

Erfahrungen mit Basic PEARL im Projekt Netzleitstelle Deggendorf¹⁾

Von D. Struhalla

Die PEARL-Anwendung hat gezeigt, daß die Sprache eine schnelle, fehlerarme und selbstdokumentierende Programmierung ermöglicht.

PEARL-Programme haben, wie alle in höheren Programmiersprachen geschriebenen Programme, einen gegenüber Assembler größeren Speicherplatzbedarf, zeichnen sich jedoch durch eine hohe Zuverlässigkeit aus.

Das Projekt

PEARL 300 für die Siemens-Systeme 300 wurde erstmals im Projekt Netzleitstelle Deggendorf der Energieversorgung Ostbayern AG (OBAG) eingesetzt. Die OBAG ist ein regionales Elektrizitätsversorgungsunternehmen. Auf einer Fläche von 22000 qkm (Bild 1) versorgt sie ein Drittel von Bayern mit elektrischer Energie. Bezogen auf die Fläche ist die abgegebene elektrische Arbeit (1119 MW Jahreshöchstleistung) des Unternehmens relativ gering, da wenig Ballungsräume, dafür aber sehr flächige Gebiete versorgt werden müssen. Zur sicheren Führung des ausgedehnten 110- und 20-KV-Netzes und zur raschen Behebung von eventuellen Störungen sollen für das Gesamtgebiet der OBAG 4 Netzleitstellen gebaut werden. Zur Koordinierung der Arbeiten in den 4 Netzleitstellen wird über den Netzleitstellen eine Netzleitung errichtet.

Die Netzleitstelle Deggendorf ist für das Verteilnetz in einem Gebiet von rd. 6000 qkm zuständig. In diesem Gebiet werden rd. vierzig Schalthäuser im 20-KV-Netz und dreißig 110/20-KV-Umspannwerke fernüberwacht, ferngemessen und ferngesteuert. Das Informationsvolumen aus diesen Schalthäusern und Umspannwerken beläuft sich im Endausbau auf

- 20.000 Meldungen
- 1.500 Meßwerte
- 1.500 Befehle

Unter den technologischen Begriffen Meldung, Meßwert und Befehl werden hier binäre Signale zwischen 1-8 Bit Länge verstanden.

¹⁾ Die diesem Bericht zugrunde liegenden Arbeiten wurden mit Mitteln des Bundesministeriums für Forschung und Technologie (Förderungszeichen: P 4.2/36; RE-OBA/1) gefördert. Die Verantwortung für den Inhalt liegt jedoch allein beim Autor.



Bild 1. Versorgungsgebiet
(22000 qkm, 2,2 Mio Einwohner)

Um diesen enormen Anfall von Informationen zu bewältigen, wird ein Prozeßrechner eingesetzt. Zur Verfügbarkeitserhöhung der Netzleitstelle ist die Rechneranlage als Doppelrechnersystem (Bild 2) ausgelegt. Vom Prozeßrechnersystem werden die Aufgaben Steuern, Melden, Überwachen, Protokollieren und Bedienen wahrgenommen. Im Einzelnen sind dies die Aufgaben

- Signalisierung von spontanen Änderungen im Netz
- Ausführung von Einzelbefehlen und Befehlsfolgen
- Überwachung der Ausführung von ausgegebenen Befehlen
- Überwachung von Meßwerten auf die Einhaltung von Grenzwerten
- Überwachung parallel geschalteter Transformatoren
- fortlaufende Protokollierung aller Betriebsvorfälle
- Archivierung von wichtigen Informationen zur späteren Auswertung.

Alle Informationen werden auf Farbsichtgeräten dargestellt und signalisiert. Die Steuerung (Befehle) erfolgt über eine

Funktionstastatur, wobei die einzelnen Bedienschritte im angewählten Schaltbild auf dem Farbsichtgerät mitgeführt werden.

Die Beschreibung des Netzes erfolgt im Dialog mit dem Prozeßrechner. Hierzu steht ein bildschirmgeführtes Dialogsystem für die Datenpflege und für die Bildkonstruktion zur Verfügung. Losgelöst von internen Softwarestrukturen werden Netzänderungen - z.B. Erweitern oder Neuerstellen von Schalt- und Umspannanlagen - auf technologischer Ebene prozeßparallel beschrieben.

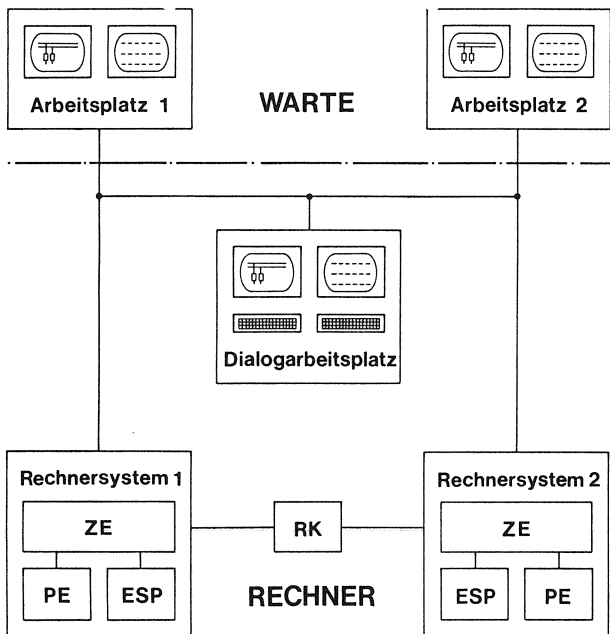


Bild 2. Anlagenkonfiguration

Das Anwendersoftwaresystem ist überwiegend in PEARL 300 programmiert worden. Nur für die besonders zeitkritische Fernwirkbearbeitung (Telegrammfolgefrequenz 16 ms) ist auf den Assembler zurückgegriffen worden. Eine quantitative Gewichtung der Codelänge und der Programmanzahl ergibt einen Anteil von 95% PEARL 300 zu 5% Assembler. Insgesamt wurden rd. 60 PEARL-Tasks mit einer Gesamtlänge von 1000 k Byte Länge implementiert.

Erfahrungen mit Basic PEARL

Die Sprache PEARL ist leicht erlernbar. Dies ist sicherlich mit auf die klare, übersichtliche und selbstdokumentierende Sprachnotation zurückzuführen. Zum wirklichen Arbeiten mit dem Werkzeug PEARL wird allerdings auch das Verständnis für das zugrunde liegende Sprachkonzept benötigt. Als Beispiel seien hier die strenge Typbindung und die Blockstruktur genannt. Der Sprachumfang von Basic PEARL hat sich bis auf einige Ausnahmen als ausreichend erwiesen. Die Festlegung des Gültigkeitsbereiches eines Selektornamens in PEARL, der sich nicht auf die zugehörige Struktur sondern

auf den übergeordneten Block bezieht, sehen wir als Nachteil an. Dadurch geht ein wesentlicher Teil der in der Strukturdefinition liegenden Transparenz verloren.

Die Formulierung von Datennahstellen zwischen einzelnen Programmen - Datei und Common Data - ist zu elementar. Der versierte Anwender vermißt die direkte Datenübergabe zwischen Programmen.

Die Koordinierung des Softwaresystem, Netzleitstelle Deggendorf, geschieht fast ausschließlich über Semaphorvariablen. Damit PEARL-Systeme noch effektiver arbeiten, müßte die Semaphorvariable dem ACTIVATE-Statement gleichgestellt werden, d.h. die Semaphorvariable muß um die Zeitmodifikation erweitert werden.

Implementierung und Test haben gezeigt, daß die Fehlerquote bei PEARL-Programmen sehr niedrig ist. Hier erweist sich die bereits zur Compilierzeit erfolgende Datentypprüfung (Typbindung) als sehr nützlich.

Die Sprachelemente von PEARL liegen fast ausnahmslos auf einer höheren, logischen Ebene. Deshalb eignet sich die Sprache sehr gut als Bindeglied zwischen dem Systemmann und dem Technologen. Sie unterstützt guten Programmierstil, Modularität und strukturierte Programmierung. Programmiersprachen sind weder die Ursache noch die Lösung von Softwareproblemen. Wegen ihrer zentralen Rolle bei allen Softwareentwicklungen wirken sie allerdings richtungsweisend. PEARL bewirkt bei der Lösung von Echtzeitproblemen eine Kosteneinsparung durch

- reduzierte Codier-, Test- und Fehlerbehebungskosten
- geringere Projektführung und Wartungskosten aufgrund erhöhter Lesbarkeit und Durchschaubarkeit der Programme.

Die Kosteneinsparung beläuft sich gegenüber einer Assemblerprogrammierung auf 20 bis 40%.

Erfahrungen mit PEARL 300

PEARL 300 - Module müssen nach der Compilierung mit dem Siemens PEARL-Compiler PC30 mit Hilfe des Binders zu einem ladbaren und danach ablauffähigem Programm gebunden werden.

Die Zeit zum Compilieren und Binden liegt bezogen auf die mittlere Programmlänge im Projekt Netzleitstelle Deggendorf bei ca. 5 Min. In Verbindung mit einem leistungsfähigen Drucker ergeben sich somit gute Compilier- und Bindezeiten.

Der Compiler PC 30 liefert ein gut dokumentiertes Fehlerprotokoll, das eine schnelle und sichere Fehleridentifikation ermöglicht.

In PEARL 300 ist ein 'online Test' zur Laufzeit möglich. Es sind die Funktionen Zeilentrace, Haltepunkte und Haltepunkte mit Bedingung realisiert. Das Anschauen und Verändern von Variablen ist in Verbindung mit der standardmäßigen Systemtesthilfe möglich.

| Funktion | Basic-PEARL | Full-PEARL | PEARL 300 |
|---|-------------|------------|-----------|
| ● Charakterselektion .CHAR(i) | o | + | + |
| ● Gruppenselektion .CHAR(i-k) | o | + | + |
| ● Gruppenselektion .BIT(i-k) | o | + | + |
| ● Initialisierung von Feldern | o | + | + |
| ● Standard Prozeduren für Datum und Uhrzeit | o | o | + |

+ = vorhanden

o = nicht vorhanden

Bild 3. Sprachumfang

PEARL 300 arbeitet mit einem variablen blocktiefenabhängigen Keller. Über diesen Keller ist PEARL 300 auf Prozedurebene mit beliebigen Sprachen mischbar.

Im Rahmen von Full PEARL ist PEARL 300 um einige Sprachelemente wie z.B. Charakterselektion, Gruppenselektion von Charactern und Bits, Initialisierung von Feldern und Prozeduren für Datum und Uhrzeit erweitert (Bild 3).

Die mittlere Programmlänge im Projekt Netzleitstelle Deggendorf liegt bei 18 k Byte, wobei das Spektrum von 4 k Byte bis 44 k Byte reicht. Eine subjektive Beurteilung der Länge von PEARL-Programmen und funktionsähnlichen Assemblerprogrammen ergibt einen Verlängerungsfaktor um 2. Dieser Faktor darf nicht verallgemeinert werden. Er gilt für das Projekt Netzleitstelle Deggendorf in Verbindung mit einem einmal im Zentralspeicher reentrant bereitgestelltem Laufzeitsystem.