D. Pflügel, FHS Furtwangen und T. Röhrich, Uni Stuttgart, erarbeitet.

In den folgenden Ausführungen berichten wir über die mit externen (nichtuniversitären) Anwendern gemachten Erfahrungen.

2. Zielgruppen

Die Konzeption zielte ursprünglich auf die Umschulung von FORTRAN - Kennern. Zu den Kursen meldeten sich jedoch gleichermassen:

- . Anfänger der Programmierung und Fortgeschrittene
- . Assembler und FORTRAN - Experten
- . Programmierer, Ingenieure und Abteilungsleiter

Aufgrund dieser Erfahrungen berücksichtigt die Schulung des PEARL - Vereins nun diese Gruppen.

3. Aufbau der Schulung "Systematisches Programmieren mit PEARL" Teil 1:

Dauer: 5 Tage Kompaktkurs (Teil 1)
40 Unterrichtseinheiten (U-Einheiten) (30 Zeitstunden)

Form: Definierte Mischung aus:

Vortrag: 20 U-Einheiten (15.5 Zeitstunden)

Tutorium und praktische Übungen am Prozeßrechner:

15 U-Einheiten (11 Zeitstunden)

Diskussion: 5 U-Einheiten (3.5 Zeitstunden)

Durchführung: 4 (mind. 3) Dozenten, die, ständig anwesend, Vorträge, Tutorium und praktische Übungen am Rechner durchführen.

Teilnehmer: gesamt max. 18, in Gruppen je 2 bzw 3 am Rechner

Ausstattung: 7 Terminals an einer Rechenanlage mittlerer Größe zur PEARL - Programmentwicklung.

Ort: Auf der Anlage eines Anwenders, hier am IRT München, an der TU - Berlin, am IVD Stuttgart.

Kursplan :

1. Tag 10:00 Begrüßung: Vorstellung der Teilnehmer und Dozenten, Klärung der Frage: Welche Erwartungen haben Teilnehmer und Dozenten? Welche Probleme hinsichtlich Automatisierung und Softwareentwicklung (Spezifikation, Entwurf, Implementierung) haben die Teilnehmer?

10:30 **Vortrag:** 'Vom Problem zum Algorithmus' 1.

13:00 Mittagspause

14:30 **Vortrag:** 'Vom Problem zum Algorithmus' 2.

16:00 **Praktikum/Tutorium:** Beispiel 1 der Schulungsunterlage wird von den Teilnehmern am Terminal eingegeben, getestet und modifiziert. Teilnehmer lernen den Dialog mit dem Rechner und machen erste praktische PEARL - Erfahrungen.

18:00 -

2. Tag 9:30 Vortrag: 'Problemgerechte Konstruktion von Algorithmen und Daten' 1.

11:00 Pause

11:30 **Praktikum/Tutorium:** Fortsetzung der Aufgabe vom 1. Tag, Diskussion mit den einzelnen Gruppen am Terminal.

Neue Aufgabe: Multiplikation zweier natürlicher Zahlen; als Operationen stehen nur == und + zur Verfügung

13:00 Mittagspause

14:30 **Vortrag:** 'Problemgerechte Konstruktion von Algorithmen und Daten' 2.

17:30 -

3. Tag 9:30 **Diskussion:** Vertiefung PEARL, Zusammenfassung der bisher erreichten Lernziele. "Automatisierungsprobleme der Zielgruppe".

Ist PEARL eine Sprache der Planner? Einbettung von Assembler, warum, wie?

PEARL, FORTRAN, ASSEMBLER: Vorzüge, Nachteile,

PEARL -) ADA, Verbreitung von PEARL.

11:30 Pause

12:00 **Tutorium:** Wie liest man die PEARL - Grammatik?

13:00 Mittagspause

14:30 **Tutorium/Praktikum:** Beispiel 2 der Schulungsunterlage, Teilnehmer bekommen die Programme als Datei zur Verfügung und programmieren das Layout der Ausgabe selbst.

17:30 -

4. Tag 9:30 **Vortrag:** 'Nebenläufige Algorithmen' 1.

11:00 Pause

11:30 **Vortrag:** 'Nebenläufige Algorithmen' 2.

13:00 Mittagspause

14:30 **Praktikum/Tutorium:** Aufgabe 2 (Nachzügler), Programmierung eines kleinen Beispiels (Autorenrennen) mit nebenläufigen Tasks, ohne Engpässe (Evtl Prozeß-I/O).

Diskussion der Problemstellung mit allen Gruppen, Einzeldiskussion in kleinen Gruppen während der Arbeit am Terminal.

18:00 -

5. Tag 9:30 **Vortrag:** 'Koordinierung nebenläufiger Algorithmen' 1.

11:00 Pause

11:30 **Vortrag:** 'Koordinierung nebenläufiger Algorithmen' 2.

Lösung eines Problems mit Nebenläufigkeit und Synchronisation (Semaphoren). Hinweise, wie man die Aufgabe "Autorenrennen" mit Engpaß (Semaphoren) lösen kann.

13:00 Mittagspause

14:00 **Vortrag, Diskussion:** Vertiefung und Abschlußdiskussion. Wurden die Erwartungen der Teilnehmer erfüllt? Perspektiven der PEARL - Anwendung, Tagungen, Literatur.

15:00 Ende des Kurses

4. Lernziele

Lernziele bzgl. Programmiermethoden und -techniken und Lernziele der Sprache PEARL:

Kapitel 1: Programmiermethoden und -techniken:

- . Entwurf von Algorithmen
- . Schrittweise Verfeinerung mit unterschiedlichen Formulierungsebenen
- . Unterstützung der Programmdokumentation

Kapitel 1: Lernziele PEARL

- | | |
|---|--|
| . Variablen, Konstanten | INV, INITIAL |
| . Vereinbarungen | DECLARE, SPECIFY |
| . Einfache Datentypen | FIXED, CHARACTER, BIT, FLOAT |
| . Wertzuweisung | ::: |
| . Einfache Operatoren | + - * / .. |
| . Verzweigungen | IF .. THEN .. ELSE .. FIN;
CASE .. ALT .. OUT .. END; |
| . Bedingte Wiederholung von Teilalgorithmen | WHILE .. REPEAT .. END; |
| . Ein/Ausgabebefehle | GET .. FROM .. , PUT .. TO .. |
| . Formatangaben | A, F, SKIP |
| . Vereinbarungen von Daterstationen | SPECIFY .. DATION .. ALPHIC; |
| . Genereller Aufbau eines PEARL-Programms | MODULE; |
| Modulkonzept | SYSTEM; .. ;
PROBLEM; ...
TASK; .. END; ... |

Kapitel 2: Programmiermethoden und -techniken

- . Verwaltung komplexer Daten und großer Datenmengen
- . Strukturierung durch Einführung benannter Teilalgorithmen

Kapitel 2: Lernziele PEARL

- . Verwendung von Dateien OPEN...BY IDF...OLD,CLOSE, CREATED, DIM, DATION
- . Typ-Vereinbarung TYPE, STRUCT
- . Vereinbarung von Operatoren OPERATOR
- . Kombination von Bedingungen für Wiederholungen von Teilalgorithmen FOR...WHILE...REPEAT...END
- . Benannte Teilalgorithmen PROCEDURE, RETURN, RETURNS
- . Aufruf von Prozeduren CALL
- . Parameterübergabe bei Funktionen und Prozeduren IDENTICAL
- . Umwandlung von Buchstaben zu Zahlenwerten TOFIXED, CHARACTER

Kapitel 3: Programmiermethoden und -techniken

- . Nebenläufige Algorithmen
- . Zeitliche Steuerung von Teilvorgängen
- . Schrittweise Verfeinerung (Vertiefung)
- . Konstruktion von Daten und ihren Zugriffsalgorithmen
- . Prozeß - Schnittstellen
- . Buchstabengrafik

Kapitel 3: Lernziele PEARL

- . Nebenläufige Prozesse TASK
- . Task-Steueranweisungen ACTIVATE, TERMINATE, RESUME, CONTINUE, PREVENT, SUSPEND
- . Bedingungen bei Task-Einplanungsanweisungen WHEN, AT, AFTER, EVERY, ALL, DURING, UNTIL
- . Datentypen CLOCK, DURATION
- . Zeitkonstanten HRS, MIN, SEC
- . Prozeß - Ein-/Ausgabe TAKE .. FROM...; SEND .. TO ..;

Kapitel 4: Programmiermethoden und -techniken

- . Konstruktion komplexer Systemteile
- . Summierung durch endliche Iteration
- . Bestimmung von Minimum/Maximum einer Reihe

Kapitel 4: Lernziele PEARL

- . Vertiefung der Prozeß - Ein-/Ausgabe BASIC; TAKE .. FROM ...; SEND .. TO ..
- . Binäre Ein-/Ausgabe ALL; READ .. FROM ...; WRITE .. TO ..
- . Vereinbarung von SEMA - Variablen SEMA .. PRESET
- . Synchronisation nebenläufiger Algorithmen REQUEST ..; RELEASE ..; Operatoren SQRT, **, CAT, TOCHAR

5. Dokumentation zur Schulung

Für die einwöchige Kompaktschulung erhält der Teilnehmer folgende Materialien:

1 Stundenplan,

1 Handbuch "Systematisches Programmieren mit PEARL" ca. 250 Seiten, das die vollständige Ausarbeitung des Kurses sowie u.a. im Anhang eine für diesen Kurs erarbeitete syntaktische Beschreibung der Sprache enthält,

1 PEARL - Kurzbeschreibung, ca 5 Seiten, bestehend aus einer Kurzversion der PEARL-Syntax, PEARL-Standardoperatoren, und PEARL-Schlüsselwörtern (PEARL-Reference-Card),

1 Kurzbeschreibung der Rechnerbedienung für den Kurs, ca 4 Seiten, enthält die Beschreibung der notwendigsten Kommandos zum Editieren und zur Bedienung des PEARL- Compilersystems. Diese Kurzbeschreibung liegt bisher für die Rechner HP1000, HP3000, PDP 11 vor.

Die einschlägige PEARL - Literatur zur Ansicht während des Kurses.

Zusätzliche Materialien des Dozenten:

- . Zerlegung in nebenläufige Teilalgorithmen
 - . Ermittlung der Schnittstellen und geeignete Koordinierung von Teilalgorithmen
 - . Konstruktion von gemeinsam benutzten Daten
- 1 Grundlagenpapier zum Kurs (ca 70 Seiten) [1]
- 1 kommentierter Stundenplan
- 1 Satz Overhead - Folien (ca 200 Stück),

6. Gedanken zur Grundkonzeption und praktischen Durchführung

Die grundlegenden Prinzipien der Didaktik, Methodik und Leitlinien des Kurses sind in [1] ausführlich dargelegt. Der Anhang von [1] enthält eine exemplarische Anwendung der Prinzipien und Leitlinien.

Methodik und Leitlinien sind anwendbar auf beliebige Zielgruppen und sind zunächst nicht an einen bestimmten zeitlichen Rahmen fixiert.

Für die konkrete Gestaltung dieser PEARL - Kurse wurden daher zusätzlich folgende Gesichtspunkte berücksichtigt:

- . Die Forderungen der Teilnehmer, in kurzer Zeit möglichst viel zu lernen (Anreise, Nebenkosten), führen zur Entscheidung der Durchführung als Kompaktkurs.

- . Die Vorkenntnisse und Voraussetzungen der Teilnehmer erfordern eine sorgfältige Dosierung von Informationsmenge und Schwierigkeitsgrad.

- . Die Trockenheit des Stoffes erfordert zusätzliche Motivationshilfen, die in Form von praktischen Erfolgserlebnissen am Rechner, Diskussion und individueller Betreuung in kleinen Gruppen den Kurs begleiten.

- . Die Heraushebung der algorithmischen Schreibweise bei der Konstruktion erlaubt die relativ leichte Implementierung in anderen Programmiersprachen (PASCAL, ADA, FORTRAN). Es wird, insbesondere am Anfang des Kurses, auf leichte Überschaubarkeit des gesamten Lösungswegs geachtet.

- . Nach der konkreten Programmierung in PEARL folgt stets die Verallgemeinerung, bezogen auf die Sprache.

- . Der vorgegebene zeitliche Rahmen führt zur Konzentration auf die "wichtigsten" Lernziele. Es müssen Schwerpunkte gesetzt werden, es wird bewußt auf Vollständigkeit verzichtet. Wichtige, in [1] genannte Bausteine des Konzepts werden herausgehoben, einige können nicht vertieft werden, einige werden reduziert - sie bleiben einem PEARL-Kurs Teil 2 vorbehalten:

Beispiele für nicht vertiefte Lernziele (Grund s.o.):

- . Formale Spezifikation, es wurde eine einfachere verbale Zwischenlösung gewählt,
- . Korrektheitsüberprüfungen (Plausibilitätskontrolle, Beweise),
- . Sicherheit der Algorithmen,
- . Große Programme, Programmsysteme.
- . Optimierung,
- . Erstellen einer Bedienungsanleitung für Programme

Beispiele für Einschränkungen:

- . Vielfalt der algorithmischen Problemstellungen,
- . dargebotener Sprachumfang, z.B. Vielfalt des DATION-Konzepts

Beispiele für Heraushebungen, Vertiefungen:

- . Der Prozeß der Programmentwicklung,
- . die genaue Beschreibung des Problems,
- . Methode der schrittweisen Verfeinerung,
- . Dokumentierung des Lösungswegs auf den verschiedenen Formulierungsebenen,
- . Verklemmungsfreiheit
- . Strikte Trennung von Konstruktion und Implementierung
- . Grundlegende Bausteine (Sprachelemente) zum Problemlösen.

Strikte Vermeidung des folgenden Themas:

- . "exotische" Feinheiten der Sprache PEARL.

7. Erfahrungen

Lob:

- . Unsere Erfahrungen zeigen, daß mit einer Einheit aus Kurs, Tutorium und integrierten praktischen Übungen am Prozeßrechner der Lehrstoff am besten vermittelt wird. Eigene Artikulierung und Versuche motivieren die Teilnehmer zusätzlich, bauen die Schwellenängste zum eigenen Tun ab. Für eine praxisorientierte Ausbildung sind praktische Übungen am Prozeßrechner we-

- sentlicher Bestandteil. Dieses bestätigten sämtliche Teilnehmer.
- . Aufbau, zeitliche Einteilung und Schwierigkeitsgrad wurden allgemein positiv bewertet.
 - . Die Schulungsunterlage wurde sehr positiv bewertet.
 - . Die intensive Betreuung durch 4 Dozenten fand breite Zustimmung
 - . Die Diskussion am 3. Tag hat die Teilnehmer stark motiviert.
 - . Die induktive Vorgehensweise vom Beispiel zur Verallgemeinerung wurde gut akzeptiert.
 - . Die Teilnehmer hatten keine nennenswerten Schwierigkeiten mit der Rechnerbedienung.
 - . Jedes behandelte PEARL-Programm ist so abgeschlossen, daß es auf der Rechenanlage lauffähig ist.
- . Modellprozesse wären eine sinnvolle Ergänzung der Übungen.
- . Der PEARL - Übungsbetrieb kann durch schnellere PEARL Compiler - Systeme effektiver werden. Die HP3000 - Anlage war mit 7 PEARL - Teilnehmern (zu) gut ausgelastet.
 - . Kursunterlagen sollten vor dem Kurs verschickt werden.
 - . Die Begeisterung der Teilnehmer schränkte die Freizeit der Dozenten etwas ein; nur mit sanfter Gewalt ließen sich die Herren abends und in den Mittagspausen von den Terminals lösen.
- Perspektiven:
- Kurse der oben beschriebenen Art werden 1982 zu folgenden Terminen vom PEARL-Verein in Zusammenarbeit mit dem VDI - Bildungswerk angeboten werden:
- 15. - 19. 3. 1982 (7. Woche) Stuttgart (IVD)
 - 21. - 25. 6. 1982 (25. Woche) Berlin (TU)
 - 6. - 10. 9. 1982 (36. Woche) Stuttgart (IVD)
 - 8. - 12.11. 1982 (45. Woche) München (IRT)

Kritik:

- . Das Lesen einer PEARL - Grammatik bleibt trotz großer Bemühungen der Kurs-Entwickler ein Problem. Die im Kurs verwendete Darstellung ist zwar relativ leicht auch von Anfängern lesbar, aber dennoch ein umfangreiches Werk.
- . Die PEARL - Kurzdarstellung ist verbesserungswürdig (PEARL- Reference Card). Die Verwendung von Syntaxdiagrammen wird geprüft.

8. Literaturhinweise

- [1] N a g e l, Klaus, N e b e l, Herbert:
Didaktische und Methodische Prinzipien und Leitlinien für die Seminarreihe: Systematisches Programmieren mit PEARL. Beitrag zur Tagung "PEARL in Lehre, Ausbildung und Schulung. Erlangen Okt 1981