# SASA – Schießausbildungs-, Sicherheits- und Auswerte-Anlage

von Dr.-Ing. **Horst Weber**, Frankfurt am Main

Zusammenfassung

Die Schießausbildungs-, Sicherheits- und Auswerteanlage ist ein größeres Automatisierungssystem, basierend auf mehreren gekoppelten Prozeßrechnern.
Zur Entwicklung der Anwendersoftware wird im Battelle-Institut e. V., Frankfurt, Basis-PEARL als Programmiersprache eingesetzt.
Es werden die Anforderungen an die Software und an ihre Entwicklung beschrieben. Die Probleme beim ausschließlichen Einsatz von PEARL und ihre Lösung durch besondere Programmiermethoden und kleinere Assemblerprogramme werden an Beispielen erläutert.

Schlüsselwörter: gekoppeltes Mehrrechnersystem, Schießplatzautomatisierung, universelles Dialogsystem in PEARL

Summary

SASA, a computerized training and security control for an air defense artillery range, is an extensive automation system based on a hierarchical realtime computer network. Battelle-Institut e. V., Frankfurt am Main, is developing the application software in PEARL.
The requirements to the software and to the software development are described. Problems which arise by exclusive usage of PEARL and its solutions by means of special programming methods and small assembler programs are described and examples are given.

Keywords: computer network, automation system for artillery range, universal dialog system with PEARL

## 1 Übersicht über Anforderungen, Hardware- und Software-Komponenten der Anlage

### 1.1 Anforderungen

Die SASA-Anlage soll den heute größtenteils manuell gesteuerten und überwachten Schießbetrieb auf dem Flugabwehrschießplatz der Bundeswehr in Todendorf an der Ostseeküste automatisieren.

Dies ist notwendig - zum einen, weil die Anforderungen an verfügbare Schießzeit, Variierbarkeit der Übungen und Geschwindigkeit der Flugziele ständig erhöht werden, und zum anderen, weil durch die Eigenschaften moderner Waffensysteme, wie große Reichweite, Richt- und Feuergeschwindigkeit, die Sicherheitsüberwachung mit dem herkömmlichen Methoden nicht mehr gewährleistet werden kann. Weiterhin möchte man die Auswertung und Anzeige der Schießergebnisse verbessern und beschleunigen, damit die leitenden Offiziere unmittelbar die Leistungen der Soldaten beurteilen können.

Die Geschütze werden zum Schießen in parallel zur Küste angelegten Schießbahnen in Stellung gebracht und feuern in Richtung See. Da die Reichweite meist über die Dreimeilenzone hinausgeht, kann jeweils nur in solche Richtungen geschossen werden, für die eine Gefährdung von eventuell vorbeifahrenden Schiffen unter Verwendung genügend großer Sicherheitssektoren ausgeschlossen werden kann ( B i l d 1 ).

Zur Zeit erfolgt die Überwachung der Richtung visuell durch einen Soldaten, der gegebenenfalls das Feuer manuell oder elektrisch unterbrechen kann.

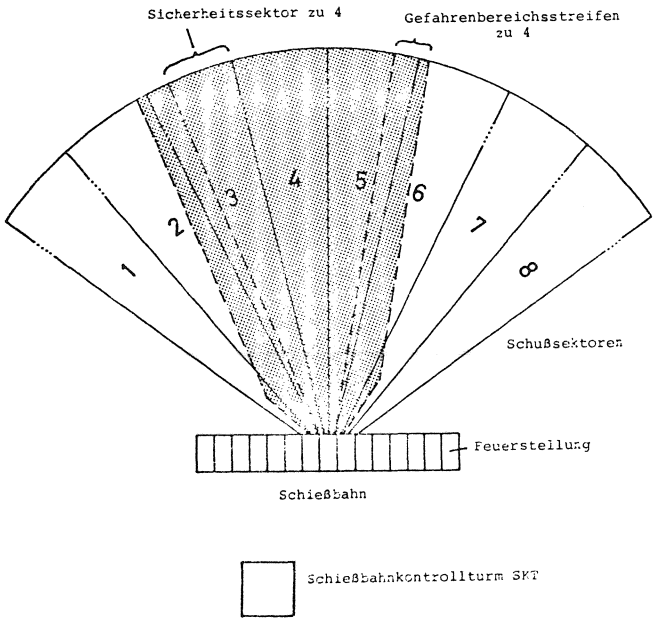


Bild 1. Sektoren einer Schießbahn und Sektorgrenzen für den Sektor 4

Zur Realisierung der Nachrichtenketten während des Schießens sind hintereinander bis zu fünf Soldaten notwendig, um zum Beispiel eine Feuerstop-Nachricht bis hin zu einem Geschütz zu übertragen. In Zukunft wird jedoch weniger Personal zur Verfügung stehen, während die Anforderungen wie oben geschildert zunehmen.

Mit Hilfe der SASA-Anlage soll der Betrieb weitestgehend automatisch ablaufen. Im wesentlichen sind dazu die folgenden Eigenschaften zu realisieren:

- Für den Aspekt Sicherheit wird eine Radarüberwachung des Schießplatzes, des Seegebietes und des Luftraumes durchgeführt.
- Die Ergebnisse der Überwachung werden automatisch schießbahnspezifisch in Informationen über freie und gesperrte Sektoren umgesetzt und den betreffenden Geschützen gemeldet.
- Die Sektorüberwachung bei den Geschützen läuft automatisch ab.

Bei Gefahr für die innere oder äußere Sicherheit kann das Schießen zentral unterbrochen werden.

Die Schleppflugzeuge werden automatisch in ihrem Kurs überwacht und so geführt, daß sich die Schleppziele auf reproduzierbarem Übungskurs an der Feuerstellung vorbeibewegen.

Der Zeitpunkt des Geschoßaustritts und des -durchgangs am Ziel wird automatisch ermittelt und ergänzt durch Angaben über das verursachende Geschütz der Auswerteanlage zur Verfügung gestellt. Auswertedaten sollen unmittelbar nach ihrer Entstehung angezeigt und später in Listen aufbereitet ausgegeben werden.

An zentraler Stelle soll eine Überwachung der gesamten Anlage mit Hilfe visualisierter Daten ermöglicht werden, die dem verantwortlichen Sicherheitsoffizier Steuerung und Beurteilung des Gesamtschießbetriebes erlauben.

Zentrale Anforderung an die SASA-Anlage ist die Gewährleistung der Sicherheit und Zuverlässigkeit in allen Betriebssituationen. Zum Erreichen dieser Eigenschaften wurde vom Auftraggeber als Programmiersprache PEARL verlangt und für die Entwicklung besondere Maßnahmen bei Entwurf und Implementierung der Software vorgeschrieben. Projektbegleitend ist die Dokumentation der entwickelten Software nach vorgegebenen Richtlinien zu erstellen.

## 1.2 Hardware-Komponenten

Die Hardware-Komponenten der SASA-Anlage sind über den gesamten Schießplatz so verteilt, daß die Prozeßrechner möglichst nahe bei der Peripherie und den Bedienstationen in massiven Gebäuden untergebracht werden können ( B i l d 2 ). So befindet sich der zentrale Auswerterechner R30 in einem Betriebstechnikgebäude nahe dem Hauptkontrollturm (HKT) zusammen mit dem Zentralrechner R10 des Sicherheitsnetzes. Beim HKT werden auch die Radaranlagen mit ihren Rechnern installiert.

Jede der fünf Schießbahnen, die an die SASA-Anlage angeschlossen werden, verfügt über einen Schießbahnkontrollturm (SKT). Darin werden die Sicherheitsrechner SKT untergebracht.

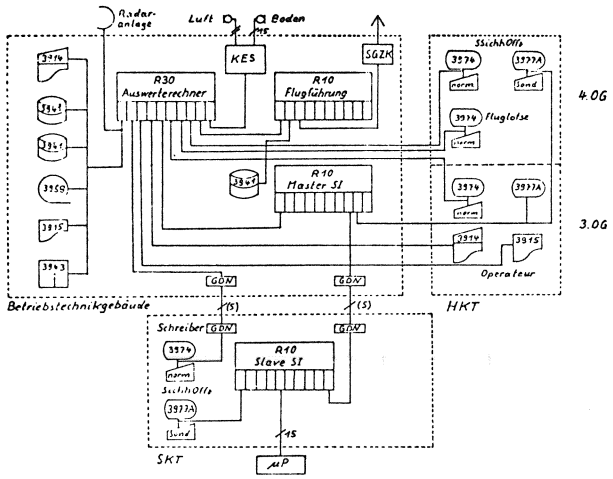


Bild 2. Hardware-Komponenten der SASA-Anlage

Bei den Feuerstellungen in einer Schießbahn werden die Einrichtungen Feuerstellung EF in Geräteboxen untergebracht. Eine EF besteht aus einem Mikroprozessor Intel 8086, einem Mikrophon zur Erfassung des Mündungsknalls, Laser-Winkelmesser, Waffenadapter, Signaleinrichtung und Eingabeeinheit.

Als Ein-Ausgabegeräte dienen Schwarzweiß- und Farbterminals mit Tastaturen, Schnelldrucker und Konsolfernschreiber. Die einzelnen Komponenten sind über Telefonvierdrahtleitungen verbunden, falls nötig mit Nahmodems ausgerüstet.

## 1.3 Software-Komponenten

In diesem Zusammenhang werden die Radarrechner und die Einrichtungen Feuerstellung nicht beachtet, die für die betrachtete Prozeßrechenanlage nur als Peripherie aufgefaßt werden.

Die Anwendersoftware läßt sich in mehrere Komponenten gliedern, von denen einige je nach Ausbaustufe der Gesamtanlage und je nach Betriebszustand zu- oder abgeschaltet werden können ( B i l d 3 ). Jede Komponente besteht aus mindestens einer Task.

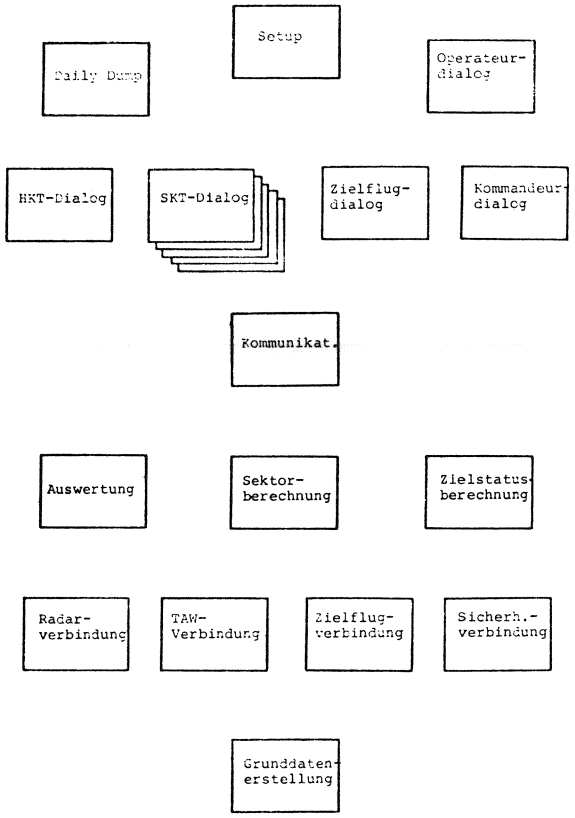


Bild 3. Software-Komponenten der SASA-Anlage

Als zentrales Steuerprogramm für den Gesamtbetrieb dient das Setup-Programm, mit dessen Hilfe auch der Operateurdialog geführt wird. Dies ist ein einfacher Dialog, der hauptsächlich an der Operateurkonsole geführt wird. Er erlaubt das Starten der übrigen Tasks in Abhängigkeit von gewünschten Betriebszuständen, zum Beispiel morgendlicher Neustart des Systems, Wiederstart nach Rekonfiguration oder Netzausfall und das Initialisieren der Listenausdrucke mit den Tagesprotokollen.

Den Arbeitsplätzen des Bedienpersonals sind Dialogprogramme zugeordnet, die universell so entwickelt wurden, daß durch Listensteuerung individuell zugeschnittene Dialoge geführt werden können [3]. Hierzu sind jedem Dialog eventuell mehrere Bildschirmformate zugeordnet ( B i l d 4 ).

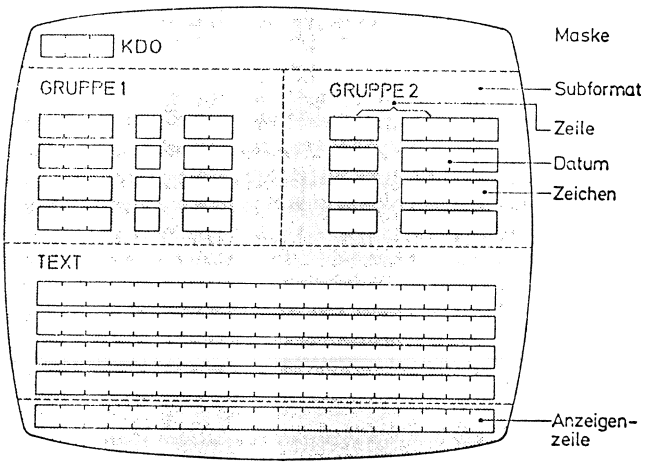


Bild 4. Verallgemeinerte Dialogmaske

Ein Bildschirmformat besteht aus einem Kommandofeld und Subformaten. Ein- und Ausgaben in Subformaten korrespondieren mit speziell zugeordneten Dateien des Datensystems. Subformate sind in logische Eingabeseiten gegliedert und diese wiederum in Einzelzeichen. Nach Eingabe eines Zeichens, bei Abschluß einer Zeile, eines Subformats oder bei Formatwechselkommando kann eine beliebige Nachbearbeitung erfolgen, je nachdem, ob der Eintrag in eine Datei, das Absenden steuernder Nachrichten mit Hilfe des Kommunikationssystems, Formatwechsel, Rollen oder Blättern erfolgen soll.
Das Layout eines Formates, die Zuordnung von Attributen zu den Eingabezeichen und die Abschlußbearbeitung können mit Hilfe spezieller Editoren off line erfolgen.

Das Dialogsystem erlaubt nach seiner Implementierung die vom Nutzer gewünschten Überwachungsfunktionen. So kann sich der verantwortliche Sicherheitsoffizier lesend in andere Dialoge einschalten. Über einen speziellen Dialog können z. B. die Kommandeure einer schießenden Einheit die Schießergebnisse ihrer Soldaten kontrollieren. Derartige Zugriffsrechte sind ebenfalls durch die Art der Nachbearbeitung auf der Basis vorheriger Attributzuordnung möglich.
Die zeichenweise Orientierung der Dialoge erlaubt die Unterbrechung der Eingabe an beliebigen Stellen ( B i l d 5 ). Durch Aufteilung eines Dialogs in Eingabetask und Verarbeitungstask und das Einrichten einer Warteschlange im Empfangsteil der Verarbeitungstask wird es möglich, die Warteschlange sowohl mit Eingabezeichen als auch mit Fehlermessages zu füllen.

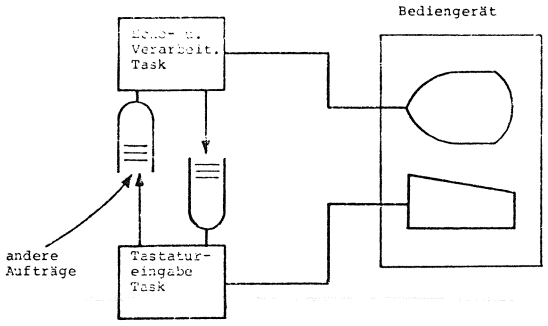


Bild 5. Funktionsprinzip der Dialogeingabe

Die bestehenden Forderungen nach Sicherheit gegen willkürliche oder fahrlässige Fehlbedienungen der Dialoge und nach sofortiger Ausgabe von Fehlermeldungen konnten somit erfüllt werden.

Das Dialogsystem besitzt gegenüber der Datenhaltung eine einheitliche Schnittstelle, die es erlaubt, aus verschiedenen Dations auf Magnetplatten gezielt Datensätze zu lesen und zu schreiben. Durch verschiedene Zugriffsparameter können Dateien unterschiedlich betrieben werden. So werden z. B. Dateien benötigt, die Random-Zugriff auf alle Datensätze erlauben, während andere für lesenden Zugriff nur einen aktuellen Datensatz zurückgeben, für schreibenden Zugriff jedoch einen neuen Datensatz anlegen. Die weiteren Tasks im Auswerterechner bedienen sich der Datenhaltung über die gleichen Schnittstellen. Von besonderer Bedeutung sind dabei die Tasks des Auswertesystems, die aus den von Treibern bereitgestellten Daten aus der Peripherie und über die Datenhaltung zur Verfügung stehenden Grunddaten die Auswertung der Schießergebnisse ermöglichen, wobei mittels der vom Radarrechner gemeldeten Zielkoordinaten eine Rekonstruktion des Schießverlaufs und eine differenzierte Trefferzuordnung beim simultanen Schießen mehrerer Waffen ermöglicht wird. Die Schießergebnisse werden auf den Bildschirmen der entsprechenden Dialoge angezeigt und in Dateien abgelegt. Nach Beendigung der Zuteilung der Schießbahn zu einer Einheit werden bei Beendigung des zugeordneten Schießbahndialogs die Inhalte der Dateien in Listen aufbereitet über Schnelldrucker ausgegeben.

Eine weitere Task berechnet aus den Koordinaten der vom Radargerät erfaßten Objekte auf See und in der Luft die für die einzelnen Schießbahnen zu sperrenden Sektoren. Eine Reihe von Input- und Outputtreibern bedient die jeweiligen seriellen oder parallelen Schnittstellen zur Peripherie. Die ankommenden oder abgehenden Daten sind in Puffern enthalten, wobei die Zugriffe über Semaphore und zugeordnete Kennfelder geordnet werden.

## 2 Erfahrungen bei der Programmierung mit PEARL

### 2.1 Vorgehensweise bei der Entwicklung

Die Entwicklung der Software basiert auf einer zusammen mit dem zukünftigen Nutzer entworfenen Funktionsspezifikation, in der die Abläufe, Bildschirmlayouts, Listen und Datenkataloge festgelegt wurden. Aus den Datenkatalogen wurden Jackson-Datenstrukturen entwickelt. Mit ihrer Hilfe und mit Funktionsdiagrammen wurden die Programmspezifikationen top-down entwickelt.

Zur Dokumentation der Programmspezifikationen wurde ein Pseudocode verwendet, der im Hinblick auf die PEARL-Programmierung die strukturierenden Ausdrücke dieser Sprache enthält. Diese Dokumentation wurde mit Hilfe von Editoren auf dem Entwicklungsrechner R30 durchgeführt.
Auf dem Niveau der Pseudocode-Programme wurden vom Projektteam die sicherheitsrelevanten Abläufe in Walkthroughs kontrolliert.

### 2.2 Programmierung der PEARL-Programme

Mit Hilfe des Editors konnten aus dem Pseudocode die Quellenprogramme entwickelt werden. Der verwendete Sprachumfang ist Basis-PEARL mit den Erweiterungen, die PEARL-300 von der Firma Siemens [1] zur Verfügung stellt. Bezüglich der Sprache gab es für das Programmierteam dabei keine besonderen Schwierigkeiten.

### 2.3 Maßnahmen für Sicherheit und Zuverlässigkeit

Zu den Bedienern der Anlage gehört an den Bildschirmterminals teils wenig geschultes, teils länger dienendes, besonders eingeübtes Personal. Zur Gewährleistung der Sicherheit dienen die besonderen Schutzmaßnahmen in den Dialogen mit enger Bedienerführung. Hierzu sind unter anderem auch einige Tasten der serienmäßigen Tastatur zu blockieren.

Zum Erlangen größtmöglicher Fehlerfreiheit dienen während der Entwicklung verschiedene Kontrollen, die innerhalb des Projektteams gegenseitig durchgeführt werden. Zur dynamischen Kontrolle sind jedoch die den einzelnen Softwaremodulen zugeordneten Testrahmen unerläßlich, die während der Entwicklung notwendig sind, aber auch später für Tests auf allen Integrationsstufen der Anlage zur Verfügung stehen. Die Testrahmen beinhalten einfache Ein- und Ausgabetasks zur Bereitstellung und Kontrolle von Daten und weiterhin Simulationstasks zur Kontrolle von zeitlichen Verläufen und Einstellung verschiedener Belastungsstufen. Zur Programmierung der Testrahmen eignete sich PEARL in hervorragendem Maße, insbesondere wegen der verschiedenen Möglichkeiten, über Zeitbedingungen den Lauf der Tasks zu beeinflussen.

### 2.4 Besondere Probleme und jeweilige Lösungen

Die in diesem Zusammenhang aufgeführten Probleme sind primär keine Probleme der Programmiersprache PEARL. Nach unseren Erfahrungen ist es allerdings nicht möglich, nur mit PEARL-Kenntnissen ein solch komplexes System wie die SASA-Anlage zum Laufen zu bringen. Im folgenden sind einige Beispiele für die aufgetretenen Probleme aufgeführt.

Über das Laufzeitsystem werden in dem verwendeten PEARL-System einige mathematische Funktionen, Zeit- und Datumsabfragen usw. zur Verfügung gestellt. Zeit- und Datums-

eingabe fehlen dabei. Als besonders wichtig erweisen sich in diesem Fall die genaue Dokumentation des Parameterübergabemechanismus und die Manipulation des Stacks. Dann ist es mit Hilfe kleiner Assembler-Routinen möglich, die Library zu vervollständigen.

Der für die Dialoge konzipierte zeichenweise Betrieb einer Eingabetask ist in PEARL nicht vorgesehen, d. h. es stehen keine entsprechenden Standardtreiber zur Verfügung. Mit Hilfe einer kleinen Assembler-Routine kann der gewünschte Betrieb zwar realisiert werden, die zugehörige Task muß aber nunmehr hauptspeicherresident geladen werden. Nun stören die nicht wiedereintrittsfähigen Laufzeitroutinen sehr, die zu jeder Task hinzugebunden werden müssen. Andere Treiber benötigen für ihr Funktionieren spezielle Speicherbereiche (sogenannte reelle Pakete), die wegen anderer Zwänge nur geringen Umfang besitzen. Die PEARL-Tasks lassen sich nur unter Beachtung vieler Randbedingungen funktionsfähig dorthin laden.

Sobald PEARL-Tasks gemeinsame Daten im Hauptspeicher benutzen, muß ein gemeinsamer Globalbereich existieren. Andererseits können Tasks mit ihren Prozeduren bei gemeinsamem Zugriff auf Daten entweder die Parameterübergabe benutzen oder globale Daten verwenden. Falls sehr viele Parameter zu übergeben wären, wird hierfür viel Code und Laufzeit nötig. Falls die Daten global gemacht werden, müßten sie allerdings dem allgemeinen Globalbereich zugefügt werden, der im Falle der SASA-Anlage sehr groß geworden wäre, was Nachteile für die Speicherverteilung, Testbarkeit und Handhabbarkeit des Systems gehabt hätte. Zur Vermeidung dieser Nachteile mußten sehr große lineare PEARL-Programme geschrieben werden, deren Handhabbarkeit allerdings schlecht ist.

Der verwendete Compiler berechnet zur Compilezeit keine Konstantenausdrücke. Falls code- oder laufzeitoptimale Programme benötigt werden, ist dies beim Codieren zu berücksichtigen.

Der verwendete Compiler übersetzt Quellenzeilen ohne Berücksichtigung des Kontextes. Aus diesem Grunde sind möglichst viele Operationen vor einer Zuweisung durchzuführen. Dies führt zwangsläufig zu wenig übersichtlichem Quellentext.

## 3 Ergebnisse bei der Integration der Komponenten

### 3.1 Testergebnisse

Durch die Vorgehensweise bei der Programmentwicklung ergeben sich relativ wenige Probleme bei der Integration der Komponenten. Durch sukzessiven Ersatz von Testrahmen durch endgültige Komponenten wird das System vervollständigt. Die durchgeführten Tests dienen zur Verifikation der Funktionen, wie sie in der Funktionsspezifikation festgelegt wurden. In diesem Rahmen konnte ein sehr großer Teil des Tests, die ohne Radargerät und Trefferauswerteanlage durchführbar sind, zufriedenstellend durchgeführt werden.

### 3.2 Zeitverhalten

Das Zeitverhalten der EDV-Komponenten ist von der Aufteilung des Hauptspeichers in die "virtuellen" Laufbereiche abhängig. Programme in reellen und anderen Paketen und die globalen Prozeduren stehen ständig zur Verfügung. Programme in den Laufbereichen unterliegen einer Überwachung und werden nach Bedarf in Abhängigkeit von Prioritäten bei E/A-Anforderungen und Einplanungen ein- oder austransferiert. Die Verteilung der Tasks des Systems auf Laufbereiche und Pakete beeinflußt in starkem Maße das Zeitverhalten. Zur Optimierung werden umfangreiche Tests mit dem fertigen System notwendig sein.

### 3.3 Fehlerbeseitigung

Die Fehlersuche und -beseitigung in den separat betriebenen Modulen ist mit Hilfe von zur Verfügung stehenden Dienstprogrammen relativ einfach möglich. Die

Dienstprogramme arbeiten jedoch auf Assembler- bzw. Grundspracheniveau, so daß ein PEARL-Programmierer darüber Kenntnisse haben muß.

Im Falle, daß Programmsysteme getestet werden müssen, erweist es sich als unpraktisch, daß Semaphore von einem PEARL-Programm nicht vorbesetzt werden können. Da sie im Commonteil deklariert sind, können sie nur durch Neuladen auf definierte Anfangswerte gesetzt werden - für Tests beim Programmsystem eine oft unerwünschte Vorgehensweise.

Im Projekt erweist sich eine spezielle Hilfstask als besonders notwendig, die mit Hilfe von Assembler-Programmen Manipulationen in den common geladenen Datenbereichen ermöglicht. Sie spart erheblichen Zeitaufwand in der Test- und Integrationsphase ein.

## 4 Ausblick, Erwartungen

Im Gegensatz zu den Zielen der PEARL-Sprachentwicklung läßt sich ein Programmsystem zur Zeit nicht ohne Betriebssystem-, Assembler- und Hardware-Kenntnisse des PEARL-Programmierers zum Ablauf bringen. Falls ständig mindestens ein Mitglied des Projektteams über diese Kenntnisse verfügt, läßt sich ein Programmier- und Testbetrieb nach "Kochrezepten" aufziehen, wobei die Beschränkung auf PEARL-Sprachkenntnisse bis zu gewissen Grenzen möglich ist.
Unbedingt notwendig wäre für diese Arbeitsweise eine Testhilfe auf Sprachebene.
Zur Vermeidung der Assembler-Ergänzungsprogramme wäre die Existenz von Libraries notwendig, die über das heutige Maß hinaus die notwendigen Prozeduren enthalten sollten. Sie sollten mit dem Compiler gekauft werden können und mit entsprechender Dokumentation versehen sein.

## Schrifttum

/1/ Siemens: Beschreibung PC30/PEARL 300. Siemens P71100-D3010-X-A3-35

/2/ Weber, Horst: Einsatz von PEARL bei der Software-Entwicklung für eine Fla-Schießplatz-Automatisierung. PEARL-Rundschau (1980) Nr. 4, S. 26/31

/3/ Weber, Horst und Egner, Karl-Dieter: Ein modulares, universelles Dialogsystem programmiert in PEARL. PEARL-Rundschau (1981) Nr. 6. S. 31/39

## Anschrift des Autors

Dr.-Ing. Horst Weber
Battelle-Institut e. V.
Abt. Technische Informatik
Am Römerhof 35

6000 Frankfurt am Main 90

Tel. (0611) 7908-2508