

**P E D M S****Ein realzeitorientiertes Datenbanksystem  
für die Siemens R30**

von Dipl.-Mathematiker Chr. Mosko

Vortrag auf der Jahrestagung 1983  
des Siemens-Prozeßrechner-Anwenderkreises I  
in Braunschweig

## INHALTSVERZEICHNIS

1. ÜBERBLICK
2. EINLEITUNG
3. ENTWICKLUNGSZIELE
4. AUFBAU DES DATENBANKSYSTEMS PEDMS
5. MASCHINENUNABHÄNGIGE REALISIERUNG
6. REALISIERUNG FÜR DIE SIEMENS R30
7. PROBLEME BEI DER ÜBERTRAGUNG
8. ZUSAMMENFASSUNG

## 1. ÜBERBLICK

Der vorliegende Vortrag stellt die allgemeine Struktur und die Eigenschaft des Datenbanksystems PEDMS (Portable and Extendable Database Management System) vor sowie die Übertragung und Installation des Systems auf dem Rechner Siemens R30.

## 2. EINLEITUNG

In den vergangenen Jahren wurden die Minicomputer ständig leistungsstärker und bezüglich der Speicher ausbaufähiger. Damit wurde es möglich, große Softwarepakete, die eigentlich für Großrechner entwickelt wurden und werden, auf den Minicomputern zu realisieren. Im Besonderen wurden Datenbankrealisierungen sinnvoll. Dabei ergeben sich jedoch im Gegensatz zu Datenbankeinsätzen auf Großrechnern spezielle Anforderungen im Hinblick auf

- Speicherplatz und
- Realzeiteinsatz.

Diese Anforderungen bildeten die Vorgaben für die Entwicklung von PEDMS:

- + Realisierung auf kleinem Hauptspeicher
- + hohe Verfügbarkeit
- + leichte Übertragbarkeit auf verschiedene Maschinen
- + Ausbau in verschiedenen Stufen
- + Verteilte Datenbank.

Im folgenden werden die Entwicklungsziele und der Aufbau von PEDMS vorgestellt sowie die Realisierung auf der R30. Abschließend wird auf einige Probleme bei der Übertragung eingegangen.

### 3. ENTWICKLUNGSZIELE

PEDMS wurde entwickelt für realzeitorientierte Anwendungen, die auch auf verschiedene Prozessoren verteilt sein können. Diese Zielrichtung implizierte folgende Eigenschaften von PEDMS

- Modularität,
- 24-Stunden-Betrieb,
- Verteilte Datenbanken,
- Maschinenunabhängigkeit/Übertragbarkeit.

#### 3.1 Modularität

Ein Datenbanksystem für Realzeitanwendungen, d.h. für Anwendungen auf Prozeßrechnern, muß mit begrenztem Adressbereich und Hauptspeicher- ausbau auskommen. Overlay-Techniken sind dabei nur bedingt möglich, da sie den Durchsatz im Rechner beliebig herabsetzen können. Diese Randbedingungen führten zu den zwei Vorgaben

- Multitasking-System,
- stufenweiser Ausbau.

Das Datenbanksystem PEDMS ist als Multitasking-System realisiert, wobei jede Task - in Siemens-Terminologie jedes Programm - eine spezielle Aufgabe des Datenbanksystems wie z.B. Hinzufügen von Daten, Verwaltung von Puffern und Aufbau von Zugriffspfaden erfüllt. Dadurch, daß jede dieser Tasks in Overlay-Technik mit anderen Tasks in demselben Hauptspeicherbereich ablaufen kann, können die hauptspeicherresidenten Teile des Datenbanksystems auf ein Minimum beschränkt werden. Zur Erhöhung des Durchsatzes auf dem Rechner können sukzessiv bei Ausbau des Hauptspeichers Tasks aus Overlay-Bereichen - in Siemens-Terminologie Laufbereiche - herausgenommen werden und als hauptspeicherresidente Tasks installiert werden.

Das Datenbanksystem PEDMS ist in seinen Funktionen modular aufgebaut, was einen stufenweisen Ausbau ermöglicht. So ist z.B. bei kleineren

Maschinen das Weglassen komfortablerer, jedoch speicherplatz- und laufzeitintensiverer Funktionen des Datenbanksystems möglich. Umgekehrt können bei größeren Maschinen umfangreichere Funktionen zusätzlich installiert werden. Bei speziellen Anwendungsforderungen können zusätzliche benutzerspezifische Erweiterungen (Datenstrukturen und Datenbankfunktionen) implementiert werden.

### 3.2 24-Stunden-Betrieb

Viele Realzeit-Einsätze eines Rechnersystems erfordern einen kontinuierlichen Betrieb, wobei nur minimale oder überhaupt keine Ausfälle des Systems zugelassen werden. Diese Bedingungen fordern von einem in Realzeitumgebung installierten Datenbanksystem die Unterstützung von

- Online-Datenbankverwaltung (Datenbankadministrator),
- Spiegelplatten und Hot-Stand-By-Betrieb,
- Online-Wechsel der Datenbankspeicher,
- Reorganisationsfreiheit,
- alternative Restart- und Recoverymechanismen,
- spezielle Datenstrukturen und Funktionen für Realzeitumgebung.

Die Datenbankverwaltung, d.h. z.B. das Hinzufügen und Ändern von Dateien und/oder Datenstrukturen muß parallel zu den anderen Aktivitäten des Systems online möglich sein.

### 3.3 Verteilte Datenbanken

Speicherplatzbeschränkungen sowie der Einsatz redundanter Hardware bei kritischen Realzeitanwendungen führten zur Forderung an Datenbanksysteme, verteilte Datenbanken zu unterstützen. Dabei kann verteilt zum einen bedeuten redundant, zum anderen disjunkt verteilt, was zu Datenbanknetzwerken führt.

### 3.4 Maschinenunabhängigkeit/Übertragbarkeit

Ein großes Softwaresystem wie ein Datenbanksystem sollte i.b. aus Gründen der Entwicklungskosten nicht für einen speziellen Rechner entwickelt werden, sondern durch weitgehende Maschinenunabhängigkeit auf den verschiedensten Rechenanlagen einsetzbar sein. Diese Bedingungen führten zu zwei Festlegungen für die Entwicklung des Datenbanksystems PEDMS:

- Isolierung der Maschinenabhängigkeit in wenige, kleine, im System separierte Module,
- Entwicklung des Systems in einer höheren Programmiersprache, die auf fast allen Minicomputern verfügbar ist.

Die Maschinenabhängigkeit besteht im wesentlichen in den Problempunkten

- + Bytemanipulation
- + Betriebssystemschnittstellen
- + Plattenspeicher I/O-Operationen.

Für die Programmierung ergab sich zum Zeitpunkt des Beginns der Entwicklung nur FORTRAN als geeignete Sprache.

Bei einer Übertragung des Datenbanksystems von einem Rechner auf einen anderen müssen bis auf minimale FORTRAN-Anpassungen nur die im Assembler realisierten Module umcodiert werden. Diese Module haben am PEDMS-Gesamtsystem einen Anteil von weniger als 5 %.

#### 4. AUFBAU DES DATENBANKSYSTEMS PEDMS

Der Aufbau von PEDMS wird im folgenden in den drei Abschnitten

- Datenmodell,
- Datenstrukturen,
- Datenbankfunktionen

dargestellt.

##### 4.1 Datenmodell

Das Datenmodell von PEDMS verwendet lineare Datenstrukturen, die sich an dem Relationenmodell orientieren. Die Daten sind als Menge verschiedener Datenbanken organisiert, von denen jede aus einer oder mehreren Dateien besteht. Dabei enthalten die Dateien gleichstrukturierte Sätze. Jede dieser Dateien kann als Relation betrachtet werden, wobei die Tupels die Dateisätze sind. Einige Einschränkungen und Erweiterungen im Vergleich zu relationalen Systemen sind folgende:

- Jedes Feld in einer PEDMS-Datei kann mehrwertig sein. Diese Option kann zur Darstellung nicht - normalisierter Relationen benutzt werden, was in der Praxis bei Datenstrukturierung und bzgl. der Zugriffszeiten wesentliche Vorteile gegenüber dem reinen relationalen Modell bringt.
- Für jede Relation kann eine beliebige Anzahl von Schlüsseln (Deskriptoren) definiert werden. Für jeden solchen Schlüssel wird eine dynamisch verwaltete Zugriffsstruktur - ähnlich einem B-Baum - aufgebaut.
- Die Eigenschaft eines Feldes, Suchschlüssel zu sein, muß nicht permanent sein. Es ist möglich, dynamische Felder zu Schlüsseln zu deklarieren oder die Schlüsseleigenschaft zu entfernen.
- Durch logische Kombination mehrerer Schlüssel können komplexe Auswahlbedingungen formuliert werden.
- Möglichkeit der Anwendung von Hash-Algorithmen für zeitkritische Anwendungen.

#### 4.2 Datenstrukturen

Die für den Benutzer von PEDMS verfügbaren Daten sind organisiert als Hierarchie von Strukturen. Die oberste Schicht ist die Datenbasis. Diese enthält die Beschreibungen einer oder mehrerer Datenbanken zusammen mit den Informationen, auf welchen Rechnern jede der Datenbanken existiert.

Die zweite Schicht der Hierarchie besteht aus den Datenbanken, von denen jede einzelne komplett auf einem Rechner des Rechnernetzes vorhanden sein muß. Jede Datenbank ist in zwei rechnerspezifischen Dateien realisiert. Dabei enthält die erste Datei die Benutzerdaten, die zweite die Verwaltungsdaten der Datenbank wie z.B. Schemainformationen und Zugriffspfade.

Die dritte Schicht bilden die verschiedenen Relationen, auch logische PEDMS-Dateien genannt. Diese Dateien sind die Zusammenfassung von Dateisätzen gleicher Struktur.

Die vierte Schicht bilden die Datenfelder. Ein oder mehrere Felder bilden einen Satz einer logischen PEDMS-Datei. Jedes Feld kann verschiedene Eigenschaften besitzen. Diese Eigenschaften sind in der Schemabeschreibung definiert. Alle Eigenschaften sind unabhängig voneinander.

Ein Feld hat folgende Eigenschaften:

- + Schlüssel-/Normalfeld
- + Länge • (feste/variable Länge)
- + Typ (Integer, Real, alphanumerisch, numerisch, binär)
- + Wertigkeit (einwertig/mehrwertig).

Bei Feldern fester Länge ist die Längenangabe in der Schemainformation abgelegt, bei Feldern variabler Länge steht die Länge bei den Feldwerten. Einwertige Felder müssen genau einen Wert pro Satz enthalten, während mehrwertige Felder keinen, einen oder mehrere Werte pro Satz enthalten können.

Der Status eines jeden Datenfeldes bestimmt die möglichen Zugriffspfade

auf dieses Feld:

- Wenn kein Feld als Suchschlüssel definiert ist, sind nur sequentielle Zugriffe möglich.
- Wenn ein Feld als Suchschlüssel definiert ist, ist sequentieller, logisch sequentieller und direkter Zugriff auf die Sätze einer Datei möglich.
- Die Definition verschiedener Zugriffsschlüssel erlaubt den sequentiellen, logisch sequentiellen, direkten und assoziativen Zugriff auf die Sätze einer Datei.
- Für zeitkritische Anwendungen kann ein direkter Zugriff, der eine lineare Transformation benutzt, eingesetzt werden.

#### 4.3 Datenbankfunktionen

Die Datenbankfunktionen von PEDMS werden in Grund- und Verwaltungsfunktionen eingeteilt. Die Grundfunktionen gliedern sich in

- + DATENBANKBEFEHLE:
  - ANMELDEN/ABMELDEN
  - HALTEN/FREIGEBEN
  - SUCHEN mit Auswahlkriterium
  - LESEN random, sequentiell, direkt
  - ÄNDERN (hinzufügen, ändern, löschen von Feldern)
  - LÖSCHEN
  - NEUZUGANG
  - TRANSAKTION
  - SPEZIALBEFEHLE
  
- + KOORDINATIONSFUNKTIONEN:  
(Multi-User, dezentrale Datenbanken)
  
- + DATENSICHERUNGSFUNKTIONEN:  
(REDO, UNDO, Doppelschreiben)

+ DATENSCHUTZFUNKTIONEN:

(Datenbasis-, Datenbank-, Datei- und Feldebene)

Als Verwaltungsfunktionen stehen zur Verfügung:

LADEN STRUKTURDATEN (DDL) / LÖSCHEN STRUKTURDATEN (DDL)

LADEN PRIMÄRDATEN / LÖSCHEN PRIMÄRDATEN

ERKLÄREN DESKRIPTOR (DDL) / LÖSCHEN DESKRIPTOR (DDL)

ÄNDERN STRUKTURDATEN (DDL)

DATENBANK-KOPIE

WIEDERANLAUF

DATENSCHUTZ (incl. Subschema-Verwaltung)

DATENBANK-REPORT

PEDMS-REPORT

Im Gegensatz zu existierenden Datenbanksystemen können die Verwaltungsfunktionen ohne Reorganisation der bereits existierenden Datenbanken eingesetzt werden. Darüberhinaus können die meisten Verwaltungsfunktionen parallel zum normalen Betrieb des Datenbanksystems laufen. Die Steuerung der Verwaltungsfunktionen erfolgt wahlweise im Dialog oder Batch über eine eigene Steuerungssprache.

## 5. MASCHINENUNABHÄNGIGE REALISIERUNG

In diesem Abschnitt wird zum einen die generelle interne Struktur von PEDMS diskutiert, zum anderen die für Prozeßrechner relevante Umsetzung in ein Multitasking-System.

### 5.1 Interne Struktur

Die PEDMS-Routinen gliedern sich vereinfacht in 6 Ebenen

- (1) Linkmodul
- (2) PEDMS-Kontrollprogramm
- (3) Auftragsannahme/Rückgabe
- (4) Module für die Datenbankoperationen
- (5) Dienstfunktionen für Auftragsannahme/Rückgabe
- (6) Pufferverwaltung.

Der Linkmodul ist die vom Anwender an seine Software anzubindende PEDMS-Komponente. Der Modul führt Vorbereitungsfunktionen für das Datenbanksystem sowie die Bereitstellung der vom Anwender gelieferten Daten für PEDMS durch.

Das PEDMS-Kontrollprogramm steuert die PEDMS-internen Abläufe und verwaltet die Warteschlangen für die einzelnen Verarbeitungsmodule.

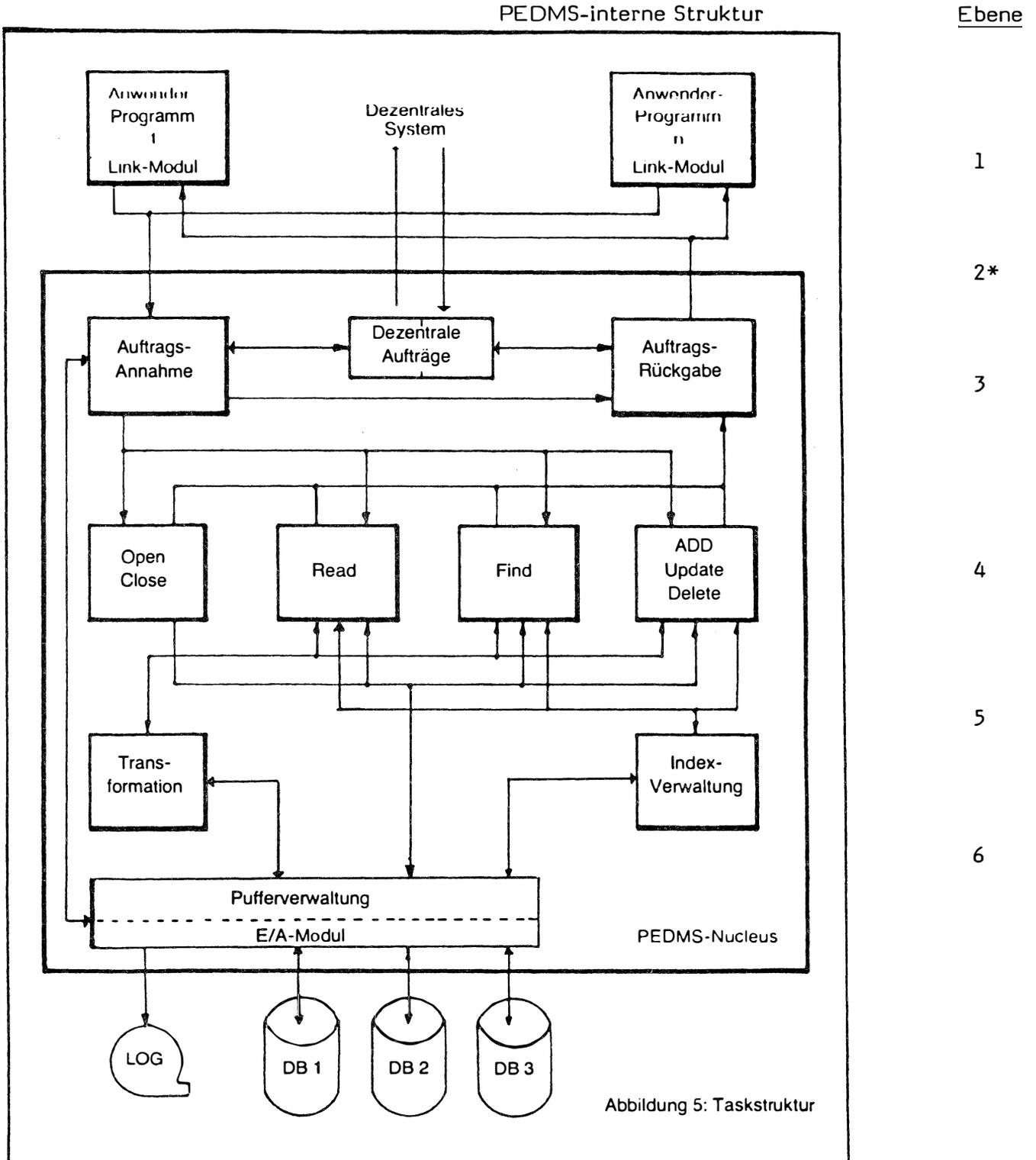
Die Auftragsannahme ist für jeden PEDMS-Auftrag der Eingangsmodule.

Die Auftragsrückgabe ist für jeden PEDMS-Auftrag der Ausgangsmodule zum Linkmodul des Anwenders.

Die Module für die Datenbankoperationen bearbeiten die PEDMS-Aufträge unter Benutzung der Dienstfunktionen sowie der Pufferverwaltung.

### 5.2 Inter-Task-Kommunikation

Die Realisierung von PEDMS auf Prozeßrechnern nutzt die in diesen Rechnern vorhandenen Möglichkeiten des Multitaskings, der Synchronisation sowie der Botschaftsdienste aus. Das erlaubt die Ersetzung des PEDMS-Kontrollprogrammes durch von den Betriebssystemen bereitgestellte Funktionen. Es ergibt sich folgende Softwaremodulstruktur.



\* Ebene 2 ist realisiert durch Betriebssystemdienste

Jede Task kommuniziert nur mit der sie beauftragenden Task der nächst höheren Ebene und mit der von ihr beauftragten Task der nächst tieferen Ebene mit der Ausnahme der Pufferverwaltungstask, die auch aus der Ebene 3 beauftragt wird.

Dadurch, daß die Betriebssystemdienste für Warteschlangen vor den einzelnen Tasks ausgenutzt werden und weil bei den Dienstfunktionen und der Pufferverwaltung Wartezustände auftreten, können parallel verschiedene Anwender gleichzeitig mit PEDMS arbeiten. Dabei kann es auch zu Überholvorgängen kommen.

### 5.3 Betriebssystemschnittstelle

Zur Realisierung des Multitaskingssystems sind folgende Betriebssystemfunktionen erforderlich:

- Aktivieren einer fremden Task,
- Warten auf Aktivierung,
- Betätigen einer Semaphorvariablen (+/-), mit Datenübergabe zur Realisierung von Warteschlangen.

Für die Realisierung von PEDMS ist auch die Möglichkeit der systemweiten Benutzung eines gemeinsamen Adreßraumes (Global-Common-Bereich) notwendig bzw. wenn dieses nicht möglich ist, die Umschaltung auf fremde Adreßräume.

### 5.4 I/O-Schnittstelle

Alle Platten -I/O-Operationen (Read, Write, Open, Close) sind in einem Modul zusammengefaßt. Dabei erfolgen alle Lese- und Schreibzugriffe in wählbar Vielfachen der rechner-spezifisch physikalischen Blockgröße.

## 6. REALISIERUNG FÜR SIEMENS R30

### 6.1 Grundsätzliches

Die R30 kennt Pakete, die jeweils max. 64KW groß sein können. Werden aus einem Paket Common-Bereiche angesprochen, so gilt folgende

Restriktion

$$\text{Paketlänge} + \sum_i \text{Common}_i \leq 64\text{KW}$$

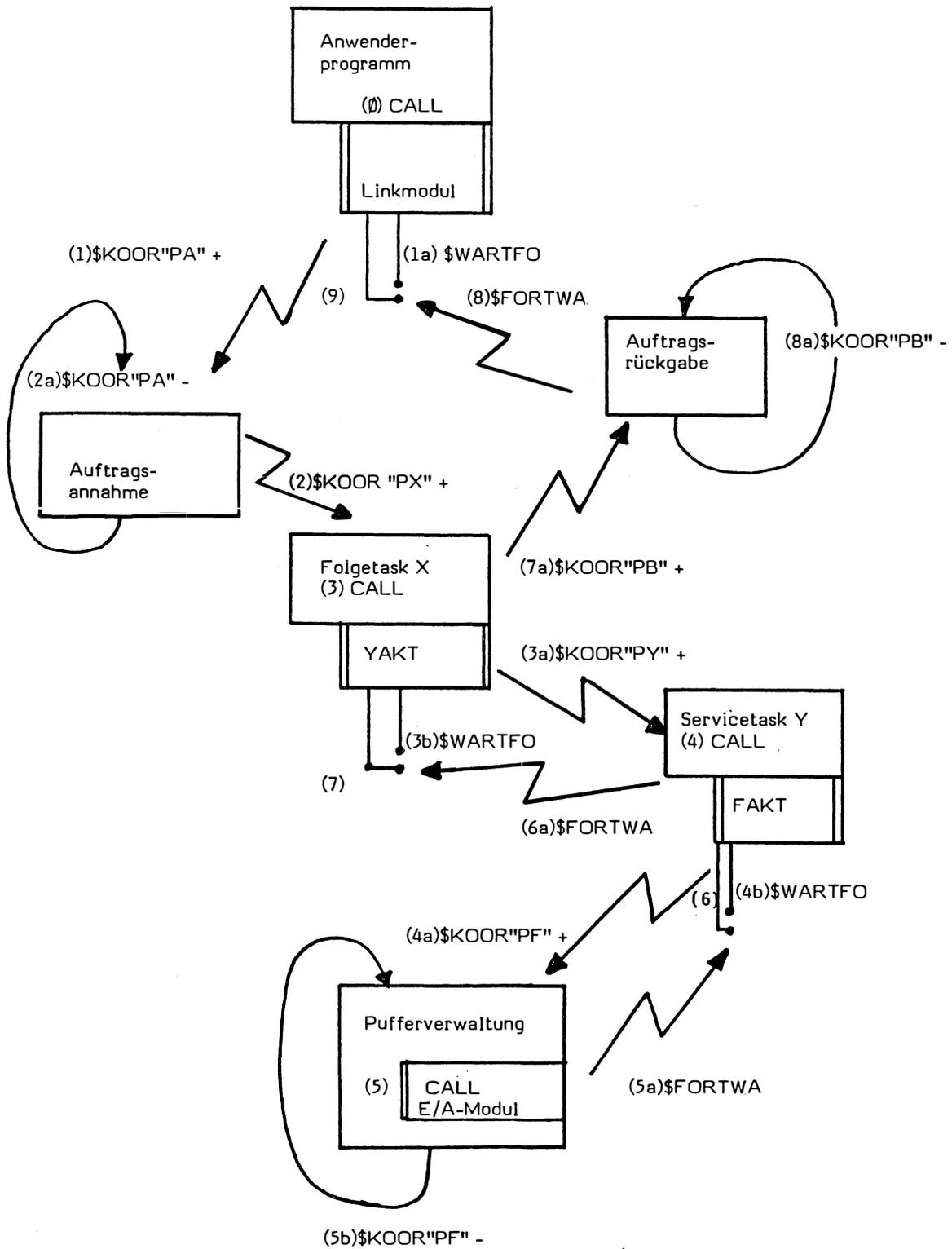
Auf alle diese Common-Bereiche kann aus FORTRAN heraus direkt zugegriffen werden, sofern beim Binden entsprechende Vorkehrungen getroffen werden.

### 6.2 Konkrete Realisierung

Es wird ein PEDMS-Paket von 64KW eingerichtet. Dieses Paket hat folgenden Aufbau (siehe Bild 5.2)

- a) Common für Verständigungsbereichkopieen
- b) PEDMS-Common
- c) Auftragsannahme (A-Task hauptspeicherresident)
- d) Auftragsrückgabe (B-Task hauptspeicherresident)
- e) Pufferverwaltung (F-Task hauptspeicherresident)
- f) Laufbereich für E-, G-, H-, I- und C-Task
- g) Laufbereich für D-, K- und V-Task.

Im folgenden ist der Ablauf der Intertask-Kommunikation dargestellt:



Der Linkmodul, der zu jedem PEDMS-Anwenderprogramm hinzugebunden werden muß, wird als FORTRAN-Subroutine mit 6 Parametern aufgerufen. Der Linkmodul ist eine Assembleroutine, die den Common für die Verständigungsbereichkopieen anschließt, in diesem Common einen freien Bereich belegt, die Anwenderpuffer hineinkopiert und die Auftragsannahme A-Task durch Erhöhen des dieser Task zugeordneten Koordinierungszählers beauftragt. Anschließend wartet der Linkmodul auf die Fortsetzung durch die Auftragsrückgabe (B-Task) im Normalfall oder kehrt bei Quick Return sofort mit einem Return in das Anwenderprogramm zurück.

Wenn die PEDMS-Programme den gewünschten Auftrag fertig bearbeitet haben (im Common für die Verständigungsbereichkopieen), setzt die B-Task den wartenden Linkmodul fort. Dieser kopiert die von PEDMS bearbeiteten Daten aus dem Common für die Verständigungsbereichkopieen in die lokalen Anwenderpuffer zurück, gibt den belegten Common-Bereich frei und führt den Return zum FORTRAN-Aufrufer durch. Beim Quick Return kopiert die Auftragsrückgabe die entsprechenden Datenbereiche ohne Fortsetzung des Linkmoduls, da dieser nicht gewartet hat. Die übrigen Abläufe sind aus dem Ablaufdiagramm ersichtlich.

## 7. Probleme bei der Übertragung

Bei der Übertragung der PEDMS-Module auf die R30 waren folgende Arbeiten notwendig:

- Übertragung der maschinenunabhängigen FORTRAN-Module per Magnetband unter Verwendung des Programms DIMOS.
- Erstellung der Rahmen der A-, B-Tasks usw.
- Erstellung des Linkmoduls und der Common-Bereiche.
- Erstellung des Datei-Verwaltungsprogramms.
- Test der Einzelmodule sowie Integrationstest.

Bei diesen Arbeiten stellten sich einige Schwierigkeiten heraus, die zu Wünschen für die Leistungen der Siemens-Systemsoftware führen.

Zugrundegelegt wurde der FTN-300-Compiler. Dieser bietet keine Unterstützung für Multitasking.

Weiterhin ist es störend, daß die Parameterüberprüfung bei FORTRAN-Calls nicht abschaltbar ist (i.b. Parameteranzahlprüfung). Es wäre weiterhin von Vorteil, wenn man im Hauptprogramm definierte Integer-Größen in einer Subroutine als Character-Größe verarbeiten könnte. Dieses Problem ist z.B auf der VAX von DEC durch den Datentyp BYTE gelöst. Aus Flexibilitäts- und Laufzeitgründen sollte es möglich sein, beim Compilieren bzw. Binden die Parameterüberprüfung wahlweise wegzulassen.

Als systembedingte Schwäche machten sich die fehlenden Bytemanipulationsbefehle bemerkbar, i.b. daß es keinen Move-Befehl für Byte-Felder gibt.

Ein für den Test der Programme relevanter Mangel ist, daß Integer\*2-Felder nicht im A-Format bearbeitet werden können.

## 8. ZUSAMMENFASSUNG

Zum Abschluß möchte ich aufzeigen, auf welchen Rechnern der verschiedensten Hersteller PEDMS zur Zeit installiert ist:

DEC PDP11/23, PDP11/34, PDP11/44 unter RSX-11M

DEC VAX-11 unter VAX/VMS

MODCOMP IV/Classic unter MAX IV

ATM 80-30 unter ATMOS

AEG 80-40/60 unter MARTOS

Siemens R30 unter ORG PV

Für alle diese Installationen müssen wir Wartungsarbeiten und ggf. Erweiterungen durchführen. Das ist nur mit vertretbarem Aufwand möglich, weil über 95 % der Datenbanksoftware den Standard FORTRAN X3.9-66 einhält. Für die Wartung der maschinenabhängigen Assembler-routinen benötigen wir jeweils einen Spezialisten für das betreffende System.

Lassen Sie mich an dieser Stelle meinen Kollegen, insbesondere den Datenbankspezialisten, Herrn Dr. Weck und Herrn Henselmann für die Unterstützung bei der Erstellung und der Durchsicht des Vortragsmanuskripts danken.