KERNFORSCHUNGSZENTRUM

KARLSRUHE

April 1974 KFK 1967

Institut für Datenverarbeitung in der Technik

Koordination kritischer Zugriffe auf verteilte Datenbanken in Rechnernetzen bei dezentraler Überwachung

E. Holler

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1967

Institut für Datenverarbeitung in der Technik

Koordination kritischer Zugriffe auf verteilte Datenbanken in Rechnernetzen bei dezentraler Überwachung

Elmar Holler

von der Fakultät für Informatik der Universität Karlsruhe genehmigte Dissertation

Gesellschaft für Kernforschung mbH, Karlsruhe

Kurzfassung:

Über ein Rechnernetz verteilte Datenbanken erfordern ein koordiniertes Vorgehen bei der Durchführung kritischer Zugriffe, wenn die Konsistenz der in den Datenbankkomponenten abgelegten oder durch simultane Abfrage mehrerer Datenbankkomponenten gewonnenen Information gewährleistet werden soll.

Der Bericht zeigt Möglichkeiten der dezentral organisierten Koordination kritischer Zugriffe zu verteilten Datenbanken in Rechnernetzen auf. Mehrere Alternativlösungen werden vorgestellt und bezüglich ihrer Leistungsfähigkeit mittels eines Rechnernetz-Simulationsmodells experimentellen Vergleichstests unterzogen.

Abstract:

Coordination of critical access to distributed databanks in computer networks based on decentralized control

Distributed databanks in computer networks require special coordination of critical access in cases where two or more components of a databank are engaged in the same update or query process in order to preserve consistency of stored and retrieved information.

This report presents several solutions to the coordination problem, applying a decentralized control mechanism. The different solutions are compared relative to their efficiency by means of experiments using a computer network simulation model.

Inhaltsverzeichnis Seite

1. Einführung

1.1. Verteilte Datenbanken in Rechnernetzen

Die zunehmende Vervollkommnung der Kommunikationstechnologie und ihre Verfügbarmachung auf breiter Basis, etwa durch öffentliche Datentransportnetze, rückte in den letzten Jahren den Zusammenschluß von Rechnersystemen zu Rechnerverbundsystemen in den Bereich der Realität und führte zu Realisierungen derartiger Systeme für verschiedene Anwendungen /1/,/4/,/17/,/24/.

Von seinem statischen Erscheinungsbild her betrachtet, besteht ein Rechnernetz oder Rechnerverbundsystem, wie in Abb. 1.1-1 dargestellt, aus drei wesentlichen Komponenten (vgl. /1/,/11/,/23/,/27/):

- einer Menge von mehreren Rechnern, auch Knoten des Netzes genannt, die zur Bearbeitung von Aufträgen eingesetzt werden können,
- einem Kommunikationssystem, das Kommunikationswege für die Übertragung von Informationen zwischen den Rechnern bereitstellt,
- mehreren Ein- und Ausgabestationen, über die Benutzer Aufträge an das System absetzen und die Resultate der Auftragsbearbeitung entgegennehmen können.

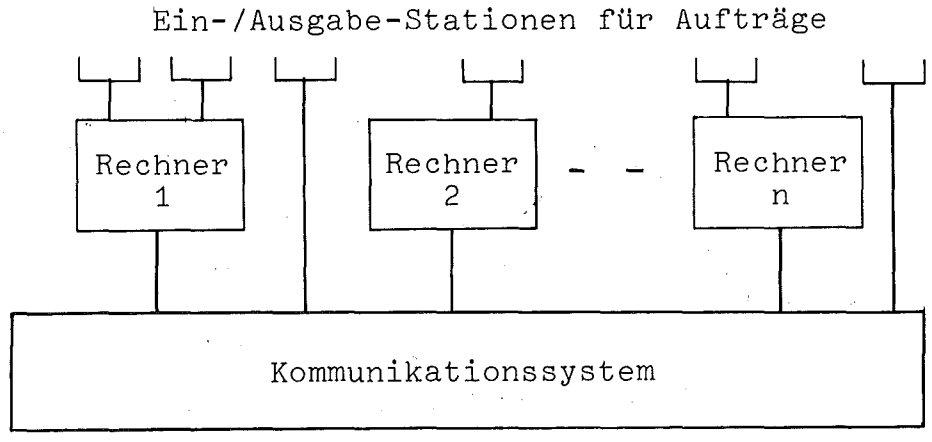


Abb. 1.1-1 Schema eines Rechnernetzes

Das aktive, auftragsbearbeitende Rechnernetz kann nach Bell /1/ aufgefaßt werden als eine Menge in Wechselwirkung befindlicher Prozesse, von denen jeder genau einem der Rechner zugeordnet ist. Innerhalb der Menge dieser Prozesse ist zu unterscheiden zwischen Benutzerprozessen, die unmittelbar mit der Bearbeitung von Benutzeraufträgen zusammenhängen, und Systemprozessen mit organisatorischen Aufgaben. Für die Kommunikation zwischen Prozessen, die nicht im gleichen Rechner lokalisiert sind, stellt das Kommunikationssystem die benötigten logischen Übertragungswege zur Verfügung.

Der Grund für den Aufbau von Rechnernetzen ist die durch den Zusammenschluß der Rechner ermöglichte gemeinsame Benutzung von Betriebsmitteln, wie Hardwareeinrichtungen, Softwarepakete usw.

Eine besondere Stellung unter diesen Betriebsmitteln, die durch Schaffung von Rechnernetzen einem größeren Benutzerkreis für Auftragsbearbeitungen verfügbar gemacht werden, nehmen die Dateien ein.

Dateien sind Realisierungen anwendungsorientierter Datenbestände. Voneinander direkt abhängige verwandte Datenbestände bilden die Komponenten von Datenbanken, die als Elemente von Datenbasen unter anderem für den Aufbau von Informationssystemen benötigt werden /16/. Abb. 1.1-2 zeigt den schematischen Aufbau einer Datenbasis.

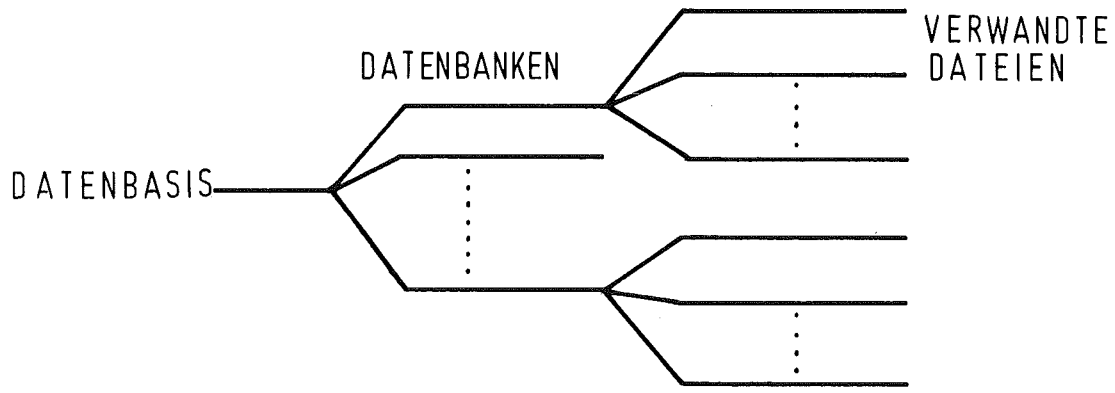


Abb. 1.1-2 Datenbanken und ihre Komponenten als Elemente einer Datenbasis

Beispiel für eine Datenbank ist ein System von Datenbeständen, von denen einige die Inhaltsverzeichnisse der übrigen sind. Die Menge aller Realisierungen eines Datenbestandes stellt ebenfalls eine Datenbank dar.

Eine verteilte Datenbank entsteht, wie in Abb. 1.1-3 dargestellt, durch Verteilung ihrer Komponenten über mehrere Rechner eines Rechnernetzes. Dies bietet, gegenüber der zentralisierten Form einer Datenbank, folgende Vorteile:

- durch mehrfache (redundante) Realisierung der Datenbestände läßt sich eine höhere Zuverlässigkeit und Verfügbarkeit des Datenbanksystems erzielen,
- durch geeignete Verteilung der Datenbankkomponenten kann eine Minimierung der operativen Kosten und eine Steigerung der Effizienz erreicht werden.

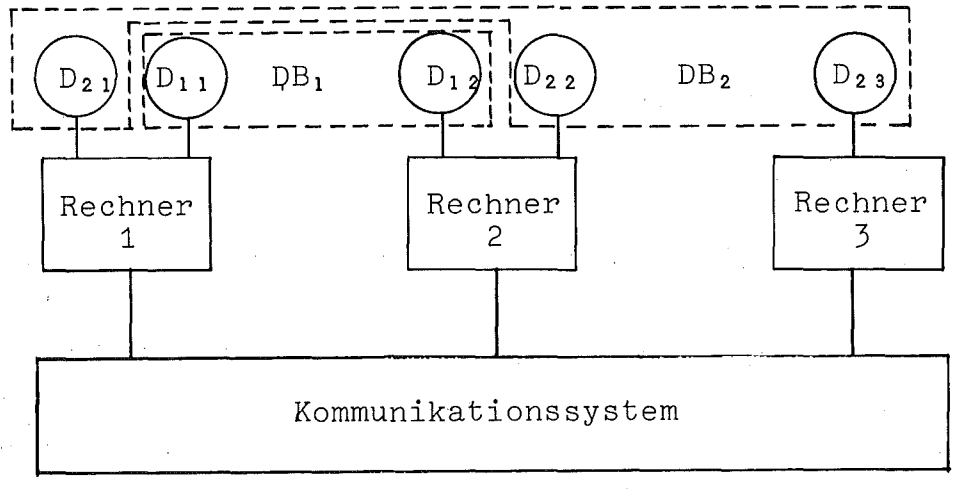


Abb. 1.1-3 Rechnernetz mit den verteilten Datenbanken DB_1 und DB_2. DB_1 umfaßt die Dateien D_{11} und D_{12}, DB_2 die Dateien D_{21}, D_{22}, D_{23}

Zuverlässigkeit und Verfügbarkeit sind z.B. dort unerläßlich, wo eine Datenbank als Systemdatenbasis in Realzeitsystemen Informationen für die Ablaufsteuerung und Überwachung technischer Prozesse (Fertigungsprozesse, chemische Prozesse etc.) enthält. In diesen Fällen wird die entscheidende Datenbank von jedem benötigten Datenbestand mehr als eine Realisierung umfassen. Bei der Verteilung dieser Dateien im Rechnernetz wird man dafür sorgen, daß die Wahrscheinlichkeit ihres gleichzeitigen Ausfalls minimal wird.

Wann die Verteilung der Komponenten einer Datenbank und die redundante Realisierung einzelner Datenbankkomponenten zur Minimierung der Kosten führt, soll im folgenden anhand eines einfachen Optimierungsmodells erläutert werden.

1.2. Optimale Verteilung von Datenbankkomponenten

Wir betrachten eine Datenbank mit m Komponenten, die über ein n Knoten umfassendes Rechnernetz verteilt werden sollen.

Sei

$$X_{kj} = \begin{cases} 1 \text{ wenn Komponente } j \text{ an Knoten } k \\ 0 \text{ sonst} \end{cases} \qquad z_j = \sum_k X_{kj}$$

mit k=1(1)n und j=1(1)m, wobei wir für die Zahl z_j der Realisierungen eines Datenbestandes j auch $z_j > 1$ zulassen wollen. Vereinfachend sei angenommen, daß die Kosten für die Übertragung von Datenblöcken konstanter Länge zwischen zwei Knoten i und k durch ω_{ik}, und die Kosten für die Speicherung einer Datei j an Knoten k pro Zeiteinheit gegeben sind durch σ_{kj}.

Sei ferner ϕ_{ij} der Umfang des Abfrageverkehrs und ψ_{ij} der Umfang des Änderungsverkehrs (in z.B. Bytes pro Zeiteinheit) zwischen dem Knoten i und Datenbestand j.

Für die pro Zeiteinheit anfallenden Gesamtkosten C_{tot} der Datenbank erhalten wir dann, unter Vernachlässigung des Verwaltungsaufwands

(1.2-1)

$$C_{tot} = \sum_{j=1}^{m} \left[\sum_{i,k=1}^{n} \psi_{ij}\omega_{ik}X_{kj} + \sum_{i=1}^{n} \phi_{ij}g_i(I_j) + \sum_{k=1}^{n} \sigma_{kj}X_{kj} \right] = \sum_{j} C_j$$

Änderungskosten Abfragekosten Speicherkosten

C_j gibt den Kostenanteil von Datenbestand j bei unabhängiger Realisierbarkeit, $g_i(I_j)$ die Kosten, die bei einer Abfrage des Datenbestands j von Knoten i aus entstehen, wobei angenommen wird, daß dazu aus der Menge der Realisierungen I_j des Datenbestands j die optimale Datei, z.B. gemäß

(1.2-2) $$g_i(I_j) = \min_{k \varepsilon I_j} \omega_{ik}$$

ausgewählt wird /5/.

Die gesuchte optimale Verteilung der Datenbank ergibt sich durch geeignete Bestimmung der X_{kj} derart, daß (1.2-1) unter gegebenen Randbedingungen, wie z.B. bei Abfragen einzuhaltende obere Schranken für die Antwortzeiten, minimal wird. Das Optimierungsproblem wird in /5/,/6/ und /46/ ausführlich behandelt. Die Resultate zeigen, daß die Kostenoptimierung in den meisten Fällen nur dann erreicht wird, wenn X_{kj} unabhängig von j für unterschiedliche Werte von $k \varepsilon \{1,\ldots,n\}$ gleichzeitig den Wert 1 annimmt, was gleichbedeutend mit der Einrichtung einer verteilten Datenbank ist.

Es soll nun nach einer hinreichenden und notwendigen Bedingung dafür gesucht werden, daß die mehrfache Realisierung einer Datenbankkomponente ökonomischer ist als ihre einfache Realisierung. Dazu ist der Kostenanteil C_j einer Komponente j an den Gesamtkosten (1.2-1) zu betrachten. Da nur eine Komponente unter-

sucht wird, kann im weiteren der Index j weggelassen werden.

Ohne Beschränkung der Allgemeinheit sei angenommen, daß C^1 die Kosten für die optimale Einfach-Realisierung des Datenbestandes angibt. Dann gilt für alle anderen Einfach-Realisierungen

$$C^k = C^1 + \delta_k \quad \text{mit } \delta_k \geq 0 \quad k \in \{1,\ldots,n\} \tag{1.2-3}$$

Unter Anwendung von (1.2-1), (1.2-2) und (1.2-3) läßt sich die Differenz

$$C - C^1 = \Delta C(\gamma) \tag{1.2-4}$$

der Kosten der z-fachen Realisierung und der optimalen Einfach-Realisierung in Abhängigkeit vom Verhältnis γ des Änderungsverkehrs zum Abfrageverkehr ermitteln. Die positive Nullstelle dieser Differenz sei γ_{MAX}. Eine z-fache Realisierung des Datenbestandes mit $z \geq 2$ ist genau dann ökonomischer als die Einfach-Realisierung, wenn γ an den einzelnen Knoten des Rechnernetzes die obere Schranke γ_{MAX} nicht überschreitet /24/.

Für spezielle Typen von Rechnernetzen, mit gleichen Speicherkosten an allen Knoten und gleichen Übertragungskosten zwischen benachbarten Knoten, die als symmetrisch und homogen charakterisiert werden können - vgl. Abb. 1.2-1 -, ist es möglich, γ_{MAX} in geschlossener Form anzugeben. Seien $\Delta Q(z)$ die ersparten Abfragekosten, die sich bei Mehrfach-Realisierung eines Datenbestandes gegenüber der Einfach-Realisierung ergeben, $\Delta\sigma(z)$ die durch Mehrfach-Realisierung entstehenden zusätzlichen Speicherkosten und Q_1 die bei der optimalen Einfach-Realisierung entstehenden totalen Abfragekosten. Für die obere Schranke γ_{MAX} des maximal zulässigen Verhältnisses von Änderungsverkehr zu Abfrageverkehr ergibt sich dann

$$\gamma_{MAX} = \frac{\Delta Q(z) - \Delta\sigma(z)}{Q_1 \cdot (z-1)} \qquad \text{für } z \geq 2 \tag{1.2-5}$$

Wenn immer der zu erwartende Anteil des Änderungsverkehrs in einem Dateiensystem dazu führt, daß der resultierende Wert von γ den Wert γ_{MAX} nicht überschreitet, stellt die z-Knoten-Realisierung die ökonomischste Lösung dar.

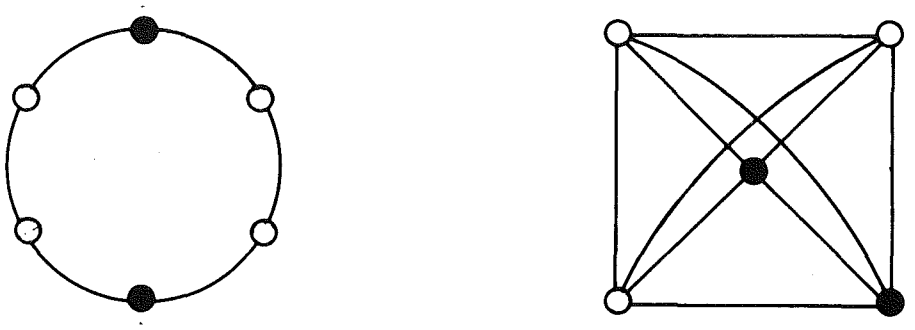


Abb. 1.2-1 Beispiele für homogene symmetrische Rechnernetze. Die gefüllten Kreise kennzeichnen die optimale Verteilung einer Datenbank mit zwei Komponenten.

1.3. Kritische Zugriffe auf verteilte Datenbanken

Mit der Einrichtung verteilter Datenbanken in Rechnernetzen tritt, ungeachtet der damit verbundenen Zielsetzung, das Problem der Handhabung der verteilten Dateien in den Vordergrund. Die Behandlung unabhängiger Abfragen einzelner Dateien ist dabei unkritisch. Problematisch wird jedoch, wie auch im Falle zentralisierter Datenbanken, die Ausführung kritischer Zugriffe /40/.

Unter kritischen Zugriffen zu Datenbanken seien im folgenden alle Zugriffe verstanden, die die exklusive Zuweisung von Teilen einer Datenbank oder einer Datenbank insgesamt als Betriebsmittel zu den Zugriff verlangenden Prozessen erforderlich machen.

Kritische Zugriffe sind damit:

- Datenbankänderungen, da sie die exklusive Benutzung von Datenbankkomponenten durch den Prozeß erfordern, der die Änderung durchführt; die parallele Änderung der Realisierungen von Datenbeständen durch mehrere Prozesse gleichzeitig kann die Konsistenz der abgespeicherten Informationen stören.

- Datenbankabfragen, die eine simultane Informationsentnahme aus mehreren Datenbankkomponenten verlangen. Hier muß durch exklusive Zuweisung der betroffenen Datenbankkomponenten die Konsistenz der gewonnenen Information gewährleistet werden.

Als Beispiel für kritische Zugriffe der letzteren Art seien Informationsentnahmen aus einer Datenbank angeführt, die eine Personaldatei und eine Gehaltsdatei umfassen möge. Die Änderung z.B. des Namens eines in der Datenbank erfaßten Individuums führt zu Änderungen der Informationen sowohl in der Gehaltsdatei als auch in der Personaldatei; eine zeitlich zwischen diesen beiden Änderungen liegende bzw. eine durch die Änderungen unterbrochene Abfrage, die beide Dateien einbezieht, würde zu inkonsistenten Informationen über den Status des Betroffenen führen. Nur durch eine exklusive Benutzung aller betroffenen Datenbankteile (Dateien), kann hier die konsistente Informationsgewinnung sichergestellt werden.

Die Koordination kritischer Zugriffe auf eine Datenbank bei zentraler Überwachung, wie dies sowohl im Falle einer zentral abgelegten als auch bei einer verteilten Datenbank möglich ist, kann durch Anwendung konventioneller Synchronisationsverfahren /14/, /19/,/35/ vorgenommen werden. Dies geschieht in der Regel derart, daß die um den exklusiven Zugriff konkurrierenden Prozesse bei Ausführung des Zugriffs einen sogenannten "kritischen Abschnitt" durchlaufen müssen /14/. Durch Ausführung von Synchronisationsoperationen vor und nach dem Durchlaufen der kritischen Abschnitte kann erreicht werden, daß sich jeweils nur einer der einen kritischen Zugriff durchführenden Prozesse in seinem kritischen Abschnitt befindet.

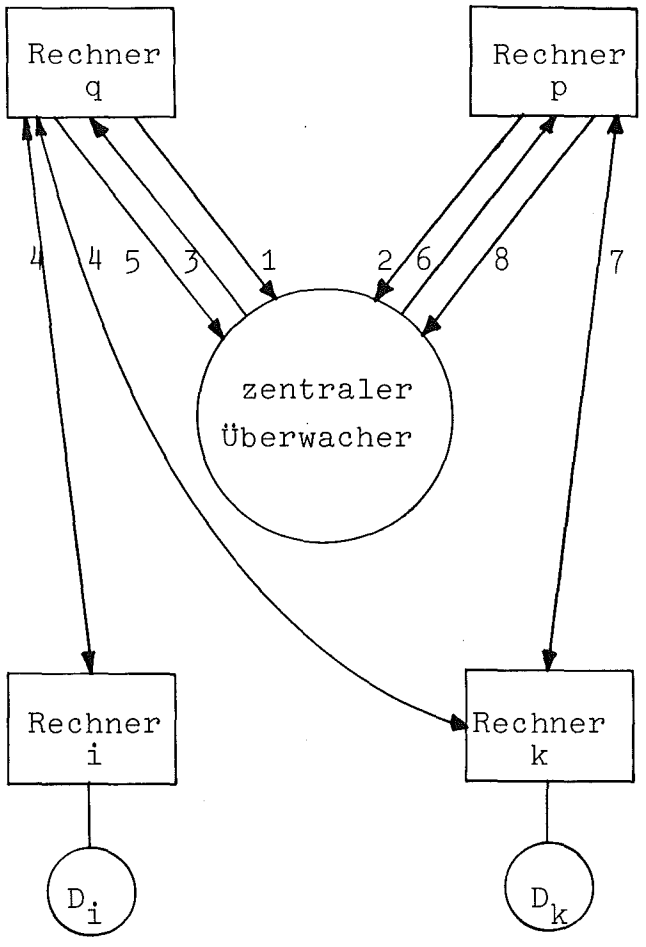


Abb. 1.3-1 Beispiel einer zentralen Überwachung kritischer Zugriffe. Dargestellt ist die Abwicklung zweier kritischer Zugriffsaufträge, die von den Rechnern p und q ausgehen. Der Auftrag von Rechner q möge als erster beim Überwacher eintreffen, wo die Bearbeitung nach FIFO erfolgen soll. Da beide Zugriffe die Datenbankkomponente D_k einschließen, ist eine Koordination notwendig,die zu der durch die Zahlen gekennzeichneten Reihenfolge des Nachrichtenaustauschs führt:

1 Anmeldung des Zugriffs auf D_i und D_k durch Rechner q
2 Anmeldung des Zugriffs auf D_k durch Rechner p
3 Zugriffsbewilligung für Rechner q
4 Zugriffsausführung durch Rechner q
5 Benachrichtigung des Überwachers über beendete Zugriffsausführung durch Rechner q
6 Zugriffsbewilligung für Rechner p
7 Zugriffsausführung durch Rechner p

Die Lösungen zu dem in der Literatur behandelten "Readers and Writers Problem" /9/,/21/ können auf das Problem der Zugriffskoordination bei Datenbanken im Falle einer zentralen Zugriffsüberwachung direkt übertragen werden, wenn man unkritische Zugriffe mit "Readers" und kritische Zugriffe mit "Writers" identifiziert.

Ein zentraler Überwachungsmechanismus für eine verteilte Datenbank, wie in Abb. 1.3-1 gezeigt, hat jedoch folgende wesentlichen Nachteile:

- Da sämtliche Zugriffe, auch die unkritischen, über den zentralen Koordinationsmechanismus geleitet werden müssen, entsteht ein Engpaß sowohl bezüglich der Zuverlässigkeit und Verfügbarkeit des gesamten Datenbanksystems als auch hinsichtlich der Effizienz der Abwicklung unkritischer Zugriffe.
- Ein zentral organisierter Überwachungsmechanismus würde von der Struktur her der sich immer mehr durchsetzenden dezentralen Betriebsorganisation von Rechnernetzen widersprechen.

Die Alternative zur zentralen Überwachung stellt eine dezentral organisierte Koordination der kritischen Zugriffe dar, die von unabhängigen, über das Rechnernetz verteilten, Überwacherprozessen getragen wird. An das dezentral aufgebaute Überwachersystem wird dabei die Forderung gestellt, daß der Ausfall einzelner Überwacherprozesse die Arbeitsweise des Restsystems nicht beeinträchtigt.

Die dezentrale Konzeption eines Überwachungssystems verbietet die Einrichtung zentraler Kommunikationsdatenbereiche und damit die Anwendung bekannter Synchronisationstechniken wie in /7/,/18/ und /20/ vorgeschlagen.

In der vorliegenden Arbeit wird untersucht

- auf welchen Methoden Verfahren aufbauen können, die für die dezentrale Überwachung und Koordination kritischer Zugriffe in Betracht kommen,
- welche alternativen Verfahren sich auf diesen Methoden aufbauen lassen,

- wie sich die möglichen Alternativen in bezug auf ihre Leistungsfähigkeit unterscheiden.

2. Koordination durch Kommunikation sequentieller Prozesse

2.1. Modell der Organisationsstruktur des Überwachungssystems

Ausgangsbasis für die Ermittlung geeigneter Methoden zur Koordination kritischer Zugriffe auf verteilte Datenbanken in Rechnernetzen sei das in Abb. 2.1-1 dargestellte Modell.

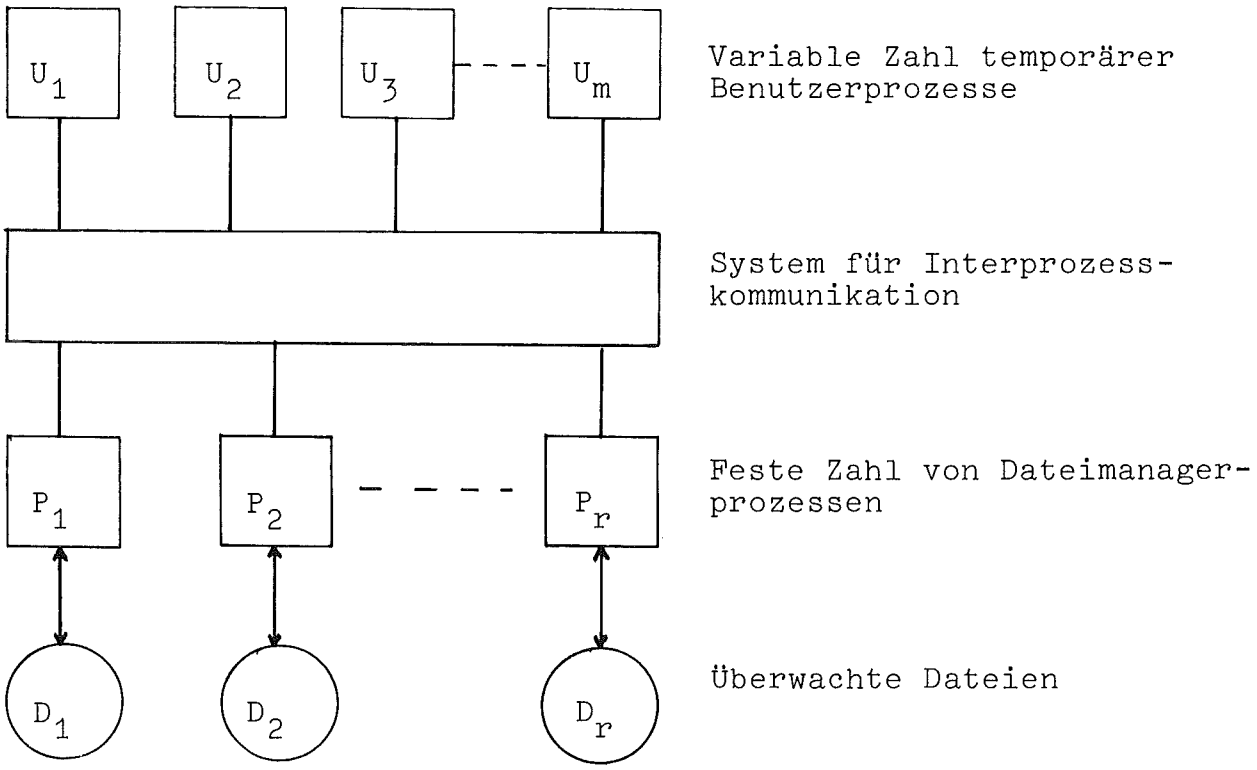


Abb. 2.1-1 Organisationsstruktur eines dezentralen Überwachungssystems

Jeder Knoten, d.h. jedes autonome Rechnersystem des Rechnernetzes verfüge über einen für die Kontrolle der Zugriffe zu den bei ihm lokalisierten Dateien verantwortlichen Prozeß, den wir als Dateimanager bezeichnen wollen. Dateimanager kann man sich als unabhängige Systemprozesse im Rahmen der Betriebsorganisation der beteiligten Rechner vorstellen.

Zugriffsanforderungen werden von Benutzerprozessen generiert. Um eine Beschränkung der Allgemeinheit zu vermeiden, sei ange-

nommen, daß jeder Knoten des Rechnernetzes als Ursprungsort für Zugriffsanforderungen (kritisch oder unkritisch) in Frage kommt, unabhängig davon, ob am Entstehungsort einer Anforderung selbst Dateien lokalisiert sind oder nicht. Jeder Dateimanager verfüge außerdem über die Fähigkeit, mit anderen Dateimanagern sowie mit Benutzerprozessen, die als "Produzenten" von Zugriffsanforderungen auftreten, zu kommunizieren.

Die Abwicklung eines Dateizugriffs geht nun so vonstatten, daß der die Zugriffsanforderung erzeugende Benutzerprozeß die Anforderung an das System der Dateimanager übermittelt, indem er mit den an der Zugriffsdurchführung beteiligten Dateimanagern direkt oder indirekt (d.h. über andere Dateimanager) kommuniziert.

Da bei unseren Betrachtungen die Koordination kritischer Zugriffe im Vordergrund steht, deren Durchführung die simultane exklusive Inanspruchnahme mehrerer Dateien in einem Rechnernetz bedingt, sei angenommen, daß für die Zahl r der an der Ausführung eines derartigen Zugriffs beteiligten Dateimanager stets $r \geq 2$ gilt. (Der Fall r=1 ist, wie in 1.3. erläutert, mit konventionellen Synchronisationsmethoden zu lösen.)

Die Arbeitsweise der Dateimanager sei als die sequentieller zyklischer Prozesse charakterisiert, die während eines Arbeitszyklus drei verschiedene Abschnitte durchlaufen (vgl. /14/,/18/):

- einen Testabschnitt
- einen kritischen Abschnitt
- einen "Rest"-Abschnitt

Ziel einer derartigen Aufgliederung der Prozeßabläufe der Dateimanager ist es, durch eine geeignete Form der Verständigung der Dateimanager untereinander während der Testabschnitte ein koordiniertes Vorgehen in den kritischen Abschnitten zu erreichen:

Alle Dateimanager sollen möglichst gleichzeitig in ihren kritischen Abschnitt zur Durchführung des exklusiven Zugriffs auf die ihnen zugeordnete Datei für dieselbe Zugriffsanforderung eintreten, damit die Gesamtdauer der Blockierung der betroffenen Daten-

bankkomponenten minimal gehalten wird.

Die Abwicklung eines kritischen Zugriffs mit den aus dem Vorangegangenen resultierenden drei Phasen

- Initialisierung des Zugriffs und Koordinierung der beteiligten Dateimanager
- parallele Durchführung der erforderlichen Einzelzugriffe
- Benachrichtigung der Dateimanager untereinander über die beendete Durchführung des Zugriffs (Desynchronisation)

ist in Abb. 2.1-2 dargestellt.

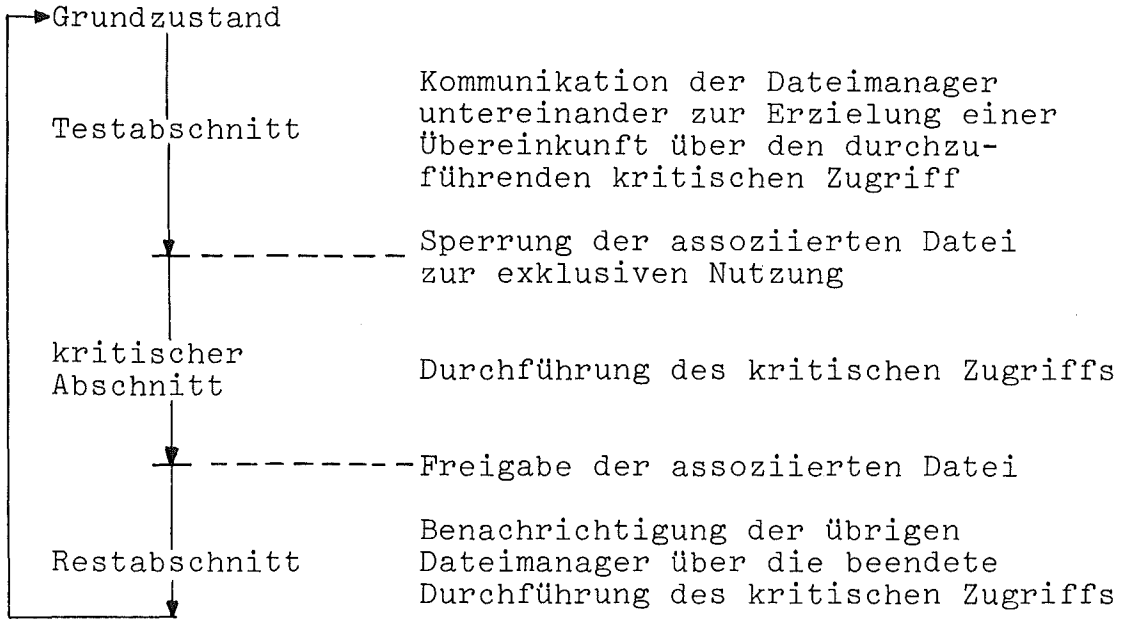


Abb. 2.1-2 Abwicklung kritischer Zugriffe durch Dateimanager

Die den hier entwickelten Modellvorstellungen zugrundeliegenden wesentlichen Voraussetzungen sind:

- eine auf Rechnernetze erweiterbare Möglichkeit der Interprozeßkommunikation ohne Verwendung eines zentralen, von allen Prozessen gemeinsam zu benutzenden Kommunikationsdatenbereiches

- einheitliche, für alle Dateimanager verbindliche Vereinbarungen zur blockierungsfreien dezentralen Abwicklung der Koordinierungsphase, d.h. der Testabschnitte der beteiligten Dateimanager.

Als erster Schritt zur Lösung des Problems der Koordinierung kritischer Zugriffe auf verteilte Datenbanken in Rechnernetzen ist daher zunächst die Untersuchung von Techniken notwendig, die die Kooperation von Prozessen in Rechnernetzen ermöglichen.

2.2. Elementarfunktionen der Interprozeßkommunikation

Unter Elementarfunktionen der Interprozeßkommunikation in DV-Systemen seien allgemein Funktionen verstanden, die es gestatten

- logische Übertragungswege für Nachrichten zwischen beliebigen Prozessen zu initialisieren,
- sowohl einseitige als auch wechselseitige Nachrichtenübertragungen zwischen Prozessen über die initialisierten Übertragungswege durchzuführen,
- den Abbau nicht mehr benötigter Interprozeßverbindungen vorzunehmen.

Eng verknüpft mit dieser Gruppe von Funktionen sind zwei weitere notwendige Elementarfunktionen mit einer in der Regel für den Benutzer (hier der kommunizierende Prozeß) nicht unmittelbar sichtbaren Wirkungsweise:

- Die Verwaltung des "virtuellen" Adreßraumes (auch Namensraum, siehe /48/), der für die Namensgebung oder Identifikation der kommunizierenden Prozesse bzw. ihrer Nachrichtenein- und -ausgänge (ports oder sockets im Englischen) zur Verfügung steht. Die Verwaltung dieses "virtuellen" Adreßraumes beinhaltet die Abbildung der gemäß getroffener Vereinbarung wählbaren Prozeßnamen auf den für die Interprozeßkommunikation verfügbaren realen Adreßraum (in einem Rechnersystem z.B. bestimmt durch die vorgegebene Wortlänge) und umgekehrt.

- Die Bereitstellung von "Transporteinrichtungen" für die Übertragung der Nachrichten vom Absender zum Empfänger, konkret also von Nachrichtenpuffern oder Rahmen (frames), in die die zu übermittelnden Nachrichten abgelegt bzw. eingebettet werden.

Zur Realisierung der Interprozeßkommunikation in einem System für Multiprogramming schlägt Hansen /20/ die Einführung folgender, im Systemnukleus zu realisierender Elementarfunktionen vor:

- sende Nachricht und warte auf Nachricht zur aktiven bzw. passiven Eröffnung (Initialisierung) eines Gesprächs zwischen Prozessen. Prozesse seien hier identisch mit den von Hansen als "internal processes" bezeichneten Einheiten. Als Parameter dieser Funktionen werden bei Beginn bzw. Beendigung der Ausführung der Namen des Absenders, die Adresse für die Ablage der Nachricht "innerhalb" des Prozesses sowie die Adresse des bereitgestellten Puffers übergeben.
- sende Antwort und warte auf Antwort zur aktiven bzw. passiven Weiterführung eines bereits eröffneten Dialogs unter Umgehung der FIFO-Warteschlangenabarbeitung der in der Nachrichtenwarteschlange auf ihre Abarbeitung durch warte auf Nachricht wartenden Nachrichten. Diese Funktionen nehmen Bezug auf die durch die Initialisierungsfunktion bereitgestellten Nachrichtenpuffer.

  Verwaltung der Nachrichtenpuffer und der Nachrichtenwarteschlangen obliegen ebenfalls dem Systemnukleus, der untersten Schicht des zugrundeliegenden hierarchisch organisierten Betriebssystems. Jedem Prozeß wird bereits bei seiner Initialisierung eine Nachrichtenwarteschlange (die natürlich zum Initialisierungszeitpunkt leer ist) zugeordnet.

Die Nachteile der von Hansen vorgeschlagenen Elementarfunktionen liegen darin, daß

- für die Verwaltung der vom System bereitgestellten Nachrichtenpuffer der Benutzer zuständig ist,

- die echte Dialogführung dadurch verhindert wird, daß nach jeder Ausführung der Funktion warte auf Antwort der Dialog mit sende Nachricht neu eröffnet werden muß. Dies kann zur ungewollten Unterbrechung des fortzusetzenden Dialogs durch ein anderes, neu eröffnetes Gespräch führen, da ein Prozeß nach Aufruf der Funktion sende Antwort mit warte auf Nachricht auf die Fortführung des Dialogs warten muß (vgl./20/).

Die vorhandenen Ansätze zur Ermöglichung der Interprozeßkommunikation in Rechnernetzen können als Versuch gewertet werden, das Schema Hansens auch auf die Kommunikation zwischen Prozessen auszudehnen, die nicht im gleichen Rechnersystem lokalisiert sind. /4/ /10/ /17/ /44/ /48/

Implementiert wurden Möglichkeiten dieser erweiterten Interprozeßkommunikation erstmals im Rahmen der Arbeiten am sogenannten ARPA-Netzwerk /4/ /10/. Die hier verfügbaren Elementarfunktionen unterscheiden sich von denen Hansens im wesentlichen durch folgende Eigenschaften:

- Ein einmal initialisierter Dialog bleibt bis zu seiner expliziten Beendigung durch eine zusätzliche Elementarfunktion aufrechterhalten. Zu jedem Dialog gehören zwei komplementäre gerichtete logische Verbindungen (Kanäle), von denen jede die Information in jeweils nur eine Richtung überträgt.
- Jeder Prozeß kann simultan mehrere gerichtete logische Verbindungen zu anderen Prozessen unterhalten. Eine logische Verbindung wird dabei identifiziert durch die Namen bzw. Adressen der beteiligten Aus- und Eingänge für den Nachrichtenfluß von und zu den betroffenen Prozessen.
- Die Pufferverwaltung obliegt dem System.

Ein anderes Rechnerverbundsystem, das DCS-System /17/,in dessen Konzeption ebenfalls die Möglichkeit der erweiterten Interprozeßkommunikation enthalten ist, sei hier deshalb erwähnt, weil es eine interessante Eigenschaft besitzt: Sie unterstützt die Kommunikation zwischen einzelnen Prozessen als Nachrichten-

sender auf der einen und ganzen Prozeßgruppierungen als Empfänger auf der anderen Seite.

Wie die aufgeführten Beispiele zeigen, ist eine Interprozeßkommunikation in Rechnernetzen ebenso wie in einzelnen Rechnersystemen, die den zu Beginn des Kapitels aufgestellten Forderungen genügt, durchaus realisierbar. Es sei jedoch auf einen wesentlichen Unterschied zwischen der Interprozeßkommunikation in Rechnernetzen und der in isolierten Rechnersystemen hingewiesen:

Während die Dauer einer Nachrichtenübertragung zwischen verschiedenen Prozessen innerhalb ein und desselben Rechnersystems nahezu ausschließlich durch die Geschwindigkeit der Verwaltung von Puffer und Warteschlangen durch den Systemnukleus bestimmt wird, tragen in Rechnernetzen die niedrigen Übertragungsgeschwindigkeiten der für die Nachrichtenübertragung verfügbaren Kommunikationssysteme oft zu Zeitverzögerungen bei, die um Größenordnungen höher lieger können als die für nukleus-interne Kommunikationsoperationen benötigte Zeit.

Inwiefern sich diese Zeitverzögerungen nachteilig auf Mechanismen zur Koordinierung der Operation der Dateimanager in dem in 2.1. präsentierten Modell auswirken werden, wird Gegenstand experimenteller Untersuchungen dieser Arbeit sein.

Aufgrund der in diesem Kapitel vorgenommenen Untersuchungen in bezug auf die Möglichkeiten einer Erweiterung der Elementarfunktionen der Interprozeßkommunikation auch für die Kommunikation zwischen Prozessen in Rechnernetzen wollen wir nun unsere in Kapitel 2.1. entwickelten Vorstellungen für das Modell eines Überwachungssystems wie folgt präzisieren:

1. Bei Initialisierung der Dateimanager erfolgt eine wechselseitige Initialisierung logischer Übertragungswege einschließlich der benötigten Nachrichtenpuffer zwischen je zwei Dateimanagern derart, daß jeder dieser Prozesse jederzeit einen Dialog mit jedem anderen Dateimanager starten kann. Dies soll (vgl. z.B. /10/) durch Elementarfunktionen der Form

initialisiere Verbindung (lokaler Prozeßausgang, fremder Prozeßeingang)

und

bereit für Verbindung (lokaler Prozeßeingang, Ereignisvariable)

entsprechend einer aktiven bzw. passiven Eröffnung einer Verbindung, ermöglicht werden. (Die Funktionen sind jeweils durch Unterstreichung, die Funktionsparameter durch Einschließung in Klammern kenntlich gemacht.) Die gleichen Funktionen dienen auch der Eröffnung von Verbindungen zu Benutzerprozessen während des Ablaufs eines Dateimanager-Prozesses. Der Parameter "Ereignisvariable" dient als Semavariable der Benachrichtigung des Prozesses über die Eröffnung der Verbindung oder den Eingang einer Nachricht.

2. Senden und Empfangen von Nachrichten während eines Dialogs erfolgt durch Elementarfunktionen der Form

   sende Nachricht (lokaler Prozeßausgang, Nachrichtenadresse)
   bringe Nachricht (lokaler Prozeßeingang, Nachrichtenadresse)

   Bei Ausführung der Funktion sende Nachricht müssen die aktuellen Parameterwerte vor der Ausführung der Funktion gesetzt sein; bei bringe Nachricht werden die Parameterwerte durch die Ausführung zugewiesen.

3. Mit dem Aufruf der Kommunikationsfunktionen bereit für Verbindung und bringe Nachricht durch einen Prozeß soll kein Statuswechsel dieses Prozesses vom aktiven in den wartenden Zustand verbunden sein. Vielmehr ist durch die Spezifikation einer Ereignisvariablen als Aufrufparameter bei der passiven Initialisierung einer Verbindung die Synchronisation an den erforderlichen Stellen innerhalb des Prozesses mittels einer Synchronisationsfunktion

   warte (Ereignisvariable)

   möglich. "Ereignisvariable" kann hierbei als "genereller Semaphor" im Sinne Dijkstras /14/, der dem Prozeß als pri-

vater Semaphor zugeordnet ist, aufgefaßt werden. Damit können z.B. sämtliche Nachrichten, die über parallel bestehende logische Verbindungen bei einem Prozeß eintreffen, durch nur eine einzige Ereignisvariable erfaßt werden, vorausgesetzt, bei der Initialisierung der Verbindungen durch bereit für Verbindung wurde die gleiche Ereignisvariable spezifiziert.

4. Kommunikationsverbindungen zu anderen Prozessen können mit der Elementarfunktion

   beende Verbindung (lokaler Prozeßausgang)

   beendet werden.

Die Anwendung der Kommunikationsfunktionen sei anhand eines Beispiels demonstriert:

Sei $\{P_1,P_2,\ldots,P_r\}$ ein System von r Dateimanagern; die Anforderung kritischer Zugriffe zu der von $\{P_1,P_2,\ldots,P_r\}$ überwachten Datenbank erfolge durch Benutzerprozesse $U_i \varepsilon U$, der Menge der authorisierten Benutzerprozesse.

Der Ablauf eines Dateimanagerprozesses P_i kann nun wie folgt beschrieben werden:

```
        PI: begin
        Ereignisvariable NACHRICHTENANKUNFT;
        text NACHRICHT; integer EINGANGSNR, MAXLAENGE;
        boolean DIALOGENDE, KOMMUNIKATION;...
        MAXLAENGE:=...;
INIT:   initialisiere Verbindung (AUS_I1,EIN_1I);
        bereit für Verbindung (EIN_I1,NACHRICHTENANKUNFT);
          :
        initialisiere Verbindung (AUS_IR,EIN_RI);
        bereit für Verbindung (EIN_IR,NACHRICHTENANKUNFT);
        bereit für Verbindung (EIN_IB,NACHRICHTENANKUNFT);
```

```
ZYKLUS:
TEST:    warte(NACHRICHTENANKUNFT);
         NACHRICHT:-blanks(MAXLAENGE);
         bringe Nachricht(EINGANGSNR,NACHRICHT);
           :
         <Nachrichtenanalyse>;
KOMM:    if KOMMUNIKATION then begin
         sende Nachricht(AUS_Ix,NACHRICHT);
           :
         go to KOMM
         end;
         if not DIALOGENDE then go to TEST;

KRIT:
           :
         <kritischer Abschnitt>;
           :

REST:    sende Nachricht(AUS_IB,NACHRICHT);
           :
         <Desynchronisation>;
           :
         go to ZYKLUS;
         end von PI;
```

Das Beispiel zeigt den Aufbau eines Dateimanagers: Nach Durchlaufen des Initialisierungsabschnitts (INIT), in dem der Aufbau der logischen Verbindungen zu allen anderen Dateimanagern erfolgt, und eine potentielle Verbindung zu Benutzerprozessen (über EIN_IB) vorbereitet wird, tritt der Prozeß in seinen eigentlichen Arbeitszyklus ein, der sich in Testabschnitt (TEST),

kritischen Abschnitt (KRIT) und Restabschnitt (REST) untergliedert (vgl. Kapitel 2.1.).

Der Testabschnitt beginnt mit der Entgegennahme von Nachrichten (Aufträgen, vgl. /47/), der eine Nachrichtenanalyse folgt. Aufgrund des Ergebnisses der Nachrichtenanalyse (angezeigt durch die Variablen KOMMUNIKATION und DIALOGENDE) wird entschieden, ob weitere Nachrichten ausgetauscht werden müssen, oder ob die angestrebte Koordination, die die Ausführung des kritischen Zugriffs möglich macht, bereits erreicht ist. Ist letzteres der Fall, so kann der Eintritt in den kritischen Abschnitt erfolgen; der Restabschnitt dient der Verständigung der beteiligten Prozesse über die erfolgte Ausführung des Zugriffs.

blanks ist eine Hilfsfunktion zur Normierung des Speicherbereiches für die Aufnahme von Nachrichten.

Die Namen AUS_xy und EIN_xy kennzeichnen die logischen Anschlußstellen der Informationskanäle an die Prozesse (Prozeß-Ein- und Ausgänge). Die Festlegung der zugehörigen Identifikation erfordert eine Vereinbarung, die die eindeutige Kennzeichnung der Informationskanäle zwischen den beteiligten Prozessen gewährleistet.

Die für den Aufbau des Überwachungssystems wesentlichste Vereinbarung ist die Absprache über die den Dialog zwischen den Dateimanagern steuernde Nachrichtenanalyse und die daraus resultierenden Prozeßaktionen. Derartige Vereinbarungen über Kommunikationsabläufe sind Gegenstand von Kommunikationsprotokollen, die im folgenden behandelt werden sollen.

2.3. Kommunikationsprotokolle

Unter Kommunikationsprotokollen wollen wir die unter Kommunikationspartnern vereinbarten Regeln verstehen, die dazu dienen, den gegenseitigen Informationsaustausch aufeinander abzustimmen. Ziel einer Festlegung derartiger Regeln ist die Koordination der Einzelaktionen der beteiligten Gesprächspartner, in unserem Falle der Dateimanager-Prozesse.

Mit dieser Definition des Kommunikationsprotokolles befinden wir uns in Übereinstimmung mit der Mehrzahl der in Veröffentlichungen zu findenden Definitionen (vgl. /4/,/10/,/27/,/29/, /42/,/48/); als Synonyme für Kommunikationsprotokoll findet man gelegentlich auch die Bezeichnungen Kommunikationsprozedur oder -disziplin.

Wir wollen versuchen, mit dem Begriff Protokoll nicht nur die bilaterale Kommunikation (Dialog im eigentlichen Sinne) zu erfassen, sondern auch, durch geeignete formale Darstellungen, die multilaterale Kommunikation zwischen einer beliebigen Zahl von Gesprächspartnern zu beschreiben.

Da der Kern der zu entwickelnden Koordinationsverfahren ein spezielles (problemorientiertes) Protokoll der Interprozeßkommunikation darstellt, ist es erforderlich, Kommunikationsprotokolle näher zu untersuchen.

Zur Realisierung der Interprozeßkommunikation in Rechnernetzen bedarf es einer Hierarchie bilateraler Kommunikationsprotokolle, in der, in Analogie zu schichtenförmig organisierten, auf der Rechnerhardware als unterster Schicht aufbauenden Betriebssystemen /15/, die für eine Kommunikationsebene bereitgestellten elementaren Kommunikationsfunktionen durch Protokolle der darunter liegenden Schicht implementiert werden /10/. Abb. 2.3-1 zeigt das Schema einer denkbaren Protokollhierarchie:

Die der Hardware am nächsten liegende Schicht regelt den Ablauf innerhalb realer (physikalischer) Kommunikationskanäle mittels Hardwarefunktionen (Steuerzeichen) zur Kommunikationseröffnung, -beendigung, Übertragung von Nachrichtenblöcken und Fehlerkorrektur. Gesprächspartner auf dieser Ebene sind Vermittlungseinrichtungen (Vermittlungsrechner).

Auf dem Kanalprotokoll baut die Rechner-Rechnerkommunikation auf; sie dient der Bereitstellung und Bedienung von Kommunikationswegen durch Verfügbarmachung und Zusammenschaltung der benötigten physikalischen Kanäle. Durch das Protokoll auf dieser

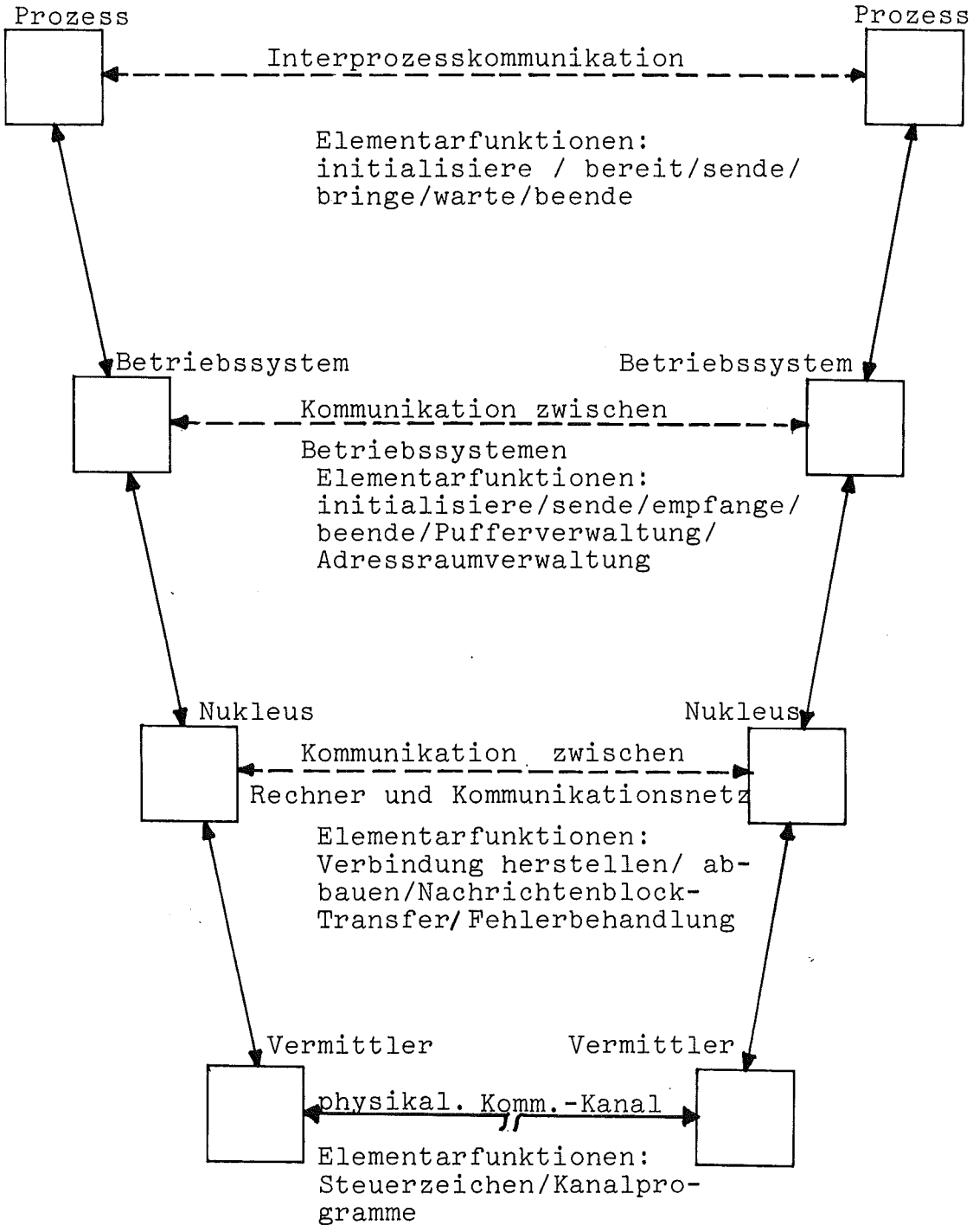


Abb. 2.3-1 Protokollhierarchie und Kommunikationsfunktionen

Ebene werden im Systemnukleus der beteiligten Rechner lokalisierte Funktionen bereitgestellt, die auf der Betriebssystemebene (Intersystemkommunikation) den Aufbau der in 2.2. beschriebenen Elementarfunktionen zur Interprozeßkommunikation ermöglichen.

Um ein Hilfsmittel zur Erstellung der benötigten Protokolle für die Inter-Dateimanager-Kommunikation zu erlangen, wollen wir nun die Möglichkeiten einer formalen Beschreibung von Kommunikationsprotokollen untersuchen.

Eine detailliertere Betrachtung von Kommunikationsprotokollen auf verschiedenen Ebenen einer Hierarchie läßt zunächst folgende Gemeinsamkeiten erkennen:

- die beim Informationsaustausch übermittelten Nachrichten bzw. Nachrichtenblöcke beinhalten in der Regel zwei verschiedene Informationsarten, die stets vorhandene Nachrichtenhülle und den nicht notwendig vorhandenen Nachrichtentext. Nachrichten ohne Text wollen wir als Kontrollnachrichten bezeichnen.
- Die Nachrichtenhülle besteht aus Kontrollinformationen, die der Nachrichtenempfänger interpretiert; gegenüber dem Nachrichtentext verhalten sich die Kommunikationspartner neutral.
- Kontrollinformationen in der Nachrichtenhülle können zu Zustandsänderungen beim Nachrichtenempfänger und einer von der Art des Zustandsüberganges abhängigen Generierung von Kontrollinformation für den Nachrichtensender führen.
- In Abhängigkeit vom augenblicklichen Zustand und der Art der empfangenen Kontrollinformation kann ein Nachrichtenempfänger in Wechselwirkung mit seiner Umwelt treten und gewisse Aktionen auf dieser Umwelt bewirken.

Die aufgeführten charakteristischen Eigenschaften von Kommunikationsprotokollen lassen erkennen, daß der Ablauf der Interkommunikation zwischen Gesprächspartnern durch die, aus den empfangenen Kontrollinformationen (Nachrichtenhüllen) resultie-

renden, Zustandsänderungen und die damit verbundenen Reaktionen (z.B. Aussenden von Kontrollinformationen) bestimmt wird.

Wir wollen im folgenden aus Gründen der Übersichtlichkeit annehmen, daß die Menge unterschiedlicher Nachrichtenhüllen, die zwischen den Kommunikationspartnern eines betrachteten Systems ausgetauscht werden, endlich ist; die Betrachtung realer Systeme berechtigt zu dieser Annahme. Ebenso sei die Menge der Zustände endlich, die die Gesprächspartner einnehmen können.

Sei Σ_i das "Alphabet" der von einem Gesprächspartner (auch: Interlokutor, vgl. /29/) P_i des Systems $\{P_1,\ldots,P_n\}$ als Kontrollinformationen akzeptierten Nachrichtenhüllen, S_i die Menge der Zustände von P_i und W_i die Menge der Aktionen von P_i auf seiner Umgebung.

Das Kommunikationsprotokoll für das System $\{P_1,\ldots,P_n\}$ ist dann gegeben durch die gleichmächtige Menge der Konventionen

$$V = \{V_1,\ldots,V_n\}$$

$$\text{mit} \quad V_i = (\tau_i,\alpha_i) \qquad i=1(1)n$$

wobei die Abbildungen

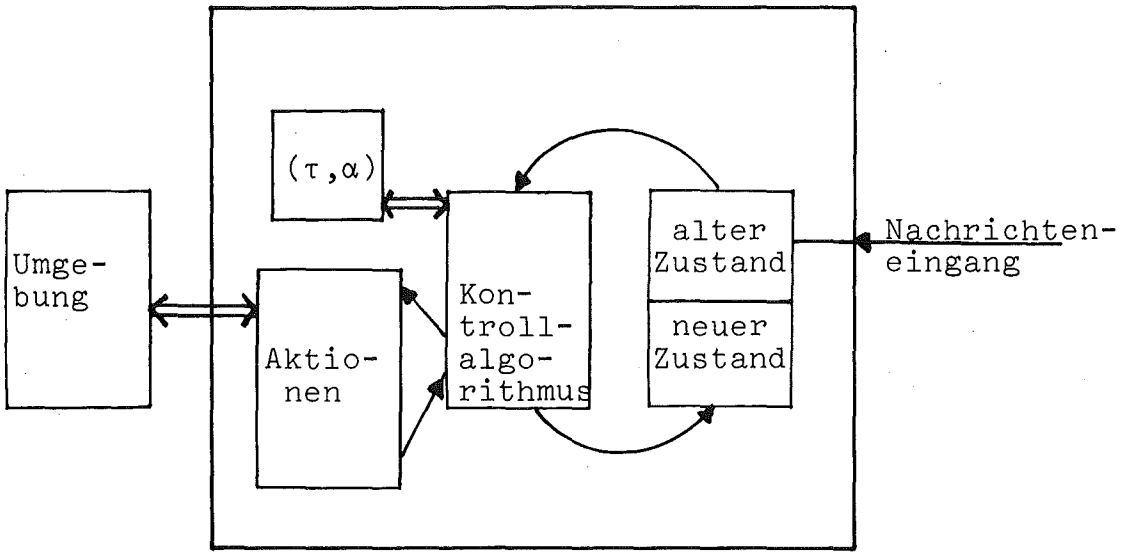
$$\tau_i: S_i \times \Sigma_i \to S_i \qquad i=1(1)n \tag{2.3-1}$$

die durch den Empfang von Kontrollinformationen bedingten Zustandsübergänge beschreiben, die im Falle des deterministischen Verhaltens der Gesprächspartner eindeutig sind und die Abbildungen

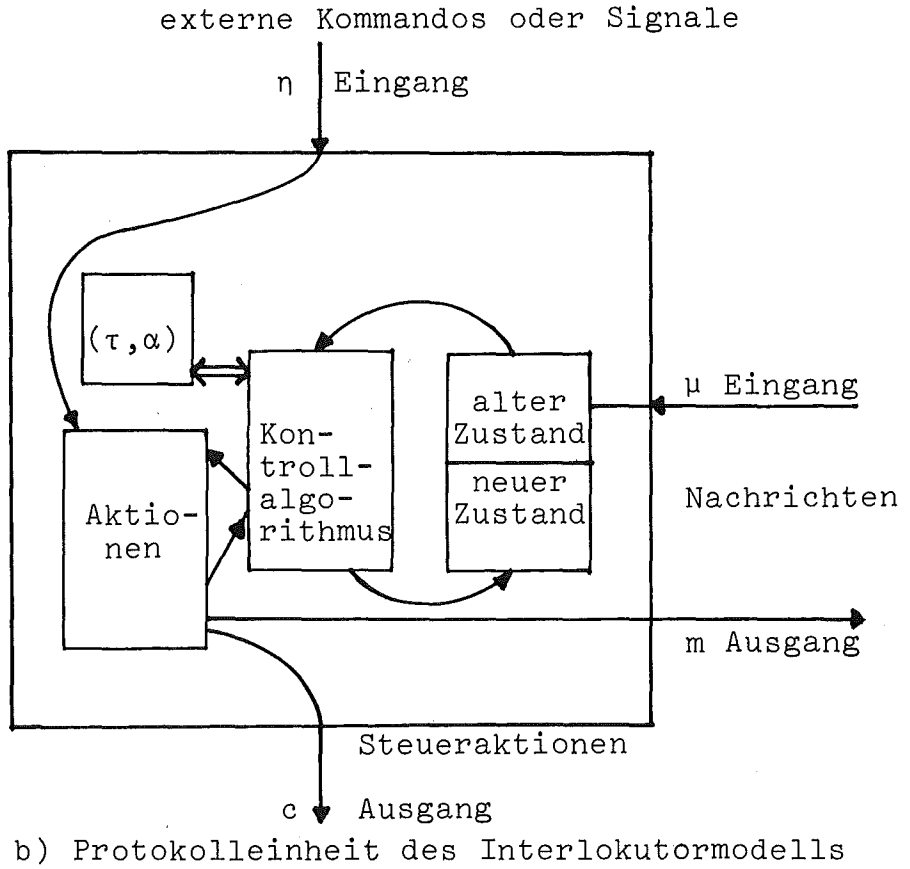
$$\alpha_i: S_i \times \Sigma_i \to W_i \qquad i=1(1)n \tag{2.3-2}$$

die durch den Nachrichtenempfang bewirkten Aktionen der Gesprächspartner.

Das Kommunikationsprotokoll entspricht damit der syntaktischen Festlegung der Kommunikationskonventionen innerhalb eines Systems von Gesprächspartnern /22/. Analog zur Grammatik einer Sprache legt das Protokoll alle gültigen Kommunikationssequenzen fest.



a) Modell von Hoffmann



b) Protokolleinheit des Interlokutormodells von Le Moli

Abb. 2.3-2 Modelle eines Gesprächsparterns zur formalen Beschreibung von Kommunikationsprotokollen

Während die oben skizzierte, von Hoffmann /22/ vorgeschlagene formale Beschreibung des Kommunikationsprotokolls die Erzeugung von Kontrollinformationen nur implizit in 2.3-2 erfaßt, werden in dem von Le Moli, Mezzalira und Schreiber entwickelten Modell (vgl. /29/,/30/) eines "Interlokutors" die durch die Semantik der in den Nachrichtenhüllen übertragenen Kontrollinformationen bewirkten unterschiedlichen Aktionen explizit dargelegt (siehe Abb. 2.3-2). Dies geschieht durch zusätzliche Einführung eines Alphabets externer Steuerkommandos, eines Nachrichtenausgabealphabets und einer Menge von Steueraktionen. Wir kommen damit zu einer Beschreibung der Kommunikationspartner durch das 7-Tupel

(2.3-3) $P = (I_\mu, I_\eta, O_m, O_c, S, M, N)$

wobei

I_μ = Alphabet der Eingabe-Kontrollnachrichten

I_η = Alphabet der externen Kommandos

O_m = Alphabet der Ausgabe-Kontrollnachrichten

O_c = Menge der Steueraktionen

S = Zustandsmenge

$M(I_\eta)$: $SxI_\mu \rightarrow S$ Menge der Zustandsübergangsfunktionen

$N(I_\eta)$: $SxI_\mu \rightarrow O_m x O_c$ Menge der Ausgabefunktionen

Das Argument I_η der Abbildungen deutet an, daß die durch den Empfang einer Nachrichtenhülle ausgelöste Abbildung vom an Eingang η anliegenden Steuerkommando und damit von den Elementen aus I_η explizit abhängig ist.

Der im Modell von Le Moli eingeführte Eingang für externe Steuerkommandos stellt eine Eingriffsmöglichkeit von außen dar, die den in der Realität vorhandenen Möglichkeiten des Rücksetzens (in einen definierten Ausgangszustand) oder des Startens von Kommunikationsoperationen Rechnung trägt. Im Falle

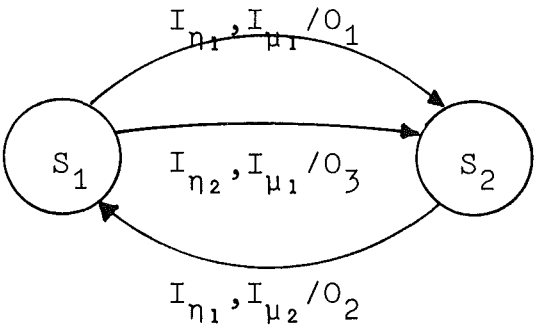
unseres Systems kommunizierender Dateimanager entspräche etwa die Benachrichtigung und damit Aktivierung eines Dateimanagers durch einen Benutzerprozeß zwecks Einleitung eines kritischen Zugriffs der Eingabe einer Steuernachricht über diesen speziellen Eingang.

Der ebenfalls zusätzlich (gegenüber dem Modell von Hoffmann) eingeführte Ausgang c, der einem "Interlokutor" die Initialisierung von Steueraktionen ermöglicht, ist notwendig, um die in realen Systemen möglichen Aktivierungen externer Prozesse oder von Peripheriegeräten berücksichtigen zu können. In Bezug auf das Dateimanagersystem kann ein Signal an diesem Steuerausgang als Aktivierung des eigentlichen Zugriffs interpretiert werden.

Der Vollständigkeit halber sei erwähnt, daß im eigentlichen Interlokutor-Modell von Le Moli die Textbehandlung ebenfalls berücksichtigt wird. Diese besteht darin, daß der Text aus der am Nachrichteneingang angelangten Nachricht entnommen und zu einem speziellen Textausgang durchgeschleust wird, so daß am Eingang nur die Nachrichtenhülle verbleibt. In analoger Weise werden die am Texteingang ankommenden Informationen als Text in eine für die Ausgabe vorbereitete Nachrichtenhülle eingefügt. Da sich der Interlokutor gegenüber dem Nachrichtentext neutral verhält, d.h. keine Zustandsänderung durch den Texttransport erfährt, wollen wir auf eine Einbeziehung der Textbehandlung in die hier entwickelte formale Beschreibung von Kommunikationsprotokollen verzichten.

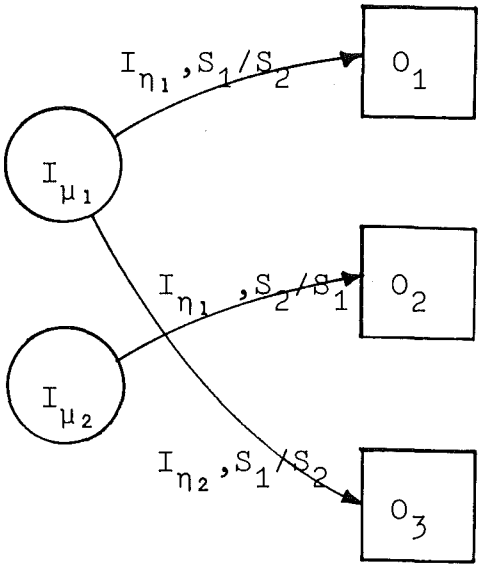
In der für die Beschreibung des Verhaltens sequentieller Maschinen üblichen Form lassen sich für durch (2.3-3) beschriebene Systeme Tabellen für Zustandsübergänge und Ein/Ausgabe-Beziehungen angeben, derart, daß für jedes Element $I_{\eta i} \varepsilon I_{\eta}$ eine entsprechende Tabelle erstellt wird, wie an einem Beispiel in Abb. 2.3-3 demonstriert. Elemente des Ausgabealphabets sind dabei Paare

Zustandsgraph



(a)

Nachrichtengraph

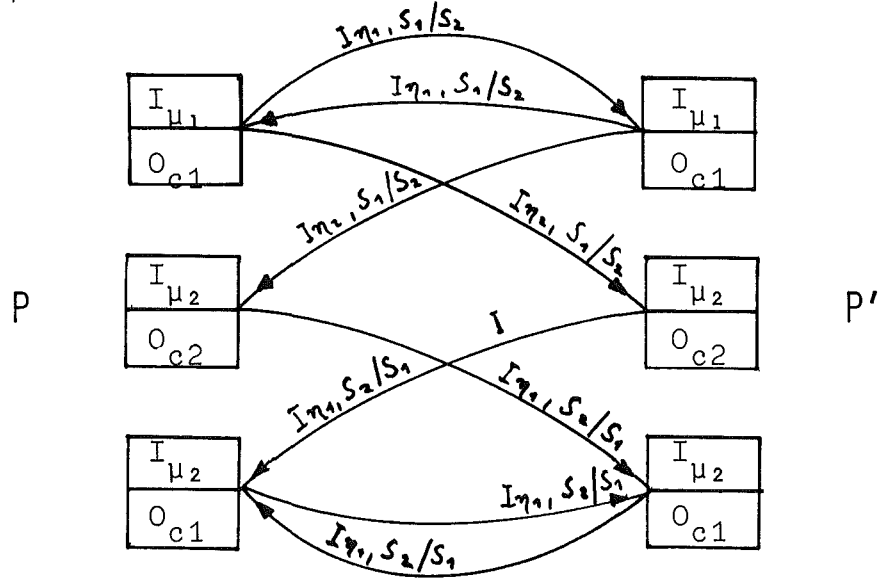


Eingabenachricht Ausgabenachricht

(b)

$O = \{(O_{m1}, O_{c1}), (O_{m2}, O_{c1}), (O_{m2}, O_{c2})\}$

$I_\mu = O_m$



(c) kombinierter Nachrichtengraph zweier Systeme P und P' , deren Zustandsgraph dem unter (a) entspricht

Abb. 2.3-3 Darstellung des Verhaltens von Kommunikationspart= nern durch Zustands- und Nachrichtengraphen

$$(O_{mi}, O_{ck}) \in O_m \times O_c$$

mit i und k als Elementen der Indexmengen von O_m und O_c.

Eine der Zustandsübergangstabelle äquivalente Darstellung ist der Zustandsgraph, der exakt das Verhalten eines Interlokutors beschreibt, und damit, bei Kenntnis der Gesamtheit der Zustandsgraphen der Gesprächspartner in einem System, einer vollständigen Beschreibung des Kommunikationsprotokolls dienen kann.

Wie in /30/ vorgeschlagen, bietet jedoch der aus der transponierten Zustandsübergangs- und Ein/Ausgabe-Tabelle (E/A-Tabelle) abgeleitete Nachrichtengraph die Möglichkeit, die Wechselwirkung zwischen Kommunikationspartnern in einer gemeinsamen Darstellung als Graph direkt zu erfassen. Dies ist in Abb. 2.3-3 für ein System

$$P = (I_\mu, I_\eta, O_m, O_c, S, M, N)$$

mit
$$I_\mu = \{I_{\mu 1}, I_{\mu 2}\}$$
$$I_\eta = \{I_{\eta 1}, I_{\eta 2}\}$$
$$S = \{S_1, S_2\}$$
$$O_m = \{O_{m1}, O_{m2}\}$$
$$O_c = \{O_{c1}, O_{c2}\}$$
$$O = \{O_1, O_2, O_3\} = \{(O_{m1}, O_{c1}), (O_{m2}, O_{c1}), (O_{m2}, O_{c2})\}$$

und den durch die Zustandsübergangs- und E/A-Tabelle Tab. 2.3-1

		$I_{\eta 1}$		$I_{\eta 2}$	
S \ I_μ		$I_{\mu 1}$	$I_{\mu 2}$	$I_{\mu 1}$	$I_{\mu 2}$
S_1		O_1/S_2		O_3/S_2	
S_2			O_2/S_1		

Tabelle 2.3-1 Beispiel einer Zustandsübergangs- und E/A-Tabelle eines Interlokutors

definierten Funktionen M und N demonstriert. Die zur Markierung der Kanten von Zustandsgraph bzw. Nachrichtengraph verwendete Notation der Form

Bedingung 1, Bedingung 2/Aktion

ist dabei zu interpretieren als eine von der Protokolleinheit eines Interlokutors ausgeführte bedingte Anweisung der Form

IF Bedingung 1 AND Bedingung 2 THEN Aktion

(Eine darüber hinaus gehende Verallgemeinerung wäre durch die Ausführung bedingter Anweisungen der Form

IF Liste v. Bedingungen THEN Aktion

gegeben.)

Im Nachrichtengraphen wurden die Übergänge zwischen Eingangsnachrichten und Ausgangsnachrichten als Funktion des anliegenden Steuerkommandos und des augenblicklichen Zustands erfaßt. Wie von Mezzalira und Schreiber /30/ gezeigt, liefert die Kombination von Zustands- und Nachrichtengraph auch unter Weglassung der erzeugten Ausgabenachrichten bzw. erreichten Endzustände (reduzierte Form) die vollständige Beschreibung eines Interlokutors. Für zwei Gesprächspartner P und P', für deren Eingangsalphabete bzw. Ausgangsalphabete gilt

(2.3-4) $\qquad I_\mu \supseteq O'_m \qquad I'_\mu \supseteq O_m$

kann, wie an einem Beispiel in Abb. 2.3-3c gezeigt, ein kombinierter Nachrichtengraph aus den individuellen Nachrichtengraphen erstellt werden. Den kombinierten Nachrichtengraphen gewinnt man, in dem man mit den von den einzelnen Gesprächspartnern erzeugten Kombinationen von Ausgabe-Kontrollnachrichten und Steueraktionen die Knoten eines Graphen markiert. Im Beispiel in Abb. 2.3-3c enthalten die Knoten links die von P erzeugten Kombinationen von Ausgabe-Kontrollnachrichten und Steueraktionen (gemäß Tabelle 2.3-1); entsprechend sind die rechts liegenden Knoten P' zuzuordnen. Da nach (2.3-4) die Ausgabenachrichten eines Gesprächspartners im Alphabet der Eingabe-

Kontrollnachrichten des anderen Gesprächspartners enthalten sind, können die Knoten des kombinierten Nachrichtengraphen mit Kanten gemäß den Vorschriften des einfachen Nachrichtengraphen, für das Beispiel in Abb. 2.3-3b dargestellt, verbunden werden. Die generierten Steueraktionen werden dabei unberücksichtigt gelassen.

Zulässige Sequenzen von Kontrollnachrichten und Steueraktionen können nun dem kombinierten Nachrichtengraphen direkt entnommen werden: Ausgehend von der an einem Gesprächspartner empfangenen Kontrollnachricht folgt man, unter Berücksichtigung der Zustände und anliegenden Steuerkommandos, einem zusammenhängenden Kantenzug.

Die Erstellung eines kombinierten Nachrichtengraphen, der als generatives Schema zur Ermittlung aller zulässigen Kommunikationssequenzen aufgefaßt werden kann, bietet sich als Verfahren zur globalen Beschreibung der Wechselwirkungen innerhalb eines Systems kommunizierender Prozesse in einer einzigen Darstellung an und kann vorteilhaft beim Aufbau spezifischer Kommunikationsprotokolle eingesetzt werden.

Diesem Vorteil des Verfahrens stehen einige Nachteile gegenüber, die hier kurz erläutert werden sollen:

- die Erweiterung zur Beschreibung der kommunikativen Wechselwirkungen von mehr als zwei Interlokutoren führt zu unübersichtlichen Darstellungen (es wäre die Einführung zusätzlicher Nachrichteneingänge bzw. -ausgänge notwendig sowie die Berücksichtigung sämtlicher Kombinationen von Eingabe- und Ausgabeinformationen).
- Modifikationen sind notwendig, wenn Systeme beschrieben werden sollen, bei denen kein alternierender Austausch von Kontrollinformationen über die Nachrichteneingänge bzw. -ausgänge stattfindet.
- Ein deterministisches Verhalten der Interlokutoren wird vorausgesetzt.

Projiziert auf die ins Auge gefaßte Anwendung stellt die erste Einschränkung kein Hindernis bei der Beschreibung von Dateimanagersystemen dar, da der Nachweis für die Zulässigkeit der Verallgemeinerung von für Zweiersysteme entwickelten Synchronisationsverfahren erbracht werden kann (vgl. Kap. 3.2.). Einschränkung zwei läßt sich umgehen durch Einführung von "leeren" Nachrichten. Da wir ein deterministisches Verhalten der Dateimanagerprozesse bereits voraussetzten, stellt Einschränkung drei keinen Nachteil bei der Behandlung unseres Systems dar.

3. Kommunikationsprotokolle zur Koordination kritischer Zugriffe

3.1. Formale Beschreibung von Zwei-Dateimanager-Protokollen

Mit der im vorangegangenen Kapitel dargelegten Methode der formalen Beschreibung von vereinbarten Wechselwirkungen zwischen kommunizierenden Systemen besitzen wir nun ein Instrument zur Entwicklung von problemorientierten Kommunikationsprotokollen, das wir zur Lösung der gegebenen Problemstellung einsetzen wollen.

Dazu sei zunächst ein System zweier Dateimanager betrachtet. Das System besitze die in 2.1. erläuterte Struktur; die die Dateimanager verkörpernden Prozesse sollen über die in Kapitel 2.2. zusammengestellten Elementarfunktionen miteinander und mit der Umwelt kommunizieren können. Initialisierungs- bzw. Terminierungsphasen der Dateimanager seien hier außer acht gelassen; wir wollen lediglich das operative Verhalten während der Arbeitszyklen untersuchen. Die Dateimanager werden als Prozesse betrachtet, die sich stets in einem der Zustände aktiv, rechenwillig oder wartend befinden - vgl. /36/.

Abbildung 3.1-1 zeigt eine schematische Darstellung des betrachteten Zwei-Dateimanager-Systems. Die Aktivierung des Systems erfolgt durch entsprechende Aufträge der Benutzerprozesse, die temporär irgendwo im Rechnernetz existieren mögen. Gelangt der Auftrag zur Durchführung eines kritischen Zugriffs bei einem der Dateimanager an, so führt dies zur Auslösung eines Steuersignals am Eingang η der zugehörigen Protokolleinheit (vgl. Abb. 2.3-2b in Kapitel 2.3.). Wird durch die nun einsetzende Kommunikation der beteiligten Dateimanager eine Übereinkunft in bezug auf die Durchführung des geforderten kritischen Zugriffs erzielt, so erfolgt die Ausgabe eines Steuersignals an Ausgang c der Protokolleinheit, welches als Startsignal für den E/A-Monitor des Rechnersystems (IOCS) zur Durchführung der notwendigen E/A-Operationen interpretiert werden kann.

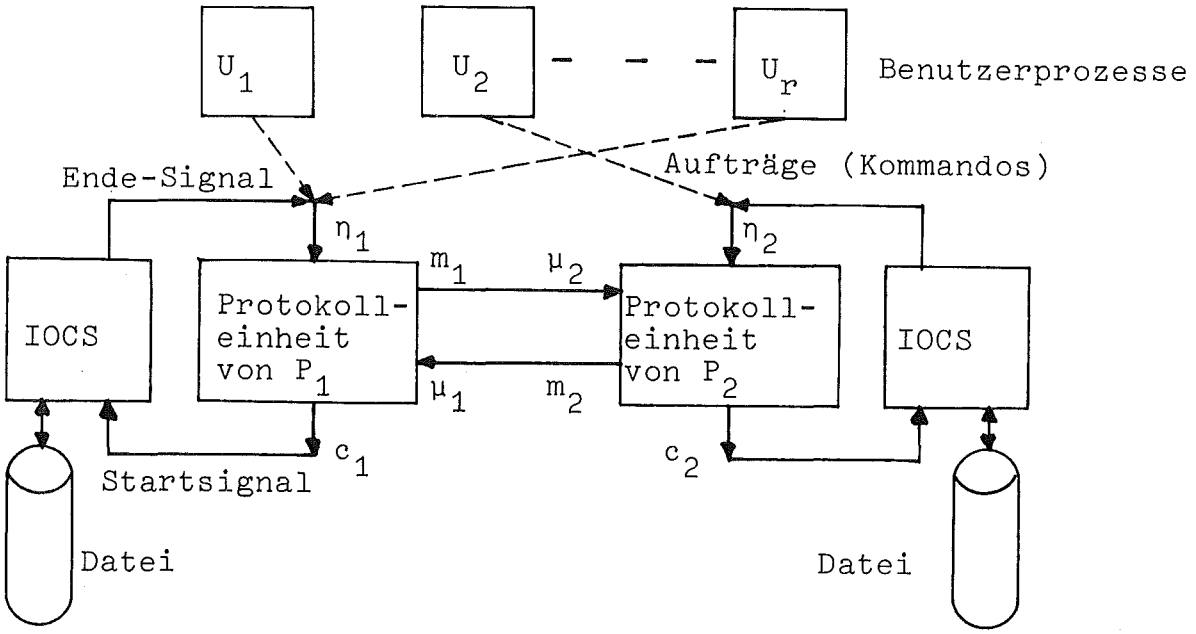


Abb. 3.1-1 Kommunizierende Protokolleinheiten in einem Zwei-Dateimanager-System

Die beendigte Ausführung des Zugriffs erzeugt am Eingang η der Protokolleinheit ein Ende-Signal, welches in eine entsprechende Nachricht an den Partnerprozeß umgesetzt werden kann. Hat ein Prozeß sowohl das Ende-Signal als auch eine Nachricht über die Beendigung des Zugriffs bei seinem Partner empfangen, so kehrt er in den Grundzustand zurück, der Zyklus ist damit geschlossen.

In diesem einfachen Fall eines Dateimanagersystems nimmt die den Ablauf steuernde Protokolleinheit mindestens zwei verschiedene Zustände an: den Grundzustand und einen "kritischen" Zustand, in dem die Durchführung des kritischen Zugriffs erfolgt. Während des Testabschnitts des Dateimanager-Prozesses findet ein Übergang vom Grundzustand in den kritischen Zustand statt; der Restabschnitt führt vom kritischen Zustand zurück in den Grundzustand und der kritische Abschnitt entspricht einem Verharren (für die Dauer der Durchführung des kritischen Zugriffs) im kritischen Zustand.

Den Dateimanager, der als erster (aufgrund eines angenommenen Auftrags) in die Testphase tritt, wollen wir als Initiator bezeichnen. Der Initiator benachrichtigt den zweiten Dateimanager über seine Bereitschaft, den Zugriff zuzulassen und verharrt bis zum Erhalt einer Antwortnachricht in einem Zwischenzustand. Da nach Beendigung des kritischen Zugriffs in der Regel nicht mit dem gleichzeitigen Empfang des Ende-Signals und der daraus resultierenden Nachricht vom anderen Dateimanager zu rechnen sein wird, ist beim Übergang vom kritischen in den Grundzustand ebenfalls das Verharren in einem Zwischenzustand möglich. Wir kommen damit zu vier verschiedenen Zuständen der Protokolleinheit, von denen während eines Zyklus mindestens zwei, der Grundzustand und der kritische Zustand, durchlaufen werden.

Um den Zustands- bzw. Nachrichtengraphen (vgl. Kap. 2.3.), der das Verhalten der einzelnen Protokolleinheit in unserem Dateimanagersystem beschreibt, aufstellen zu können, wollen wir für das hier betrachtete System folgende Vereinbarungen treffen:

Die Menge der zulässigen Steuerkommandos an Eingang η beider Dateimanager werde beschrieben durch das Alphabet

$$I_\eta = \{a,d,\lambda\}$$

Das Signal a bedeutet die Initialisierung eines kritischen Zugriffs (durch den Auftrag eines Benutzerprozesses), d kennzeichnet die beendete Durchführung und λ steht für das leere Kommando (Nullkommando).

Für das Alphabet O_c der von der Protokolleinheit erzeugten Steuersignale an Ausgang c gelte:

$$O_c = \{u,\lambda\}$$

wo u für das Startsignal (Aufforderung zur Durchführung des kritischen Zugriffs) steht. λ ist definiert wie oben.

Für den Austausch der Kontrollinformationen über die Ein- und Ausgänge μ und m wollen wir annehmen, daß die verschiedenen Typen von Kontrollnachrichten Elemente der Alphabete I_μ und O_m sind, für die im Falle eines symmetrischen Dateimanager-Systems, wie wir es hier betrachten wollen, gilt

(3.1-1) $$I_{\mu_1} = I_{\mu_2} = I_\mu$$

$$O_{m_1} = O_{m_2} = O_m$$

und $$O_m = I_\mu$$

mit $$I_\mu = \{A_1, A_2, E, \Lambda\}$$

A_1: = Bereitschaft, den vom Dateimanager P_1 angenommenen Auftrag auszuführen

A_2: = Bereitschaft, den vom Dateimanager P_2 angenommenen Auftrag auszuführen

E : = kritischer Zugriff ist ausgeführt

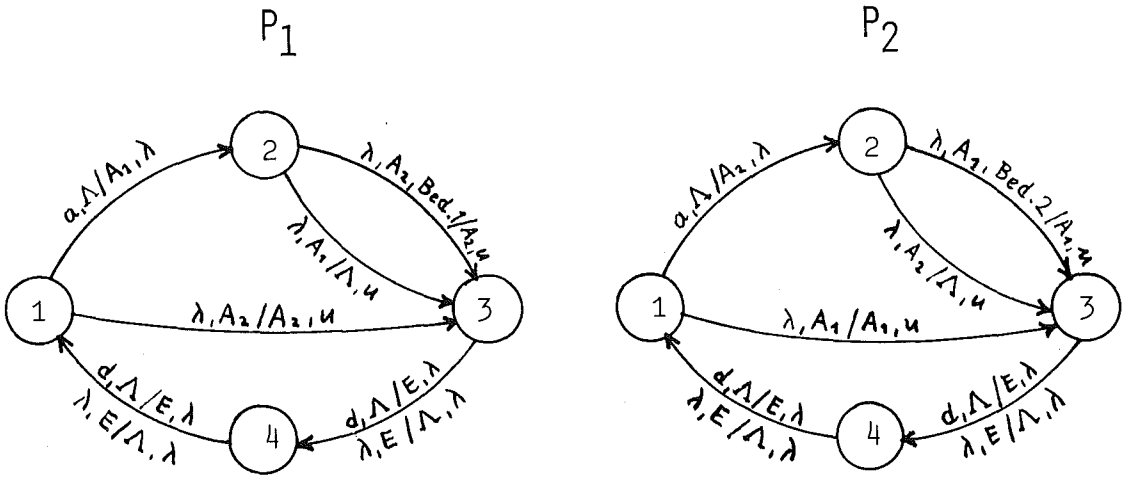
Λ : = keine Nachricht wird übermittelt (leere- oder Nullnachricht)

Schließlich sei die Zustandsmenge beider Dateimanager beschrieben durch

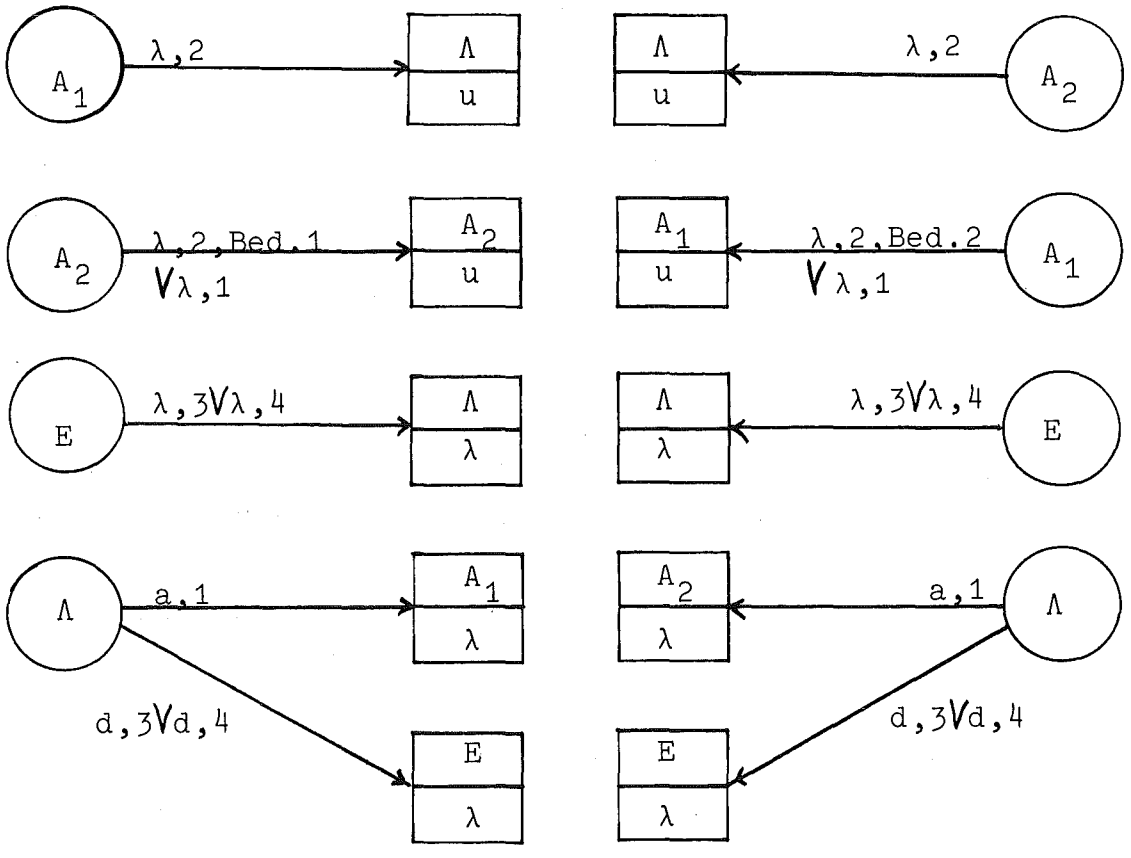
$$S = \{1,2,3,4\}$$

wobei die Stati Grundzustand, Zwischenzustand in der Testphase, kritischer Zustand und Zwischenzustand in der Restphase in dieser Reihenfolge durch entsprechende ganze Zahlen gekennzeichnet sind.

Wir kommen damit zu den in Abbildung 3.1-2 wiedergegebenen, zueinander symmetrischen Darstellungen der Abläufe in den Protokolleinheiten der Dateimanager P_1 und P_2 durch je einen Zustands- und Nachrichtengraphen. Wir sehen, daß (wir wollen zunächst die mit Bed.1 bzw. Bed.2 zusätzlich bezeichneten Kanten ignorieren) es nur dann zu einem koordinierten Eintreten beider Prozesse in den kritischen Zustand kommt, wenn bei beiden jeweils sowohl der Empfang als auch die Aussendung einer Bestätigung für den vom gleichen Dateimanager initiierten Auftrag erfolgt ist. Dabei gehen wir davon aus, daß eine Sequentialisierung der beiden Ereignisse



(A)



(B)

Abb. 3.1-2 Zustands- und Nachrichtengraphen eines Systems mit zwei Dateimanagern

- Ankunft eines Signals an Eingang η
- Ankunft einer Kontrollnachricht an Eingang μ

vorgenommen werden kann, was einer übersichtlicheren Darstellung zugute kommt, im übrigen jedoch keine Beschränkung der Allgemeinheit darstellt.

Die oben formulierte Bedingung für den Übergang der Prozesse in den kritischen Zustand impliziert, daß jeweils nur im Grundzustand eines Dateimanagers ein Auftragssignal a am Steuersignaleingang η zu einer Neuaktivierung des Systems von außen (von den Benutzerprozessen her) führt. Dieser Umstand verhindert jedoch Konfliktsituationen dann nicht, wenn die Wahrscheinlichkeit dafür, daß während der für den Transfer einer Kontrollnachricht von einem Prozeß zum anderen benötigten Zeit an beiden Dateimanagern unterschiedliche Anforderungen auf Durchführung eines kritischen Zugriffs eintreffen, von Null verschieden ist.

Denn wenn an P_1 und P_2 verschiedene Aufträge zur Durchführung eines kritischen Zugriffs eintreffen und sich P_1 und P_2 jeweils im Grundzustand befinden, führt dies zum Übergang von P_1 und P_2 nach Zustand 2 (aber jeweils für einen anderen Auftrag), was zu einer permanenten Blockierung beider Prozesse und damit des Gesamtsystems führt.

Die Möglichkeit derartiger Konfliktsituationen zwingt zur Einführung einer Vorrangregelung , die für das Gesamtsystem globale Geltung haben muß und durch eine Festlegung der Bedingungen Bed.1 und Bed.2 für die zusätzlichen Zustandsübergänge 2 → 3 (vgl. Abb. 3.1-2) in eindeutiger Weise bestimmt, welcher Auftrag verdrängt und welcher bearbeitet wird. Eine Vorrangregelung legt Prioritäten der sich um eine Bearbeitung bewerbenden Aufträge fest, so daß die Formulierung von Bed.1 und Bed.2 wie folgt vorgenommen werden kann:

(3.1-2) Bed.1: = Priorität (A_2) > Priorität (A_1)

Bed.2: = Priorität (A_1) > Priorität (A_2)

Eine eindeutige Vorrangregelung ist durch die Überprüfung obiger Bedingungen dann gewährleistet, wenn entweder bei der Generierung der Aufträge dafür gesorgt wird, daß keine zwei Aufträge gleicher Priorität in einem Zeitintervall erzeugt werden, welches der maximalen Auftragsbearbeitungsdauer (Zyklusdauer eines Dateimanagers) entspricht, oder wenn die Priorität eines Auftrags durch den Initiator aufgrund einer einmal getroffenen Vereinbarung (z.B.: Priorität aller von P_1 akzeptierten Aufträge ist kleiner als die Priorität aller von P_2 akzeptierten Aufträge) festgelegt wird. Die letztere von beiden Regelungen führt im Zwei-Dateimanager-System dazu, daß eine der beiden Bedingungen Bed.1, Bed.2 a priori falsch ist.

Abbildung 3.1-3 zeigt den kombinierten Nachrichtengraphen des in Abbildung 3.1-2 beschriebenen Zwei-Dateimanager-Systems. Berücksichtigen wir zunächst nur die dem Austausch von Kontrollnachrichten entsprechenden Kanten, so stellen wir fest, daß der kombinierte Nachrichtengraph in vier nicht zusammenhängende Teilgraphen zerfällt, bedingt dadurch, daß nicht jede empfangene Kontrollnachricht die Generierung einer Antwortnachricht zur Folge hat.

Zu einem generativen Schema für zulässige Sequenzen von Kontrollnachrichten gelangen wir indes, wenn wir die Rückwirkungen der am Ausgang c der Protokolleinheiten erzeugten Steuersignale auf die am Eingang η empfangenen Steuerkommandos berücksichtigen. Dies ist in Abbildung 3.1-3 durch die Einführung zusätzlicher Kanten für die Wechselwirkung zwischen dem Signal u an Ausgang c und dem aufgrund des Empfanges von Signal d an Eingang η erfolgenden Erzeugung der Kontrollnachricht E geschehen. Unter Zugrundelegung eines festen Prioritätsschemas (vgl. oben)

$$\text{Bed.1} \equiv \text{True}$$
$$\text{Bed.2} \equiv \text{False}$$

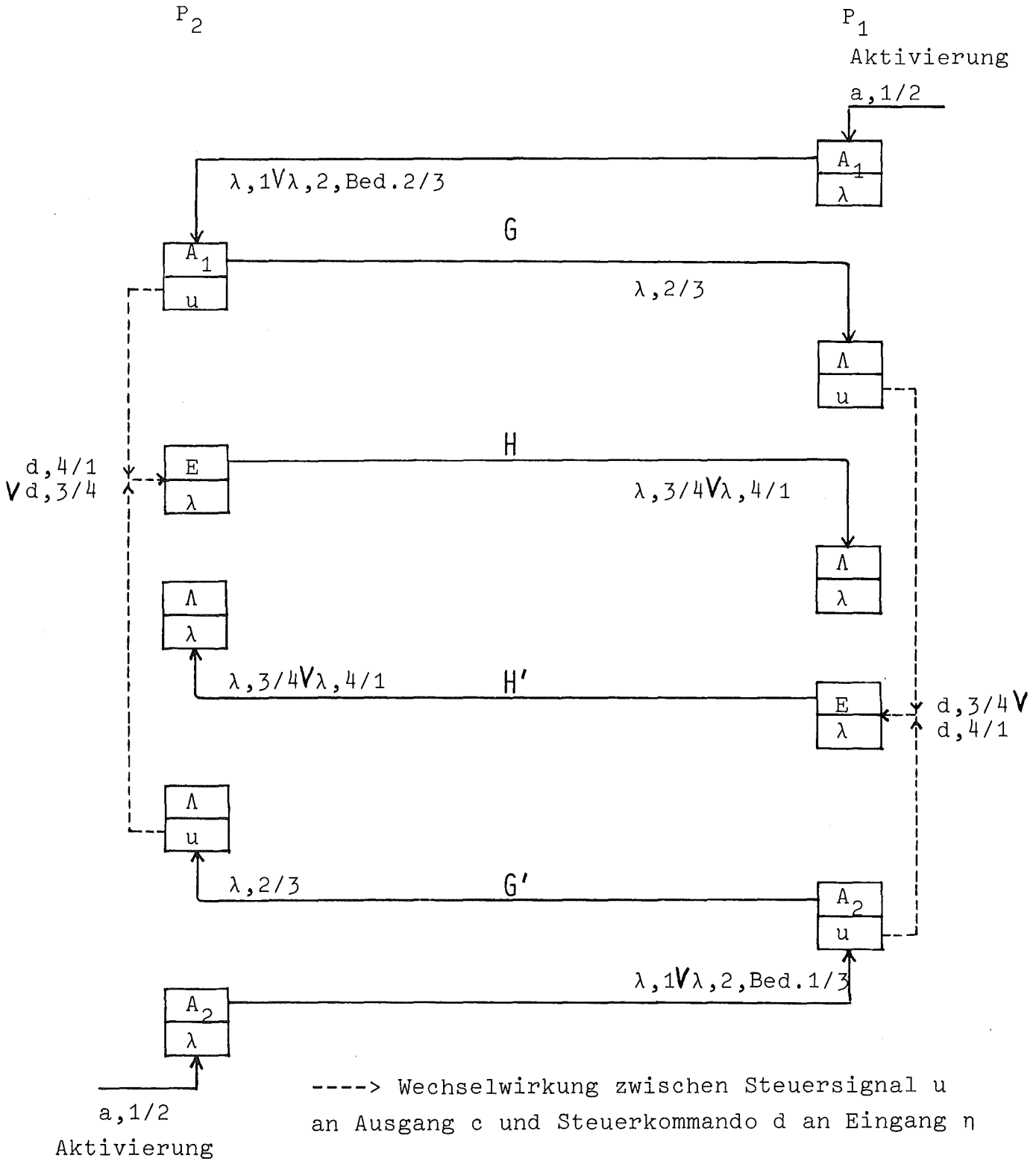


Abb. 3.1-3 Kombinierter Nachrichtengraph eines Zwei-Datei=manager-Systems als generatives Schema zur Erzeugung der zu=lässigen Kommunikationssequenzen (einstufiges Protokoll). Die Kommunikationssequenzen erhält man durch das in 2.3 beschriebene Vorgehen, unter zusätzlicher Berücksichtigung der Wechselwirkungen zwischen Steuersignal u und dem Steuerkommando d .

erhalten wir vier zulässige Kommunikationssequenzen:

P_1 ist Initiator:	$A_1\ A_1\ E\ E\ .....$
P_2 ist Initiator:	$A_2\ A_2\ E\ E\ .....$
P_1 und P_2 sind Initiatoren:	$A_1\ A_2\ A_2\ E\ E\ .....$ $A_2\ A_1\ A_2\ E\ E\ .....$

die sich in beliebiger Reihenfolge aneinanderreihen können.

Anhand des in Abb. 3.1-3 dargestellten Schemas der einfachsten Form eines Koordinationsprotokolls ("einstufiges" Protokoll, s.u.) können wir nun ein komplexeres Kommunikationsprotokoll zur Koordination von Dateimanagern aufbauen, welches folgender Anforderung genügt:

Sei $(Z_{t_1}, Z_{t_2} ..... Z_{t_i} ..... Z_{t_n})$

eine Folge von Aufträgen zur Durchführung kritischer Zugriffe die zu den Zeitpunkten $t_1, t_2, ..., t_n$ im Rechnernetz generiert werden mit

$$t_{i-1} < t_i \qquad \text{für } i=2(1)n$$

Aufgrund ungleichmäßiger Zeitverzögerungen beim Transport der Anforderungen über das Kommunikationssystem des Rechnernetzes treffen die Aufträge jedoch in gestörter Reihenfolge bei den Dateimanagern ein. Wir wollen annehmen, daß die Durchführung der kritischen Zugriffe in der Reihenfolge ihrer Generierung (Konsistenzherstellung) angestrebt wird; es wird daher eine Konstruktion des Kommunikationsprotokolls gesucht, welche diese Aufgabe der Konsistenzherstellung möglichst weitgehend erfüllt. (Probleme dieser Art treten speziell in Realzeitsystemen auf; man denke etwa an die Experiment-Datenerfassung, on-line Lagerbestandserfassung, Platzbuchungssysteme etc.)

Es sei angenommen, daß für zwei sich um die Durchführung eines kritischen Zugriffs bewerbende Aufträge Z_{t_i} und Z_{t_k} gilt:

(3.1-3) $\quad$ Priorität (Z_{t_i}) > Priorität (Z_{t_k})

dann und nur dann, wenn $t_i < t_k$

Um den Konfliktfall $t_i = t_k$ (die Wahrscheinlichkeit dafür ist eine Funktion der Genauigkeit und Auflösung der Zeitangabe) zu eliminieren, kann ein festes Prioritätsschema derart überlagert werden, daß

(3.1-4) $\quad$ Priorität $(Z_{t_i}) \neq$ Priorität (Z_{t_k}) $\quad$ für $t_i = t_k$

wenn Z_{t_i} und Z_{t_k} an verschiedenen Dateimanagern zur Initialisierung kommen.

Seien t_i', t_k' mit $t_i' < t_k'$ die Ankunftszeiten der Aufträge Z_{t_i} und Z_{t_k} nach Durchlaufen des Kommunikationssystems im Dateimanagersystem.

Unter der Voraussetzung der Konvergenz der Wahrscheinlichkeit

(3.1-5) $\quad$ $\text{Prob}(t_i - t_k \geq 0) \rightarrow 0 \quad$ für $t_k' - t_i' \rightarrow \infty$

kann durch Einbeziehung eines "Karenzzeitintervalles" neben der durch Verdrängung von Aufträgen aufgrund der Bedingungen (3.1-2) erfolgenden Konsistenzherstellung eine zusätzliche Ordnung in der Bearbeitungsreihenfolge erreicht werden. Dazu sei neben dem Steuersignal u am Ausgang c der Protokolleinheit in Abb. 3.1-2 ein weiteres Signal v zugelassen, welches den Zeitdienst des zuständigen Rechnersystems veranlaßt, nach Ablauf der an (3.1-5) orientierten Karenzzeit T_K ein Signal an Eingang η zu bewirken. Dieses Signal sei durch das Symbol b charakterisiert, wenn während der Zeit T_K kein Auftrag höherer Priorität als der des bereits initialisierten beim betroffenen Dateimanager einging; im anderen Falle werde mit dem Ablauf

von T_K das Signal a erzeugt, wodurch die Aufnahme der Bearbeitung des Auftrags mit der nun höchsten Priorität initialisiert werden kann. Es wird hier der Aufbau und die Verwaltung einer prioritätsorientierten Warteschlange von Aufträgen vor den einzelnen Dateimanagern vorausgesetzt; Überlegungen zur Warteschlangenverwaltung sollen jedoch in Kapitel 4 angestellt werden.

Wir kommen damit zu einem erweiterten, im folgenden als "zweistufig" bezeichneten Koordinationsprotokoll. Ausgehend von den Teilgraphen G und G' in Abb. 3.1-3 wollen wir nun die Teile G_1 und G_1' des Graphen für das zweistufige Protokoll aufbauen, die die Einleitung des Karenzzeitintervalls bewirken. Dies geschieht in einfacher Weise durch Ersetzung des Steuersignals u durch das Signal v, die Ergänzung der Markierung der Kanten wollen wir zunächst ignorieren.

Zustand 3 entspricht nun natürlich nicht mehr dem kritischen Zustand, den wir in diesem Falle mit der Nummer 5 versehen wollen; entsprechend wird Zustand 4 zu Zustand 6.

Die neuen Zustände 3 und 4 bezeichnen den Wartezustand (warten auf den Ablauf der Karenzzeit) und den Zustand der einseitigen Bereitschaft, den durch die Protokollteile G_1 bzw. G_1' initialisierten Zugriff auszuführen. Der Übergang vom Wartezustand 3 in den kritischen Zustand nach Ablauf der Karenzzeit wird, sofern es bei der durch G_1 bzw. G_1' erzielten Übereinstimmung bleibt, durch die neu hinzukommenden Teilgraphen F_1 und F_1' beschrieben. Die gegenseitige Unterrichtung der Dateimanager darüber, daß sich an der vor dem Eintritt in das Karenzzeitintervall erzielten Übereinstimmung nichts geändert hat, erfolgt durch Übermittlung der Kontrollnachricht B.

Ergibt sich jedoch während der Karenzzeit eine Änderung derart, daß ein neuer Auftrag den bereits initialisierten Auftrag verdrängt, so kann diesem Umstand durch Reaktivierung der Protokollteile G_1 bzw. G_1' Rechnung getragen werden. Dazu ist jedoch eine entsprechende Erweiterung der Bedingungen für den Übergang nach Zustand 3 in G_1 und G_1' notwendig.

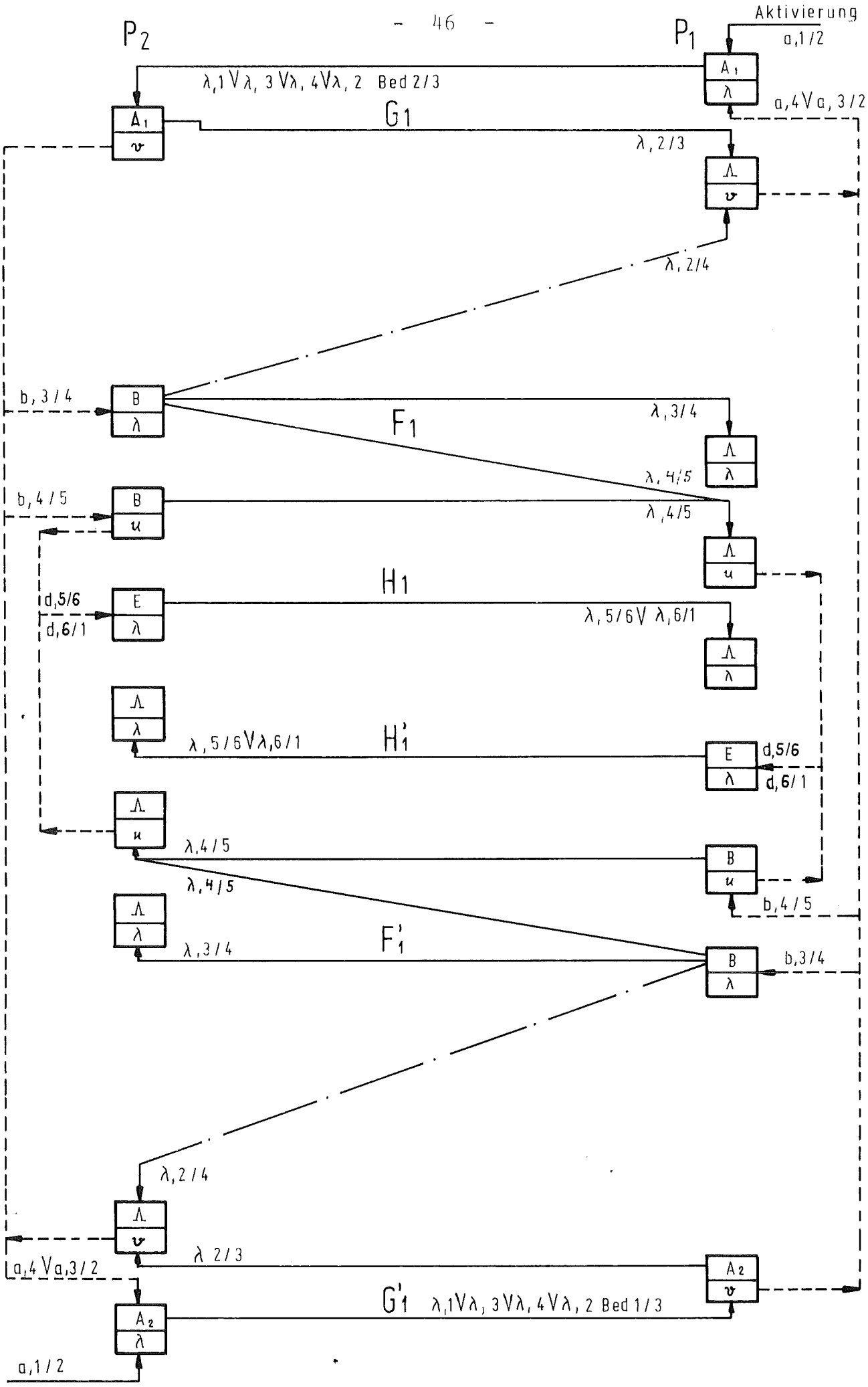


Abb 3.1-4 Kombinierter Nachrichtengraph des erweiterten (zweistufigen) Kommunikationsprotokolls des Zwei-Dateimanager-Systems

Die zueinander symmetrischen Teilgraphen H und H' finden sich im erweiterten Protokoll in unveränderter Form in den Teilgraphen H_1 und H_1' wieder und beschreiben auch hier den Übergang der Protokolleinheiten aus dem kritischen Zustand in den Grundzustand.

Die zur Beschreibung der Protokolleinheiten notwendigen Alphabete haben nun im Falle des erweiterten Protokolls die Form

$$I_\mu = O_m = \{A_1, A_2, B, E, \Lambda\}$$
$$I_\eta = \{a, b, d, \lambda\}$$
$$S = \{1, 2, 3, 4, 5, 6\}$$
$$O_c = \{v, u, \lambda\}$$

Ein Vergleich der Struktur der beiden hier entwickelten Koordinationsprotokolle mit unterschiedlichem Konsistenzherstellungsvermögen zeigt, daß der prinzipielle Unterschied in der Zahl der Synchronisationsstufen bis zum Eintritt in den kritischen Zustand liegt. Das einfache erste Protokoll ist dem Prinzip nach einstufig, während das zweite Protokoll im konfliktfreien Fall zweistufig ist, beim Auftreten von Verdrängungen jedoch eine beliebige Zahl von Stufen durchlaufen kann und somit (3.1-5) als Voraussetzung für seine Anwendbarkeit verlangt. Wir wollen im folgenden die beiden Protokolltypen durch die Attribute einstufig und zweistufig klassifizieren.

Bei der Realisierung zweistufiger Protokolle zur Herstellung der maximal erreichbaren Konsistenz der Bearbeitungsreihenfolge von Aufträgen ist für den Fall, daß der Zeitbedarf für die Nachrichtenübermittlung (Kommunikationsverzögerung) in der gleichen Größenordnung wie das an (3.1-5) orientierte Karenzzeitintervall liegt, der Sonderfall $T_K \to 0$ von Interesse. Dieser Spezialfall kann ebenfalls durch das in Abb. 3.1-4 gezeigte Schema abgedeckt werden. Man sieht jedoch, daß es in diesem Falle zu Überholvorgängen in bezug auf die von einem Dateimanager generierten Kontrollnachrichten der Typen A_i mit i=1,2 und B kommen kann, wenn der Zeitbedarf für die Nachrichtenübertragungen Schwankungen

unterliegt. Durch die Verbindung der Teilgraphen G_1, F_1 und G_1', F_1' durch eine zusätzliche, mit der Markierung λ, 2/4 versehene Kante kann jedoch auch dieser Fall in der in Abb. 3.1-4 gegebenen formalen Beschreibung des zweistufigen Protokolls berücksichtigt werden.

3.2. Erweiterung auf Systeme mit einer beliebigen Zahl von Dateimanagern

Wir wollen nun versuchen, die in Kapitel 3.1. anhand formaler Beschreibungen entwickelten zwei Protokolltypen für die Koordination von Dateimanageroperationen auf Systeme mit einer nach oben nicht begrenzten Zahl von Dateimanager-Prozessen zu erweitern. Es gilt zu zeigen, daß auch in diesem Falle eine Koordination kritischer Zugriffe durch einstufige oder zweistufige Protokolle erzielt werden kann, unter der Voraussetzung, daß die Aktionen der einzelnen Dateimanager durch die Operationen von Protokolleinheiten des in Abb. 3.1-1 dargestellten Typs beschrieben werden können.

Eine Verallgemeinerung der in 3.1. gefundenen notwendigen Voraussetzungen für die Koordinierung einer beliebigen Zahl r von Dateimanagerprozessen durch Protokolle der in Abb. 3.1-3 und 3.1-4 beschriebenen Art verlangt:

a) den spontanen Übergang eines Dateimanagers in den kritischen Zustand nach Empfang von r-1 Kontrollnachrichten, die die Bereitschaft der restlichen Dateimanager zur Durchführung des gleichen kritischen Zugriffs bestätigen, wenn der Dateimanager seinerseits durch Aussendung von r-1 identischen Kontrollnachrichten allen übrigen Dateimanagern seine Bereitschaft zur Durchführung des kritischen Zugriffs bestätigt hat;

b) die Rückkehr eines Dateimanagers in den Grundzustand, wenn die lokale Durchführung des kritischen Zugriffs erfolgt und die Bestätigung der Durchführung des Zugriffs durch die r-1 übrigen Prozesse übermittelt worden ist;

c) eine eindeutige Festlegung von Vorrangregeln durch Einführung von Auftragsprioritäten zur Auflösung von Konfliktsituationen;

d) daß eine Verdrängung von bereits in Bearbeitung befindlichen Aufträgen durch Aufträge höherer Priorität nur möglich ist, solange sich die Dateimanager im Testabschnitt befinden.

Voraussetzung d) bedeutet, daß sowohl beim einstufigen als auch beim zweistufigen Koordinationsprotokoll preemptive Maßnahmen (Verdrängungen) von einem Dateimanager auf Grund der Nachricht eines anderen Dateimanagers nur dann durchgeführt werden können, wenn noch nicht r-1 Kontrollnachrichten empfangen worden sind, die gemäß Voraussetzung a) die Bereitschaft der anderen Dateimanager zur Ausführung des kritischen Zugriffs bestätigen. Daraus folgt wiederum, daß ein Dateimanager nur solange aktiv die Verdrängung einer bereits initialisierten Auftragsbearbeitung durch Entgegennahme eines neuen Auftrags bewirken kann, wie er noch keine Bereitschaftsmeldung $A_i \varepsilon I_\mu$ (vgl. (3.1-1)) zur Durchführung des zu verdrängenden Auftrags abgesandt hat. Alle Zustände eines Dateimanagers für die letzteres gilt, wollen wir zu einem makroskopischen Grundzustand GR zusammenfassen; der Rest der Testphase sei durch den makroskopischen Zustand TE und die kritische Phase durch den Zustand KR erfaßt.

Wir können nun zeigen, daß unter den Voraussetzungen 3.2a) bis 3.2d) jedes r-Dateimanagersystem, dessen Protokolleinheiten dem in Kapitel 3.1. (siehe Abb. 3.1-1) beschriebenen Typ entsprechen, die exklusive Inanspruchnahme der Gesamtheit der kontrollierten Dateien durch ein und denselben Auftrag gewährleistet, indem wir die Richtigkeit des folgenden Satzes beweisen:

Führt die Initialisierung der Bearbeitung eines kritischen Zugriffs Z_i aus der Menge der konkurrierenden Aufträge in einem System von r Dateimanagern $\{P_1,\ldots,P_r\}$ zum Eintritt der Prozesse $\{P_1,\ldots,P_k\}$, $1 \leq k \leq r$ in den kritischen Zustand (und damit zur lokalen Durchführung des kritischen Zugriffs), so führt sie auch zum Eintritt der Prozesse $\{P_{k+1},\ldots,P_r\}$ in den kritischen Zustand für Z_i.

Seien $\{P_1,\ldots,P_k\}$ in Zustand KR für Auftrag Z_i. Ist der Satz falsch, so muß es möglich sein, daß $\{P_{k+1},\ldots,P_r\}$ für Aufträge $Z_j \neq Z_i$ nach Zustand KR übergehen, ohne vorher Zustand KR für Auftrag Z_i eingenommen und dann Grundzustand GR durchlaufen zu haben. Dies ist aber nicht möglich, da gemäß Voraussetzung 3.2a) die Dateimanager $\{P_{k+1},\ldots,P_r\}$ mindestens in Zustand TE sein müssen, damit $\{P_1,\ldots,P_k\}$ sich im kritischen Zustand befinden können. In Zustand TE ist jedoch als Folgerung aus Voraussetzung 3.2d) die Annahme neuer Aufträge Z_j nicht zulässig.

In Anlehnung an das von Gilbert und Chandler /18/ vorgeschlagene Verfahren zur Beschreibung der Wechselwirkungen zwischen kommunizierenden parallelen Prozessen wollen wir nun versuchen, Zustände und mögliche Zustandsübergänge eines Systems mit einer beliebigen Zahl von Dateimanagern zu erfassen, um dadurch die Richtigkeit der entwickelten Kommunikationsprotokolle nachweisen zu können. Dies erfolgt in der Weise, daß die Unmöglichkeit von Übergängen des Systems zu unzulässigen Zuständen gezeigt wird.

Der Zustand eines aus r Dateimanagern $\{P_1,\ldots,P_r\}$ bestehenden Systems ist gegeben durch die Zustände p_i, i=1(1)r dieser Prozesse und die Werte d_{ik} einer Menge assoziierter Variabler $\{\nu_{11},\ldots,\nu_{ik},\ldots,\nu_{rr}\}$. Wir wollen die Variablen ν_{ik} mit den logischen Kommunikationskanälen zwischen je zwei Dateimanagern i und k identifizieren; der Wert d_{ik} der Variablen entspricht der durch Übertragung über diesen Kanal empfangenen Kontrollnachricht.

Der kombinierte Gesamtzustand s des Dateimanagersystems wird dargestellt durch das r-Tupel $(p_1,\ldots,p_r)$ kombiniert mit der Gesamtheit der Werte der assoziierten Variablen (d_{ik}):

$$(3.2\text{-}1) \qquad s = (p_1 \ldots p_r)\ (d_{ik}) \qquad i,k=1(1)r$$

wobei $p_i \varepsilon \{GR,TE,KR\}$ mit i=1(1)r den Zustandswert von P_i und d_{ik}, i,k=1(1)r die Kontrollnachricht im (logischen) Übertragungskanal ν_{ik} von P_i nach P_k angibt.

Die Werte der Variablen ν_{ii}, i=1(1)r, sollen den von P_i an alle übrigen Dateimanager P_k mit k≠i zuletzt übermittelten Nachrichten entsprechen.

Aufgrund der durch das physikalische Übertragungssystem bedingten endlichen Übertragungszeit für Nachrichten zwischen den Dateimanagern ergibt sich zwangsläufig, daß die von einem Dateimanager P_k empfangene Nachricht d_{ik} durchaus vom augenblicklichen Wert von d_{ii} verschieden sein kann.

Die Zustandsübergänge des durch die Prozesse und assoziierten Variablen beschriebenen Gesamtsystems

(3.2-2) $$\{P_1,\ldots,P_r,\nu_{11},\ldots,\nu_{ik},\ldots,\nu_{rr}\}$$

ergeben sich aus den Kombinationen der Elemente der Menge partieller Regeln (vgl. /18/), die die mit dem Testen und Setzen der assoziierten Variablen verbundenen Zustandsübergänge der Einzelprozesse festlegen. In Anlehnung an die in /18/ eingeführte Notation haben diese partiellen Regeln die Form

(3.2-3)

$$(x..x\ p_i\ x..x)\begin{pmatrix} x..x & d_{1i} & x..x \\ \vdots & & \vdots \\ \vdots & d_{ii} & \vdots \\ x..x & d_{ri} & x..x \end{pmatrix} \rightarrow (x..x\ p_i'\ x..x)\begin{pmatrix} x & \cdots\cdots & x \\ \vdots & & \vdots \\ \vdots & d_{ii}' & \vdots \\ x & \cdots\cdots & x \end{pmatrix}$$

entsprechend einem Testen der empfangenen und Setzen der auszusendenden Kontrollnachrichten durch einen Prozeß P_i, i=1(1)r. Das Zeichen x deutet hier an, daß der betreffende Wert im Zusammenhang mit der partiellen Regel nicht relevant ist.

Der in (3.2-3) als partielle Regel formulierte Zustandsübergang orientiert sich dabei stets an lokalen Informationen, die nur einem Teil des globalen Zustands (3.2-1), der einem "idealen Beobachter" zugänglich wäre, entsprechen.

Versetzen wir uns nun in die Lage eines solchen idealen Beobachters, so können wir diejenigen Zustände des Gesamtsystems präzisieren, in die unter keinen Umständen ein Zustandsübergang führen darf. Dies sind unter Verwendung des in 3.1. eingeführten und sinngemäß erweiterten Alphabets der Kontrollnachrichten $I_\mu = \{A_1, A_2, \ldots, A_r\}$ alle Zustände der Art

$$(3.2\text{-}4) \qquad (\ldots KR..KR..) \begin{pmatrix} x \ldots A_j \ldots A_m \ldots x \\ A_j \quad A_m \\ A_j \quad A_m \\ x \ldots A_j \ldots A_m \ldots x \end{pmatrix} \quad \text{mit } m \neq j \text{ und } m,j \varepsilon \{1,..,r\}$$

die dem gleichzeitigen Übergang mehrerer Dateimanager in den kritischen Zustand zur Ausführung verschiedener Aufträge entsprechen.

Wenn für einen beliebigen Spaltenvektor d_k von (d_{ik}) gilt

$$d_{1k} = d_{2k} = \ldots = d_{rk} = A_m$$

folgt jedoch aus den Voraussetzungen 3.2a)-3.2d):

$$(3.2\text{-}5) \qquad d_{ii} = A_m \qquad \text{für alle } i=1(1)r$$

Damit ist die Unmöglichkeit eines Übergangs des Dateimanagersystems in den Zustand (3.2-4) gezeigt.

3.3. Fehlertolerante Koordinationsprotokolle

Systeme kooperierender Prozesse in Rechnernetzen sind auf vielfache Art und Weise Störsituationen ausgesetzt, die die Funktion eines solchen Systems unter Umständen erheblich beeinträchtigen können /37/. Wir wollen im folgenden untersuchen, in welchem Umfange die durch die Konzeption des hier betrachteten Systems zur Überwachung kritischer Zugriffe bedingte redundante

Auslegung die Umgehung kritischer Störsituationen bei geeigneter Erweiterung der erarbeiteten Kommunikationsprotokolle ermöglicht.

Die für ein System von Dateimanagern relevanten Störfälle entstehen durch

a) Ausfälle einzelner Rechner und damit der dort lokalisierten Prozesse

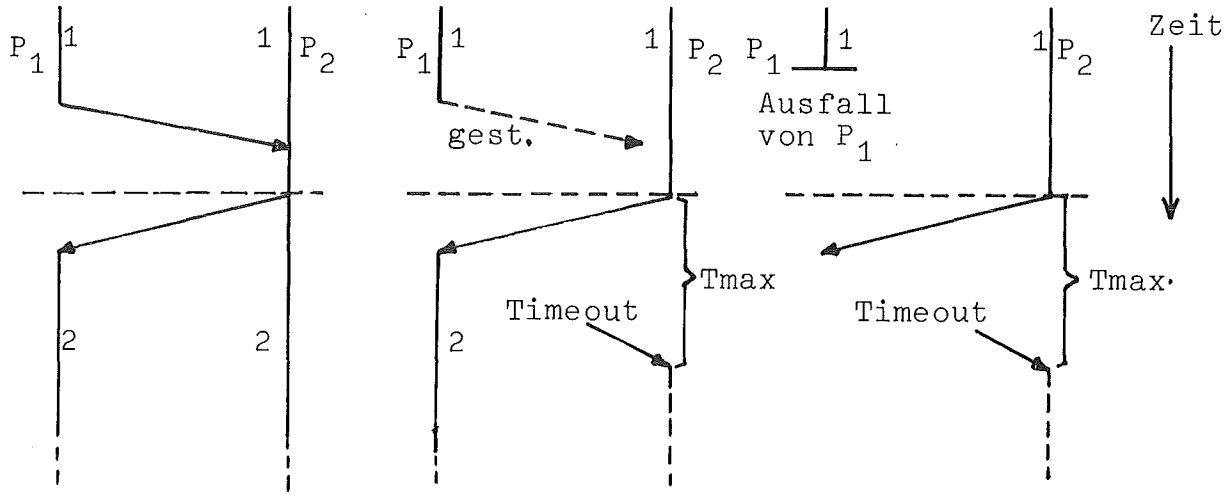
b) Ausfälle und Fehlfunktionen des Kommunikationssystems

In bezug auf Störfälle vom Typ b), die auf Fehlfunktionen des Übertragungssystems zurückzuführen sind, kann bei geeigneter Implementierung der Elementarfunktionen zur Interprozeßkommunikation in Rechnernetzen angenommen werden, daß die resultierenden Fehlübertragungen von Nachrichten durch Kommunikationsprotokolle auf unteren Ebenen der Protokollhierarchie (vgl. Kap. 2.3.) mit hoher Wahrscheinlichkeit korrigiert werden. Wir wollen uns daher der Betrachtung solcher Störfälle zuwenden, die entweder durch Ausfälle einzelner Dateimanager oder Ausfälle von Interprozeß-Kommunikationsverbindungen bedingt sind. Beide Fälle äußern sich im Ausbleiben von Kontrollnachrichten, wenn die Fehlfunktion in die Bearbeitungsphase eines Auftrags zur Durchführung eines kritischen Zugriffs fällt.

Da das Ausbleiben von (erwarteten) Kontrollnachrichten, wie der Zustandsgraph des Dateimanagers in Abb. 3.1-2 z.B. leicht erkennen läßt, zur Blockierung der von einer Störung nicht direkt betroffenen Prozesse führen kann, muß zur Vermeidung einer derartigen Situation die Einführung eines speziellen Signals, eines "Timeout", vorgenommen werden. Die Erzeugung von "Timeouts" durch Inanspruchnahme des Systemzeitdienstes muß sich dabei am maximal tolerierbaren Zeitintervall orientieren, innerhalb dessen erwartete Kontrollnachrichten eintreffen müssen; die Länge dieses Zeitintervalls ergibt sich in Abhängigkeit vom Zustand der wartenden Protokolleinheit.

Das "Timeout"-Signal kann vom Dateimanagerprozeß zur Erzeugung einer Pseudo-Kontrollnachricht an die Protokolleinheit herangezogen werden, die das auf andere, eindeutig identifizierbare Dateimanager bezogene Ausbleiben von erwarteten Kontrollnachrichten anzeigt.

Aufgrund eines erzeugten "Timeout"-Signals kann von einem Dateimanager bzw. seiner zugeordneten Protokolleinheit (wie aus Abb. 3.3-1 hervorgeht) grundsätzlich nicht entschieden werden, ob Störungsfall a) oder b) vorliegt; es kann lediglich angenommen werden, daß der den "Timeout" empfangende Dateimanager Bestandteil eines "funktionsfähigen" Teilnetzes darstellt (wobei im Extremfall ein Teilnetz auf ein einzelnes, einen Dateimanager umfassendes, Rechnersystem reduziert sein kann). "Funktionsfähig" ist hier in dem Sinne zu verstehen, daß die Koordination kritischer Zugriffe zu den im Teilsystem verbliebenen Datenbankkomponenten noch durchführbar ist; "funktionsfähig" bedeutet jedoch nicht, daß die weitere Durchführung kritischer Zugriffe in den verbliebenen Teilsystemen die Funktionsfähigkeit der überwachten Datenbank gewährleistet: der Ausfall nicht redundant realisierter Komponenten (nur mit einer einzigen Kopie abgelegte Dateien) kann eine Datenbank unbrauchbar machen.



a) ungestörte Koordination der Zustandsübergänge 1→2 zweier Prozesse P_1,P_2

b) Störfall Typ b

c) Störfall Typ a

Abb. 3.3-1 Erkennung von Störzuständen durch Timeout

Im weiteren soll vorausgesetzt werden, daß die in einem Teilnetz verbliebenen Dateien als Komponenten einer Datenbank deren Funktionsfähigkeit gewährleisten. Der Empfang von Timeout-Signalen und der daraus resultierenden Pseudokontrollnachrichten würden dann die Protokolleinheiten der r Dateimanager eines Überwachungssystems veranlassen, Zustandsübergänge nicht erst nach Empfang von r-1 gleichlautenden Kontrollnachrichten (vgl. Kapitel 3.2.) sondern bereits nach r-1-q Nachrichten vorzunehmen, wenn q verschiedene Dateimanager nicht mehr antworten.

Neben der Erweiterung des Kommunikationsprotokolls um Vereinbarungen, die die durch Timeout-Pseudokontrollnachrichten bewirkten Aktionen der Protokolleinheiten betreffen, ist eine zusätzliche Erweiterung notwendig, die nach Beseitigung der Störungen die Rückkehr zur koordinierten Operation des Gesamtsystems ermöglicht. Ein Timeout läßt, wie angedeutet, keinerlei Rückschlüsse darauf zu, ob Störungen vom Typ a) oder b) vorliegen; die Funktionsbereitschaft von Dateimanagern kann jedoch durch differenzierte "Präsenznachrichten" kenntlich gemacht werden, die nach Wiederherstellung der Funktionsfähigkeit eines Dateimanagers sowie nach Erkennung eines Störzustandes aufgrund eines Timeout anstelle regulärer Kontrollnachrichten gesendet werden. Abbildung 3.3-2 zeigt am Beispiel eines Zwei-Dateimanager-Systems, wie eine Erweiterung des Kommunikationsprotokolls unter Einbeziehung der Pseudokontrollnachricht TO (Timeout) und von drei Typen von Präsenznachrichten

- NT Notext, keine Nachricht empfangen
- TX Text, Nachricht empfangen
- PR Präsenz, Funktionsbereitschaft hergestellt

konzipiert sein muß, damit im Falle von Störsituationen sowohl des Typs a) (Abb. 3.3-2a) als auch des Typs b) (Abb. 3.3-2b,c) eine Rückkehr zur koordinierten Operation des Gesamtsystems möglich ist.

Mit der Wiederherstellung der Kooperationsbereitschaft des Gesamtsystems von Dateimanagern taucht jedoch ein weiteres Prob-

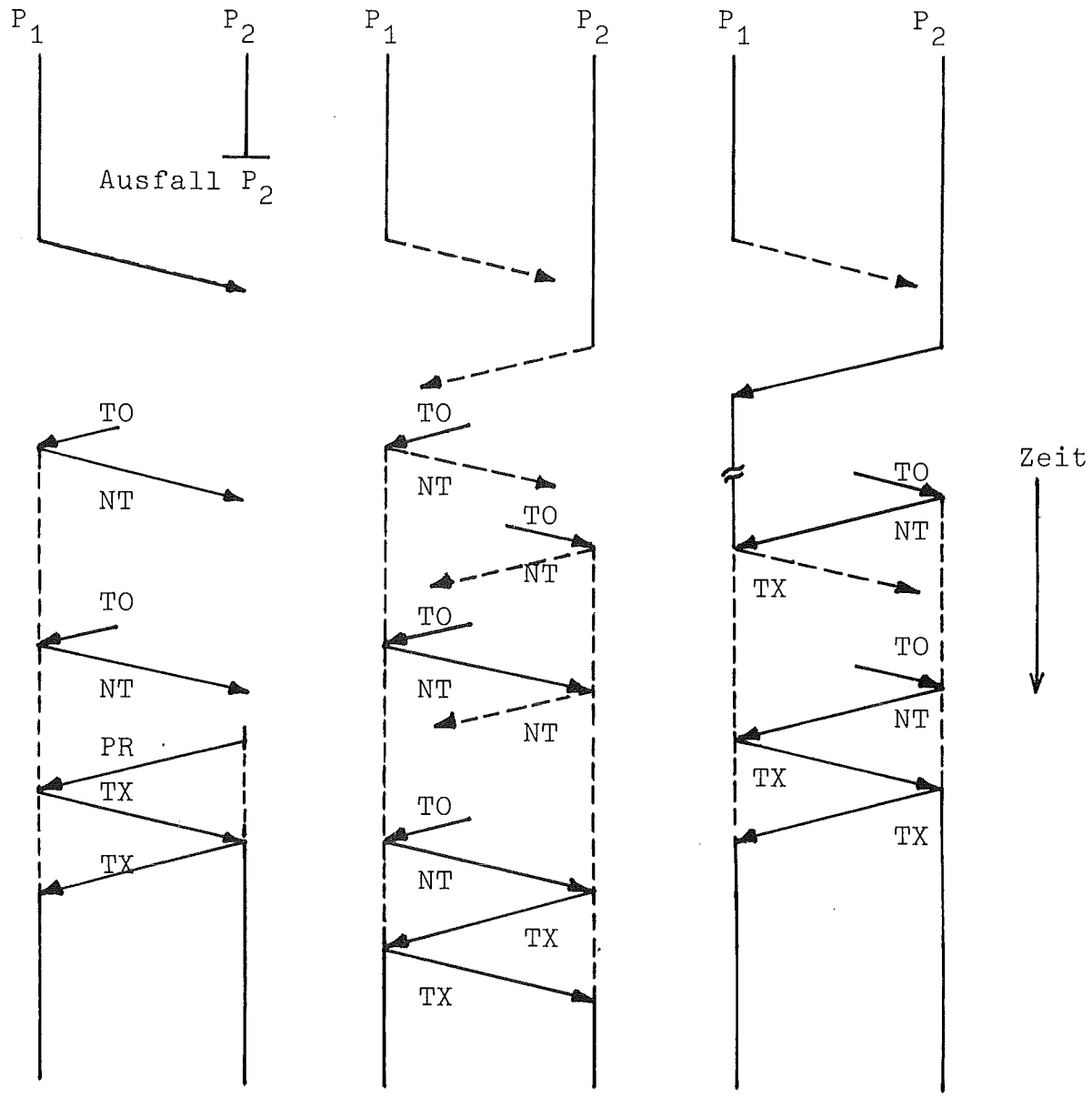


Abb. 3.3-2 Rückkehr zur koordinierten Operation der Prozesse P_1 und P_2 durch differenzierte Präsenznachrichten

lem auf: Es ist nicht sichergestellt, daß sich die überwachte Datenbank in ihrer Gesamtheit in einem konsistenten Zustand befindet. Dies kann darauf zurückzuführen sein, daß die durch eine Störung entstehenden Teilsysteme mit unterschiedlichen Folgen von Zugriffsanforderungen beaufschlagt wurden; ein Extremfall wäre, daß beim Zusammenbruch eines einzelnen Dateimanagers ein Teilsystem für die Dauer der Störung mit keiner Zugriffsanforderung konfrontiert wurde.

Ist die Reihenfolge der Durchführung kritischer Zugriffe in einem ins Auge gefaßten System irrelevant, so bereitet die erforderliche Konsistenzherstellung keine Schwierigkeiten: Die zwischenzeitlich aufgelaufenen und von einem Teilsystem nicht ausgeführten Aufträge werden in der Form entsprechender Nachrichten in die Nachrichtenwarteschlange (vgl. Kap. 4.) eines Dateimanagers des betreffenden Teilsystems überführt und gelangen dadurch zur nachträglichen Bearbeitung.

Ist jedoch die Reihenfolge der Durchführung kritischer Zugriffe relevant für die Konsistenz der in einer Datenbank abgespeicherten Information (dies gilt speziell für Dateiänderungen), so muß dafür gesorgt werden, daß es zur Ausführung unterschiedlicher Auftragssequenzen bei den durch einen Störfall entstehenden funktionsfähigen Teilsystemen nicht kommen kann, d.h. genau genommen, daß die Wahrscheinlichkeit für das Zustandekommen unterschiedlicher Auftragssequenzen gegen Null geht. Dies kann durch das in Abb. 4.2-1 skizzierte Verfahren der Mehrfachinitialisierung zur Durchführung kritischer Zugriffe realisiert werden, bei dem, ähnlich einem Schrotflinten-Effekt, von einem Benutzerprozeß aus gleichzeitig sämtliche Dateimanager zur Durchführung des gleichen kritischen Zugriffs aktiviert werden. Dabei wird vorausgesetzt, daß die kommunikativen Verbindungen zwischen Benutzer- und Dateimanagerprozessen durch den Störfall nicht beeinträchtigt werden. Die Wiederherstellung der Konsistenz einer einzelnen, durch den temporären Ausfall eines Dateimanagers betroffenen Datei bereitet auch in diesem Falle keine Schwierigkeiten, da hierbei die nachzuholende Durchführung kritischer Zu-

griffe in der erforderlichen Reihenfolge gewährleistet werden kann.

Eine weitere, mehr ins einzelne gehende Behandlung fehlertoleranter Kommunikationsprotokolle für Dateimanagersysteme unter Berücksichtigung implementierungsspezifischer Details, wie etwa der Redundanz der Kommunikationswege etc., würde den für diese Arbeit gesteckten Rahmen sprengen und soll daher gesonderten Untersuchungen vorbehalten bleiben.

4. Koordinationsverfahren auf der Basis von Kommunikationsprotokollen

4.1. Präzisierung der Dateimanagerfunktionen

Die im vorausgegangenen Kapitel erarbeiteten Koordinationsprotokolle sollen als Basis für die Entwicklung von Verfahren angewendet werden, die zur Zugriffskoordination durch eine beliebige Zahl von Dateimanagern eingesetzt werden können.

Gemäß der in 2.1. dargelegten Organisationsstruktur eines Überwachungssystems sind die für die Zugriffsabwicklung verantwortlichen Dateimanagerprozesse der Konzeption nach identisch, womit ein der Struktur nach symmetrisches System entsteht. Kernstück aller im Überwachungssystem realisierten Dateimanager sind identische Protokolleinheiten mit den durch das vereinbarte Kommunikationsprotokoll festgelegten Zustandsübergängen.

Die anhand einer formalen Beschreibung vorgenommene Festlegung zweier Protokolltypen (einstufige und zweistufige Kommunikationsprotokolle) setzt voraus, daß die Verwaltung der eingegangenen Nachrichten und der zur Bearbeitung anstehenden Aufträge (d.h. Anforderungen auf Durchführung kritischer Zugriffe) sowie die Zuordnung zwischen Nachrichten und Aufträgen über einen geeignet organisierten Mechanismus von den Dateimanagern durchgeführt wird. Erst die Ergänzung der Koordinationsprotokolle um diesen Mechanismus führt, in Verbindung mit der Spezifikation der Informationen, die in den Nachrichten und den Beschreibungen der Zugriffsaufträge enthalten sein müssen, zu den angestrebten Koordinationsverfahren. Die an den Verwaltungsmechanismus und an die Information gestellten Anforderungen sollen im folgenden präzisiert werden.

Abbildung 4.1-1 zeigt ein unter Einbeziehung eines Warteschlangensystems für Nachrichten und Aufträge dargestelltes Dateimanagersystem. Wir wollen annehmen, daß die zwischen den Prozessen des Systems ausgetauschten Nachrichten, unabhängig davon, ob die Nachrichtenquelle ein Dateimanager oder ein Benutzerprozeß ist, in die gleiche, dem jeweiligen Empfängerprozeß zugeordnete Nachrichtenwarteschlange NQ eingereiht werden.

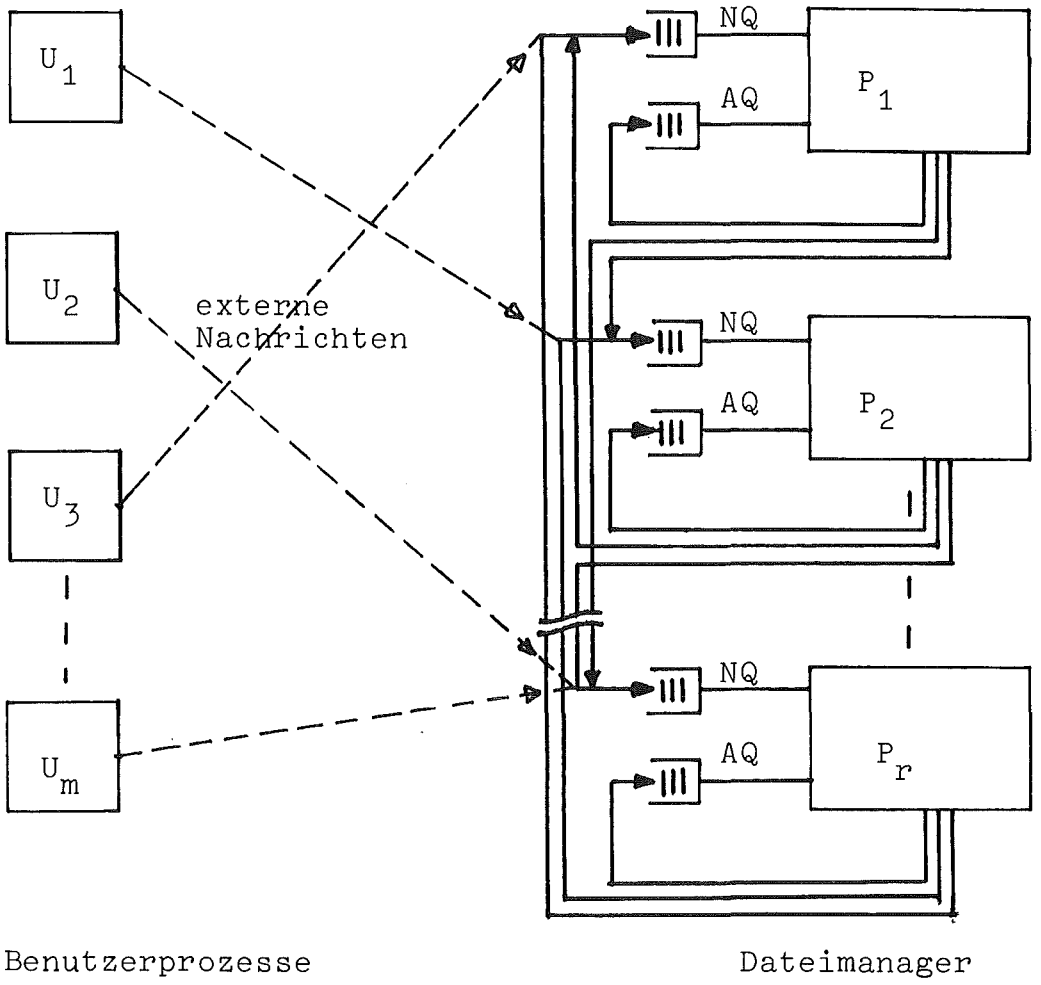


Abb. 4.1-1 Dateimanagersystem mit Nachrichtenwartenschlangen (NQ) und Auftragswarteschlangen (AQ)

Die Nachrichtenhülle (d.h. im allgemeinen der Nachrichtenkopf), die gemäß Kap. 2.3. die Kontrollnachricht darstellt, muß Informationen enthalten, die eine eindeutige Zuordnung zu einem Auftrag ermöglichen, was durch eine Einbettung der Auftragsidentifikation in den Nachrichtenkopf realisiert werden kann. Weitere Informationen in der Kontrollnachricht sollen sich auf den Initiator des Auftrags, d.h. den Dateimanager, der die von einem Benutzerprozeß kommende Anforderung entgegennahm, sowie auf den Nachrichtensender beziehen. Entsprechend sind die Aufträge repräsentierenden Elemente der Auftragswarteschlange durch Identifikationen zu kennzeichnen, wobei gewährleistet sein muß, daß

der zur Kennzeichnung von Aufträgen verfügbare Namensraum eine eindeutige Auftragsidentifizierung zuläßt. Neben den Angaben über Auftragsidentifikation und Auftragsinitiator muß ein Element der Auftragswarteschlange, wie im weiteren deutlich werden wird, Informationen über den augenblicklichen Zustand der Auftragsbearbeitung beinhalten. Der Zustand der Bearbeitung eines Auftrags zur Durchführung eines kritischen Zugriffs in einer Dateimanager-Auftragswarteschlange ist definiert durch den Zustand der Protokolleinheit bei der letzten Unterbrechung der aktiven Bearbeitung des Auftrags und die seit der Unterbrechung empfangenen, sich auf diesen Auftrag beziehenden Kontrollnachrichten.

Die funktionelle Vorgehensweise des Warteschlangenmechanismus (WVM) eines Dateimanagers besteht, wie in Abb. 4.1-2 schematisch dargestellt, in der Abwicklung folgender Aktivitäten:

- Eine Nachricht wird aus der Nachrichtenwarteschlange entnommen und die Nachrichtenidentifikation mit den Identifikationen der in der Auftragswarteschlange abgelegten Elemente verglichen. Ergibt der Vergleich ein negatives Ergebnis, so wird ein Element mit der der Nachrichtenidentifikation entsprechenden Kennung erzeugt und gemäß der dem so akzeptierten neuen Auftrag zuzuordnenden Bearbeitungspriorität (vgl. 3.1.) in die Auftragswarteschlange eingereiht. Eine Einordnung vor Elementen der Auftragswarteschlange, die einen bereits in Bearbeitung befindlichen Auftrag beschreiben, ist dabei nur dann möglich, wenn der zum Zeitpunkt der beabsichtigten Verdrängung vorliegende, im Auftragselement abgespeicherte Bearbeitungszustand dies zuläßt. Der Bearbeitungszustand des an erster Stelle der Auftragswarteschlange befindlichen "aktiven" Auftrags entspricht dem augenblicklichen Zustand der Protokolleinheit (vgl. 3.1.).
- Wurde die aus der Nachrichtenwarteschlange entnommene Nachricht von einem der Dateimanager gesendet, so ist die resultierende Änderung des Bearbeitungszustands des zugehörigen Auftrags im Auftragselement entsprechend zu erfassen. Sich auf den aktiven Auftrag beziehende Nachrichten werden an die

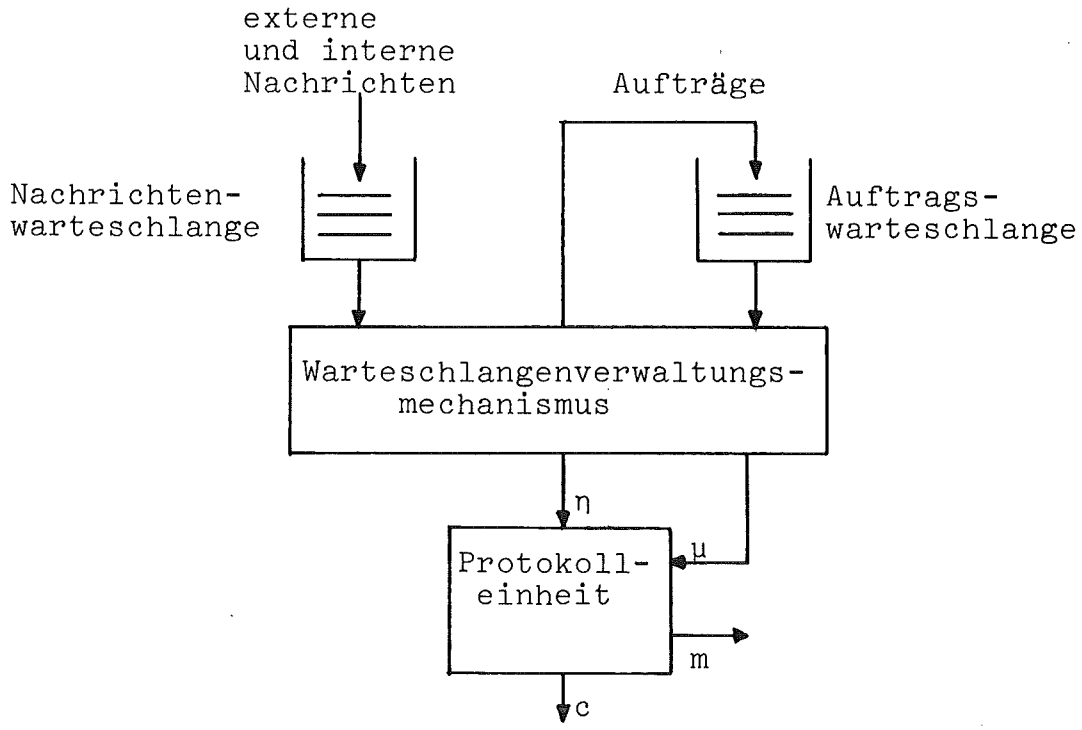


Abb. 4.1-2 Warteschlangenverwaltungsmechanismus eines Dateimanagers

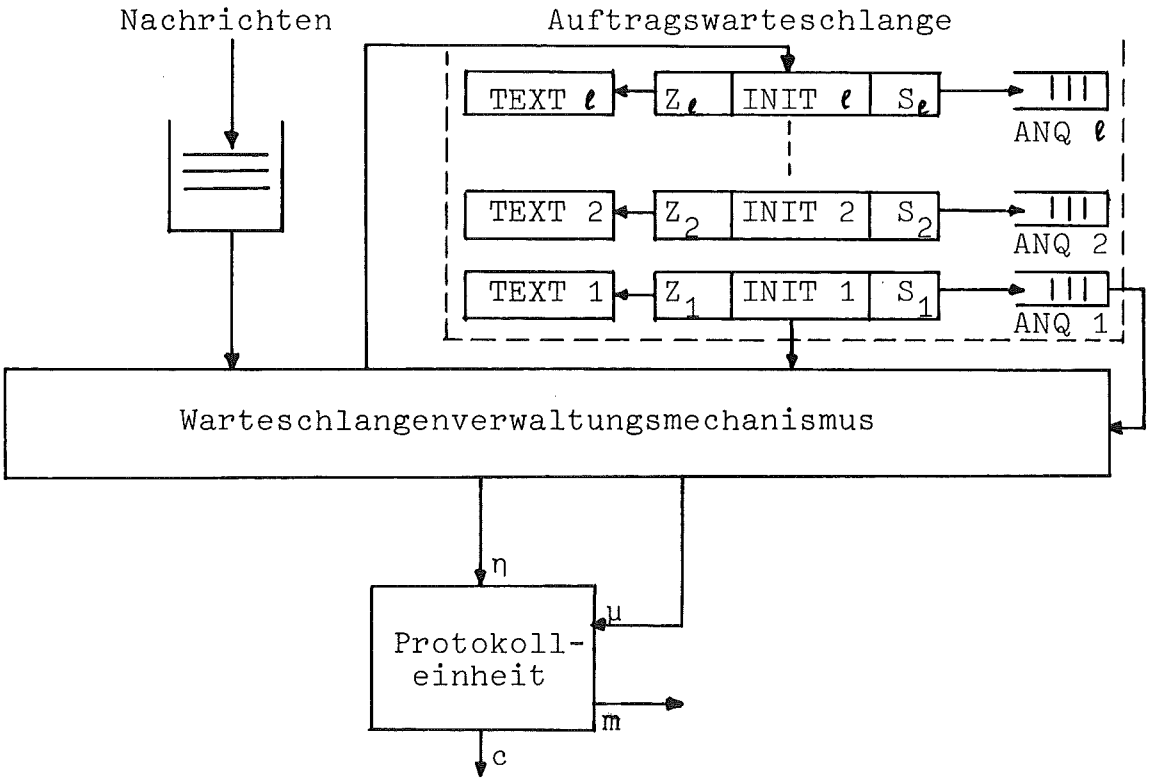


Abb. 4.1-3 Detailstruktur der Auftragswarteschlange eines Dateimanagers. Z_1 entspricht dem in Bearbeitung befindlichen Auftrag. Die ANQ_i sind die den Elementen Z_i assoz. Nachrichtenwarteschlangen.

Protokolleinheit weitergeleitet. Zu Änderungen des Bearbeitungszustandes von Aufträgen, die sich nicht in "aktiver" Bearbeitung durch den Dateimanager bzw. die Protokolleinheit befinden kommt es aufgrund der Verzögerungen beim Transport von Nachrichten über das Kommunikationssystem und bei Neuinitialisierungen von Aufträgen (s.o.). Kommunikationsverzögerungen können bewirken, daß ein Dateimanager Nachrichten empfängt, die sich auf Aufträge beziehen, die aus der aktiven Bearbeitungsphase bereits verdrängt sind.

- Mit der Rückkehr der Protokolleinheit in den Grundzustand erfolgt die Entfernung des jeweils ersten Elementes aus der Auftragswarteschlange. Der dem folgenden Element entsprechende Auftrag wird zum aktiven Auftrag im Dateimanager. Dies geschieht dadurch, daß der im Element abgespeicherte Zustand als aktueller Zustand der Protokolleinheit übernommen wird. Die Protokolleinheit wird dann der Reihe nach mit den bereits aufgelaufenen, ebenfalls über das Element der Auftragswarteschlange zugänglichen sowie mit neuankommenden Nachrichten beaufschlagt. Durch Überholvorgänge im Kommunikationssystem vorzeitig eintreffende Nachrichten (dies gilt speziell für zweistufige Kommunikationsprotokolle) müssen vom WVM bis nach dem Eintreffen aller vorher benötigten Nachrichten zurückgehalten werden.

Aus den oben präzisierten Aufgaben des WVM eines Dateimanagers resultiert die in Abbildung 4.1-3 dargestellte Detailstruktur der Auftragswarteschlange:

Über die mit der Auftragsidentifikation Z_i gekennzeichneten Elemente der Warteschlange sind (etwa durch geeignete Verzeigerung) der Zustand S_i der Protokolleinheit, bezogen auf die Bearbeitung von Z_i sowie ein dem Auftrag zugeordneter privater Nachrichtenpuffer ANQ_i zugänglich, der alle während der nicht-aktiven Phase von Z_i aufgelaufenen oder vorzeitig empfangenen Kontrollnachrichten in einer für die Annahme durch die Protokolleinheit geeigneten Form erfaßt. $Text_i$ entspricht dem ggf. für die Durch-

führung des kritischen Zugriffs benötigten Text, der beim Übergang der Dateimanager in den kritischen Zustand bereitstehen muß. $INIT_i$ kennzeichnet den Initiator des Auftrags. (Ein neu erzeugtes Element kann bei Verwendung einstufiger Protokolle nicht vor einem Element mit gleichem Initiator in der Auftragswarteschlange eingeordnet werden - vgl. 3.1.) Die über die Elemente der Auftragswarteschlange zugänglichen Informationen werden zweckmäßig in einem Auftragskontrollblock zusammengefaßt.

Im Zusammenhang mit der Detailbeschreibung der Dateimanagerfunktionen sei abschließend auf eine Analogie zwischen dem Aufbau der generellen Ablauforganisation von DV-Systemen und dem der Dateimanagerorganisation hingewiesen:

Der Warteschlangenverwaltungsmechanismus des Dateimanagers hat die Aufgabe, die Vergabe der Protokolleinheit an zu bearbeitende Aufträge zwecks Durchführung kritischer Zugriffe vorzunehmen, wobei diese Zuordnung von Auftrag und Protokolleinheit vom "Dispatcher" /36/, dem WVM, unter Anwendung preemptiver Maßnahmen, realisiert werden kann. Auslöser der "Dispatcher"-Funktionen des WVM sind Initialisierungen durch die Protokolleinheit (entsprechend dem Aufruf von Dispatcherfunktionen), wenn diese die Eingabe den Zustand fortschaltender Kontrollnachrichten erwartet. Dabei wird der "Dispatcher" solange neu aktiviert, bis eine sich auf den aktiven Auftrag beziehende Kontrollnachricht eintrifft oder aber die Verdrängung des aktiven Auftrags durch einen Auftrag höherer Priorität zustande kommt.

4.2. Grundtypen und Varianten der Koordinationsverfahren

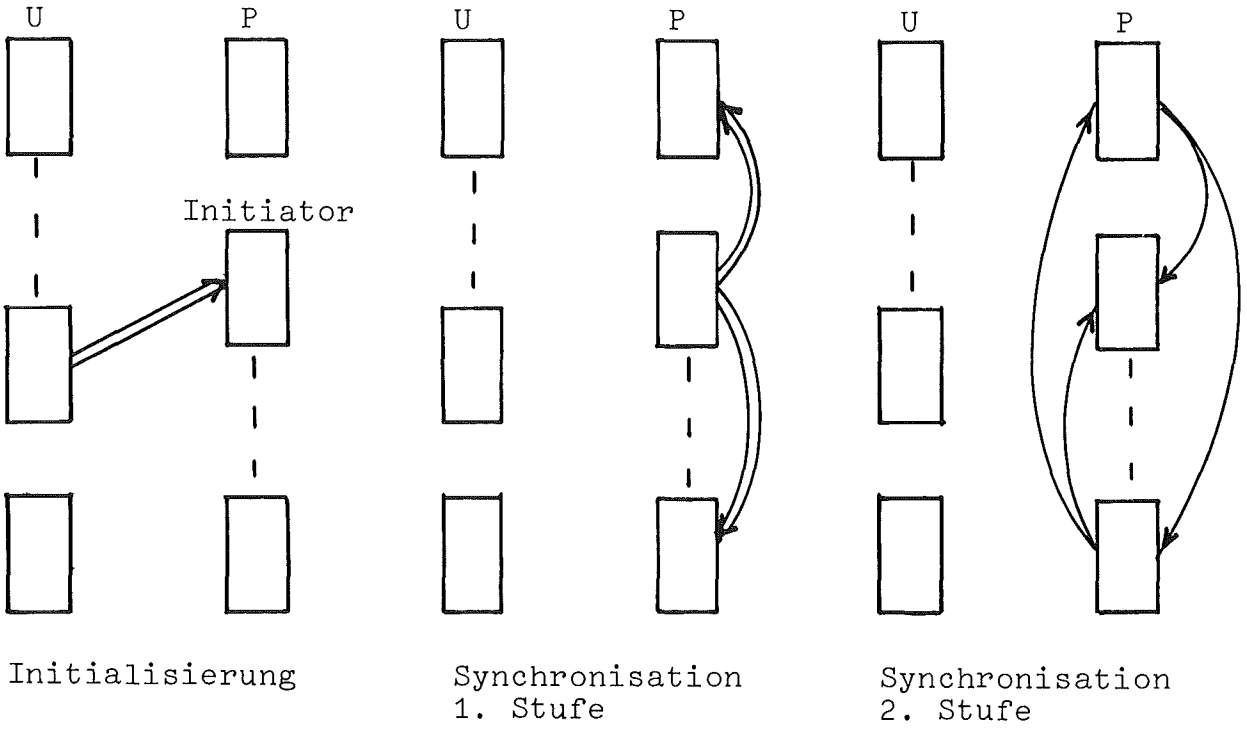
In Kapitel 3 wurden zwei Grundtypen von Koordinationsprotokollen eingeführt, einstufige und zweistufige Protokolle. Während der einstufige Typ ein elementares Synchronisationsmittel zur Durchführung kritischer Zugriffe ohne Berücksichtigung einer gegebenenfalls einzuhaltenden Reihenfolge verkörpert, erlaubt der zweistufige Protokolltyp durch Einführung eines adaptierbaren Karenzzeitintervalles die Abarbeitung von Zugriffsaufträgen in

Konsistenz erhaltender Reihenfolge, sofern die in (3.1-5) spezifizierte Bedingung erfüllt ist. Prinzipiell ist unter Einhaltung der in 3.2. formulierten Voraussetzungen der Aufbau von Koordinationsprotokollen mit einer beliebigen Zahl von Synchronisationsstufen möglich, ohne daß jedoch den ins Auge gefaßten Anwendungen eine Begründung für die Notwendigkeit derartiger Protokolle zu entnehmen wäre. Wir wollen uns daher auf die bisher erörterten Protokolltypen beschränken.

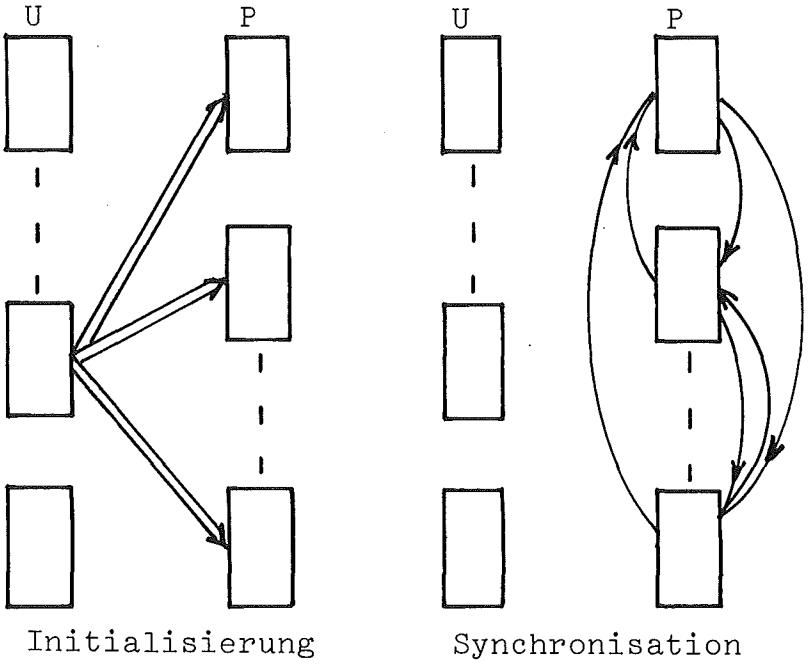
Beim Ausbau einstufiger und zweistufiger Koordinationsprotokolle zu Koordinationsverfahren unter Einbeziehung der im vorangegangenen Kapitel präzisierten Dateimanagerfunktionen lassen sich auf den beiden Protokolltypen eine Reihe von Varianten von Koordinationsverfahren aufbauen. Zwei Variationsmöglichkeiten der Dateimanagerfunktionen seien hier besonders hervorgehoben:

a) FIFO oder prioritätsorientierte Abarbeitung der Nachrichtenwarteschlange
   Die Abarbeitung der Nachrichtenwarteschlange kann auf zweierlei Arten erfolgen: Abarbeitung nach FIFO und prioritätsorientierte Abarbeitung. Die prioritätsorientierte Abarbeitung bietet dann Vorteile, wenn die Bearbeitungsreihenfolge für Zugriffsaufträge möglichst nach Prioritäten gestaffelt vorzunehmen ist (vgl. 3.1.).

b) Einfach- oder Mehrfachinitialisierung von kritischen Zugriffen
   Die Initialisierung von Zugriffsaufträgen beim Überwachungssystem, d.h. bei der Gesamtheit der betroffenen Dateimanager, kann von den Benutzerprozessen aus zum einen durch Übermittlung einer entsprechenden Anforderung an nur einen aus der Menge der Dateimanager, etwa den nächstgelegenen, erfolgen (Einfachinitialisierung), zum anderen durch parallele Übermittlung von Anforderungen an mehrere Dateimanager (Mehrfachinitialisierung), im Extremfall an alle Dateimanager gleichzeitig. Die beiden Initialisierungsverfahren sind in Abb. 4.2-1 skizziert. Sowohl bei der Einfachinitialisierung



a) Einfachinitialisierung



b) Mehrfachinitialisierung

Abb. 4.2-1 Varianten der Initialisierung kritischer Zugriffe

U = Benutzerprozesse, P = Dateimanagerprozesse

⟹ Transport von Kontrollinformation + Änderungstext

⟶ Transport von Kontrollinformation ohne Änderungstext

als auch bei der Mehrfachinitialisierung muß dafür gesorgt werden, daß alle zur Zugriffsausführung benötigten Informationen bei Eintritt der Dateimanagerprozesse in die kritische Phase am Ort der betroffenen Datenbankkomponenten verfügbar sind (andernfalls kommt es lokal zu Verzögerungen der Zugriffsdurchführung).

Die Vorteile der Mehrfachinitialisierung gegenüber der Einfachinitialisierung liegen in der möglichen Verkürzung der Synchronisationsphase, wie auch anhand des Schemas 3.1-3 ersichtlich wird; darüber hinaus stellt die Mehrfachinitialisierung ein Instrument zur Steigerung der Zuverlässigkeit des Überwachungssystems dar (vgl. 3.3.). Die Nachteile dieser Variante liegen im aufwendigeren Nachrichtentransport zwischen Benutzerprozessen und Dateimanagern.

Wie ein Vergleich der beiden Alternativen in Abb. 4.2-1 zeigt, kann es bei der Mehrfachinitialisierung früher als bei der Einfachinitialisierung zum Eintritt der Dateimanager in die autonome Phase kommen, die durch die individuelle Verantwortlichkeit der Dateimanager für die Durchführung des kritischen Zugriffs gekennzeichnet ist. Die Voraussetzungen dafür sind gegeben, wenn für die Verzögerungszeitdifferenz ΔT_{UP} bei der Übertragung der Anforderungen für den gleichen Zugriff von einem Benutzerprozeß U zu den r Dateimanagerprozessen P_i gilt:

$$\Delta T_{UP} = \max_{i\in I} T_{UP_i} - \min_{i\in I} T_{UP_i} \leq T_{P_l P_k} \qquad (4.2\text{-}1)$$

$$\text{mit } l\neq k,\quad l,k\in I \qquad I=\{1,\ldots,r\}$$

T_{UP_i} steht für die Übertragungszeit des Zugriffsauftrags von einem Benutzerprozeß zu den Dateimanagerprozessen, $T_{P_l P_k}$ ist die für die Übertragung von Kontrollnachrichten zwischen Dateimanagerprozessen benötigte Zeit.

Für den Fall, daß in einem System mit vereinbarter Mehrfachinitialisierung Bedingung (4.2-1) nicht erfüllt ist, ergibt

sich automatisch, gemäß den in Schema 3.1-3 und 3.1-4 vereinbarten Protokollen, eine Kombination von Einfach- und Mehrfachinitialisierung: Die Dateimanager, die die Anforderung eines Benutzerprozesses zur Durchführung eines bestimmten kritischen Zugriffs zuerst erreicht, fungieren als Initiatoren und stoßen die Synchronisationsprozedur an. Dabei ergibt sich jedoch folgende Schwierigkeit:

Sei P_i, $i\varepsilon\{1,\ldots,r\}$ ein Dateimanager, der als erster in einem Überwachungssystem mit vereinbarter Mehrfachinitialisierung zur Durchführung eines bestimmten Zugriffs initialisiert wird. P_i reagiert auf diese Initialisierung mit der Aussendung von Kontrollnachrichten des Typs A an alle übrigen Dateimanager P_k, $k\neq i$, $k\varepsilon\{1,\ldots,r\}$. Sei $T_{P_iP_l}$ die Zeit, die vergeht bis P_l, $l\neq i$, $l\varepsilon\{1,\ldots,r\}$ die von P_i gesendete Nachricht empfängt. Im Falle, daß

$$\Delta T_{UP} < T_{P_iP_l} \quad , \tag{4.2-2}$$

ist bei Dateimanager P_l zum Zeitpunkt des Eintreffens der Kontrollnachricht vom Typ A bereits ein entsprechendes Element in die Auftragswarteschlange von P_l eingereiht, bedingt durch die zwischenzeitliche Ankunft der Anforderung des Benutzerprozesses. Demgemäß darf der Empfang der Kontrollnachricht vom Typ A, im Gegensatz zur Einfachinitialisierung, hier nicht zur Erzeugung eines neuen Auftragswarteschlangen-Elementes führen. Es muß jedoch die Positionierung des aufgrund der externen Anforderung erzeugten Elementes in der Warteschlange überprüft und gegebenenfalls unter Berücksichtigung der durch das Protokoll gegebenen Regelung korrigiert werden. Dies sei mit Hilfe des in Abb. 4.2-2 wiedergegebenen Beispiels erläutert:

Wir wollen annehmen, daß in einem auf der Basis der Mehrfachinitialisierung arbeitenden Zwei-Dateimanager-System zur Zeit t' bei Dateimanager P_1 die Anforderung Z_m und bei P_2 die Anforderung Z_{m+1} eintreffen. Die Prioritäten seien so verteilt, daß

$$\text{Priorität}(Z_m) > \text{Priorität}(Z_{m+1}) \tag{4.2-3}$$

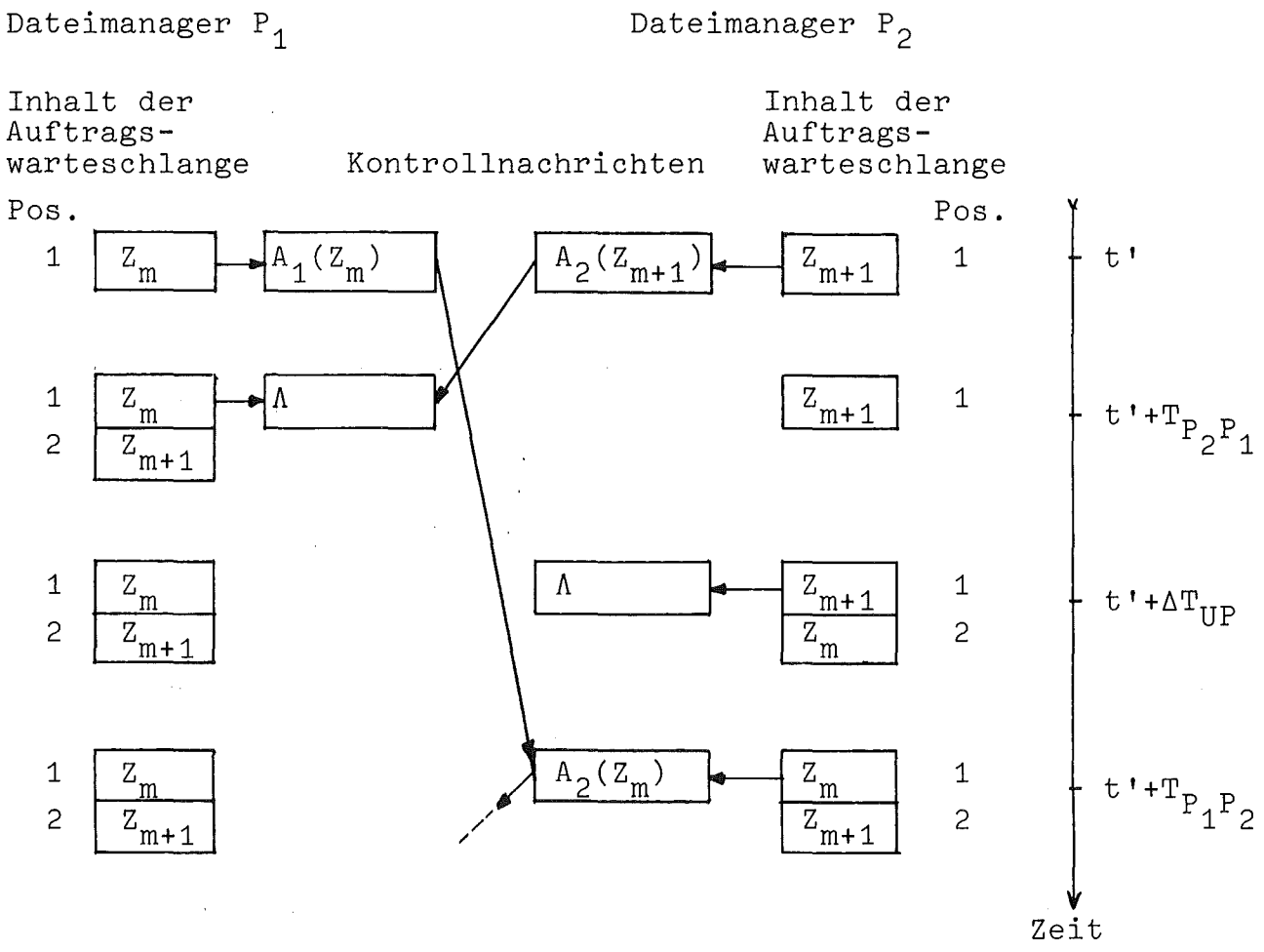


Abb. 4.2-2 Umorganisation der Auftragswarteschlangen bei Mehrfachinitialisierung in einem System mit Dateimanagern P_1 , P_2

Unter der Annahme, daß beide Auftragswarteschlangen leer sind, führt die Entgegennahme der Aufträge bei P_1 und P_2 zur Produktion der Kontrollnachrichten $A_1(Z_m)$ bzw. $A_2(Z_{m+1})$. Es sei weiterhin angenommen, daß

(4.2-4) $$T_{P_1P_2} > \Delta T_{UP} > T_{P_2P_1}$$

Aus (4.2-3) und (4.2-4) folgt, daß zum Zeitpunkt $t'+\Delta T_{UP}$ die Reihenfolge der ersten beiden Elemente der Auftragswarteschlange von P_2 der Umkehrung der Reihenfolge der ersten zwei Elemente der Auftragswarteschlange von P_1 entspricht. Das Eintreffen von Kontrollnachricht $A_1(Z_m)$ zum Zeitpunkt $t'+T_{P_1P_2}$ bei P_2 muß daher zur Umorganisation der Warteschlange bei P_2 führen, entsprechend einer Neuerzeugung und Einordnung eines Elementes Z_m an erster Stelle der Auftragswarteschlange von P_2, Übertragung des Zustandes aus dem bereits vorhandenen in das neue Element und Entfernung des alten Elementes Z_m.

Es sei an dieser Stelle darauf hingewiesen, daß die Anwendung der Mehrfachinitialisierung lediglich eine entsprechende Berücksichtigung beim organisatorischen Aufbau des Warteschlangenverwaltungsmechanismus der Dateimanager finden muß. Das die Wechselwirkungen zwischen den Protokolleinheiten festlegende Koordinationsprotokoll bleibt davon unbeeinflußt.

Die in diesem Kapitel erläuterten Variationsmöglichkeiten können beim Aufbau von Koordinationsverfahren kombiniert berücksichtigt werden. Insgesamt ergeben sich durch die Heranziehung von jeweils einer Alternative aus 4.2a) und 4.2b) und deren Kombination mit den beiden Koordinationsprotokoll-Grundtypen 8 Varianten von Koordinationsverfahren, die in Tabelle 4.2-1 zusammengestellt sind.

Kurzbezeichnung der Variante	Art der Initialisierung	Protokolltyp	Art der Nachrichtenabarbeitung
EEF	einfach	einstufig	FIFO
EEP	einfach	einstufig	Priorität
EZF	einfach	zweistufig	FIFO
EZP	einfach	zweistufig	Priorität
MEF	mehrfach	einstufig	FIFO
MEP	mehrfach	einstufig	Priorität
MZF	mehrfach	zweistufig	FIFO
MZP	mehrfach	zweistufig	Priorität

Tabelle 4.2-1 Varianten der Koordinationsverfahren

5. Untersuchung des operativen Verhaltens durch Simulation

5.1. Zielsetzung der Simulationsexperimente

Die entwickelten Koordinationsprotokolle sowie die darauf aufbauenden alternativen Koordinationsverfahren wurden im vorangegangenen ohne Berücksichtigung der zu erwartenden Leistungsfähigkeit diskutiert. Im Zusammenhang mit dem Entwurf von Alternativen sind Vergleiche von Leistungsmerkmalen von Interesse, vor allem um die günstigsten Einsatzbereiche der Varianten in Abhängigkeit von den beeinflussenden Parametern zu ermitteln.

Um Vergleiche vornehmen zu können ohne kostspielige Implementierungsexperimente durchführen zu müssen, bietet sich im Falle des zu erstellenden Überwachungssystems der Aufbau eines das Dateimanagersystem beschreibenden Warteschlangenmodells an, wie in Abb. 5.1-1 skizziert. Modelle dieser Art stellen eine geeignete Ausgangsbasis dar zur Beurteilung operativer Charakteristiken auftragsbearbeitender Systeme,wie Verteilungen der Auftragsbearbeitungsdauer, Auftragswartezeiten, Warteschlangenlängen und Verweilzeiten in Abhängigkeit vom Auftragsprofil (Zusammensetzung des in das System gelangenden Auftragsstromes), von statistischen Eigenschaften des Ankunftsstromes und von Systemparametern, die die Auslegung des zu untersuchenden Systems beschreiben.

Angewandt auf Untersuchung und Vergleich alternativer Verfahren zur Koordination kritischer Zugriffe kann die Untersuchung eines Dateimanagersystems als Warteschlangenmodell Aufschlüsse über die Statistik folgender Systemeigenschaften bieten:

- Dauer der Koordination und Durchführung kritischer Zugriffe
- Wartezeiten bis zur Durchführung eines angemeldeten kritischen Zugriffs
- Längen von Nachrichten- und Auftragswarteschlangen
- Grad der Konsistenzherstellung bei Auftragsströmen mit gestörter Reihenfolge

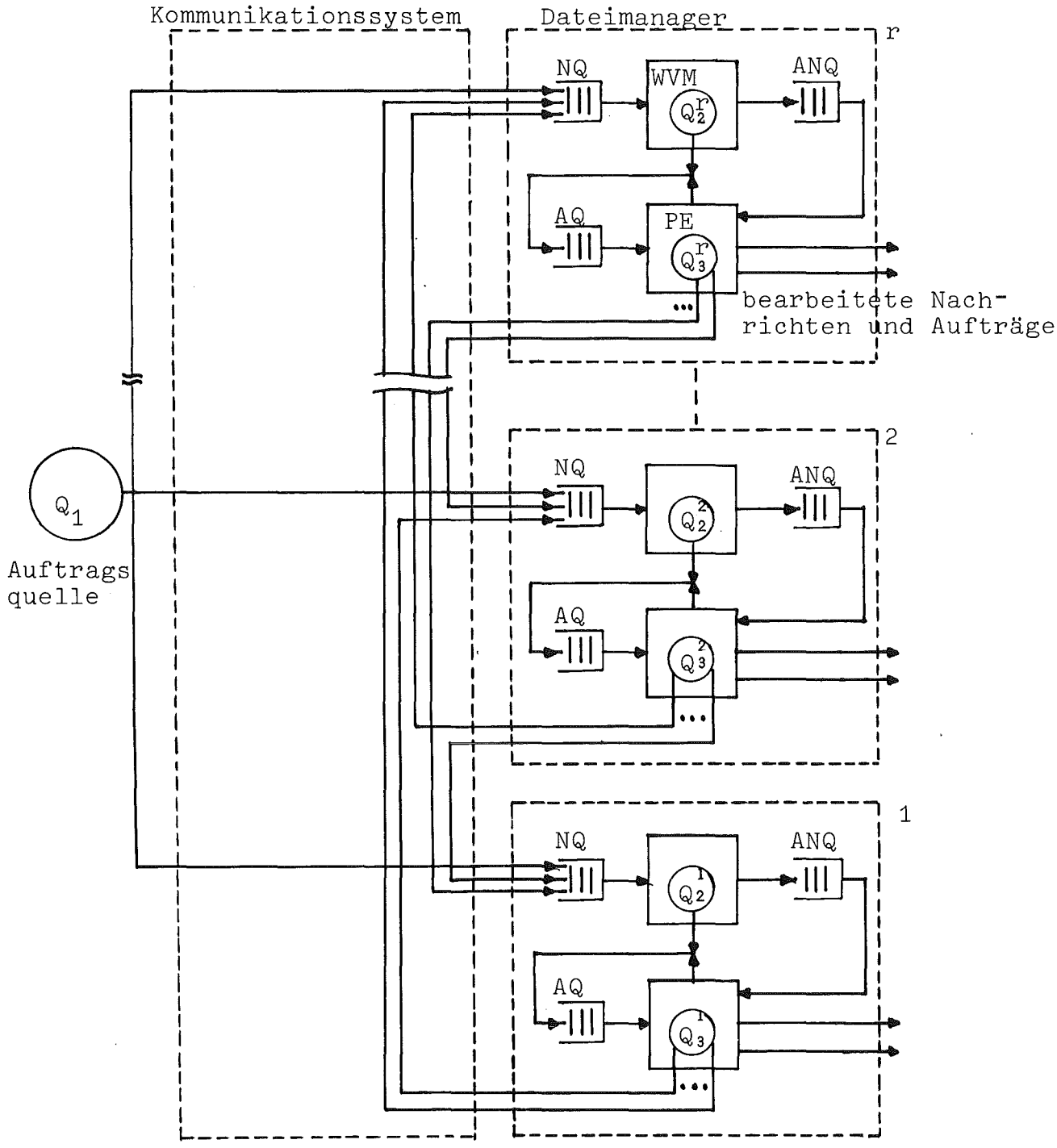


Abb. 5.1-1 Warteschlangenmodell eines Überwachungssystems mit r Dateimanagern. NQ = Nachrichtenwarteschlange, ANQ = assoziierte Nachrichtenwarteschlange, PE = Protokolleinheit, WVM = Warteschlangenverwaltungsmechanismus, Q_2^i = Dateimanager-interne Erzeugung von Auftragselementen, Q_3^i = interne Nachrichtenquelle

- maximale Differenz zwischen den Zeitpunkten des Eintritts der Dateimanager in den kritischen Abschnitt zur Durchführung des gleichen kritischen Zugriffs

Abbildung 5.1-1 zeigt ein r-Dateimanager-Überwachungssystem als ein Netz von Bedienungsstationen. Bedienungsnetze sind jedoch generell einer analytischen Behandlung nur sehr schwer und nur unter speziellen Voraussetzungen zugänglich:

- Die zu einem Netz zusammengeschlossenen Wartesysteme müssen einzeln analysiert werden können.
- Sollen globale Bedienungscharakteristika, wie etwa die höheren Momente der Gesamtbearbeitungsdauer oder der Gesamtwartezeit ermittelt werden, so müssen die zu den globalen Größen beitragenden Größen der einzelnen Wartesysteme unabhängig voneinander sein. Dies ist im allgemeinen jedoch nicht der Fall /3/.

Damit in einem Netz von Bedienungsstationen die einzelnen Teilsysteme analysiert werden können, müssen die Eingangsströme dieser Teilsysteme, die ganz oder teilweise wieder Ausgangsströme anderer Systeme sind, rekurrent sein. Dies ist nur in Spezialfällen, wie etwa bei Netzen aus Bedienungssystemen vom Typ M/M/k mit Poisson Eingabe und exponentiell verteilter Bedienungszeit erfüllt (vgl. auch /25/,/28/,/34/,/39/).

Für das in Abb. 5.1-1 wiedergegebene Modell kann zwar in guter Übereinstimmung mit der Realität für bestimmte Formen des Auftragsankunftsstromes Poisson-Charakteristik angenommen werden; die Annahme einer exponentialverteilten Bedienungszeit würde jedoch zu einer starken Vereinfachung führen, deren Auswirkungen bei der Komplexität des hier vorliegenden Modells nicht abzuschätzen sind. Zudem ist die zweite der oben aufgeführten Voraussetzungen auch bei Approximation der Bedienungsstationen innerhalb des Dateimanagersystems durch Wartesysteme vom Typ M/M/k, (k>1), nicht zu erfüllen.

Da die Anwendung von Methoden der Warteschlangentheorie zur Ermittlung charakteristischer Leistungsdaten auf analytischem Wege für die zu untersuchenden Koordinationsverfahren (ausgehend von den derzeit verfügbaren Grundlagen) nicht zum Ziel führt, bleibt im wesentlichen nur die empirische Bestimmung der Verteilungen bzw. der Momente der interessierenden Variablen mit Hilfe von Simulationsexperimenten. Zum Aufbau und zur experimentellen Untersuchung von Warteschlangensystemen eignen sich vornehmlich die Methoden der diskreten Simulation.

Um vergleichende Untersuchungen der in 4.2. vorgeschlagenen Varianten vornehmen zu können, wurde das in Abb. 5.1-1 wiedergegebene Warteschlangenmodell eines Dateimanagersystems als Simulationsmodell unter Verwendung der Programmiersprache SIMULA /12/ implementiert. Für die Wahl von SIMULA als Simulationssprache war dabei von den in /38/ dargelegten Gründen vor allem die mit der Anwendung des in SIMULA realisierten Klassenkonzepts verbundene leichte Rekonfigurierbarkeit von Modellen maßgebend. Sie erlaubte einen flexiblen, durch Parameter steuerbaren Modellaufbau, der die Vergleiche der Verfahrensvarianten bei wechselnden Systemkonfigurationen wesentlich erleichterte.

5.2. Aufbau des Simulationsmodells

Es soll zunächst versucht werden, soweit wie möglich, den prinzipiell von den geplanten Experimenten unabhängigen Aufbau des Simulationsmodells gesondert vom Experimententwurf zu behandeln. Details des Modellaufbaus, die dennoch durch Berücksichtigung von Randbedingungen der konzipierten Experimente beeinflußt wurden, sollen im Rahmen der Diskussion des Experimententwurfs abgehandelt werden.

Das in SIMULA erstellte Simulationsmodell zur Untersuchung des operativen Verhaltens der 8 verschiedenen, in 4.2. erläuterten Varianten von Koordinationsverfahren, setzt sich aus 5 verschiedenen Komponenten unterschiedlicher Komplexität zusammen (vgl. Abb. 5.2-1): Den drei "permanenten" Komponenten Auftragsgenerator, Kommunikationssystem und Dateimanagersystem sowie aus zwei temporären, in der Anzahl ihrer Realisierungen nicht begrenzten Komponenten, den Nachrichten und Auftragselementen.

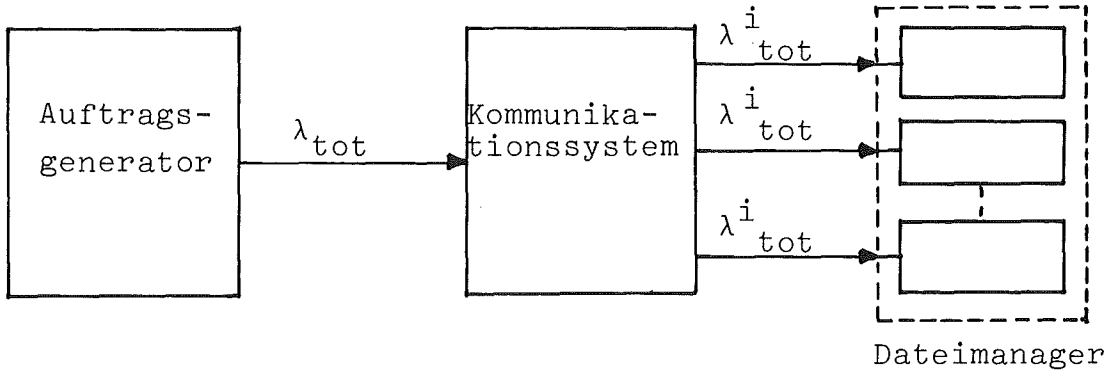


Abb. 5.2-1 Komponenten des Simulationsmodells

Die Aufgaben sowie die Möglichkeiten der Beeinflussung der Arbeitsweise der permanenten Komponenten sind dabei im einzelnen wie folgt festgelegt:

Auftragsgenerator:
Der Auftragsgenerator nimmt die Rolle der in Abb. 5.1-1 dargestellten Auftragsquelle Q_1 ein und tritt damit an die Stelle der Gesamtheit temporärer Benutzerprozesse. Seine Aufgabe besteht in der Erzeugung eines Auftragsstromes, der sich aus kritischen und nichtkritischen Zugriffsanforderungen auf die vom Dateimanagersystem kontrollierte Datenbank zusammensetzt.

Der Auftragsstrom stellt einen Strom von Nachrichten an das Überwachungssystem dar. Die in diesen Nachrichten enthaltenen Informationen beinhalten:

- die Auftragsidentifikation (d.h. eine sequentielle Durchnumerierung der Aufträge in der Reihenfolge ihrer Erzeugung),
- eine Kennzeichnung des Auftragstyps (kritisch - nicht kritisch),
- die Adresse (Identifikation) des Dateimanagers an den sich der Auftrag wendet,
- eine Angabe über den Zeitpunkt der Auftragsgenerierung.

Die Auftragsbearbeitungspriorität wird durch die Auftragsidentifikation bestimmt: je früher der Erzeugungszeitpunkt, desto höher die Priorität. Gleiche Auftragsprioritäten sind durch diese Festlegung verhindert (vgl. 3.1.).

Bei einer Erzeugungsrate λ kritischer Aufträge und einem Verhältnis γ der Zahl der kritischen zu der Zahl der unkritischen Aufträge ergibt sich eine Gesamtankunftsstromrate λ_{tot} von

$$\lambda_{tot} = (1+\frac{1}{\gamma}) \cdot \lambda \qquad (5.2\text{-}1)$$

Im Falle der Einzelinitialisierung wird der Strom der kritischen Aufträge gleichmäßig (dies setzt eine optimale Verteilung der Datenbankkomponenten im Netz voraus) auf die r vorhandenen Dateimanager verteilt, wodurch sich partielle Auftragsankunftsströme λ^i bzw. λ^i_{tot}

$$(5.2\text{-}2) \qquad \lambda^i = \frac{1}{r}\lambda \qquad \lambda^i_{tot} = \frac{1}{r}\lambda\left(1+\frac{1}{\gamma}\right) \qquad i=1(1)r$$

ergeben. Bei Mehrfachinitialisierung gilt dagegen

$$(5.2\text{-}3) \qquad \lambda^i = \lambda \qquad \lambda^i_{tot} = \lambda + \frac{1}{r}\,\frac{\lambda}{\gamma}$$

Für den vom Auftragsgenerator erzeugten Auftragsankunftsstrom wird generell angenommen, daß er Poisson-Charakteristik besitzt. Die Operation des Auftragsgenerators wird im Simulationsmodell durch die Modellparameter

- Ankunftsrate des Auftragsstroms λ_{tot}
- Verhältnis der Zahl kritischer zur Zahl unkritischer Zugriffe γ
- Mehrfachinitialisierung - Einfachinitialisierung

festgelegt.

Kommunikationssystem:
Entsprechend seiner Aufgabe in realen Systemen stellt das Kommunikationssystem das Transportmedium zur Übertragung von Nachrichten zwischen Auftragsgenerator und Dateimanagern sowie zwischen den Dateimanagern dar. Auf eine Detailmodellierung der in einem Kommunikationssystem ablaufenden Vorgänge, die den Aufbau dieses Teilmodells als Netz von Bedienungsstationen erforderlich machen würde (vgl. /39/), kann verzichtet werden, da im Zusammenhang mit den beabsichtigten Untersuchungen der Leistungsmerkmale von Koordinationsverfahren lediglich die im Kommunikationssystem erfahrene Gesamtverzögerung der Nachrichten relevant ist. Dazu bietet sich die Modellierung des Kommunikationssystems durch ein Bedienungssystem an, welches die parallele Durchführung einer beliebigen Zahl von Nachrichtentransporten

erlaubt, wobei die resultierenden Verzögerungen durch Ziehung von Zufallszahlen nach vorgebbaren Verteilungen ermittelt werden. Die Parameter der Verteilungen werden dabei in Abhängigkeit vom Typ der zu übertragenden Nachricht festgelegt.

Im vorliegenden Simulationsmodell wird die individuelle Übertragungsverzögerung, von der jede Nachricht bei der Übertragung über das Kommunikationssystem betroffen wird, bereits bei Erzeugung der Nachricht bestimmt. Als Aufgabe des Kommunikationssystems bleibt dann lediglich, nach Ablauf der Verzögerungszeit - gemessen vom Zeitpunkt der Aktivierung des Kommunikationssystems durch die die Nachricht erzeugende Systemkomponente - die Nachricht in die Nachrichtenwarteschlange des Empfängers einzureihen.

Dateimanagersystem:
Diese Komponente, die wichtigste des Simulationsmodells, setzt sich aus einer variablen Zahl von identisch aufgebauten Dateimanagern zusammen. Der einzelne Dateimanager als Teilmodell entspricht in seinem organisatorischen Aufbau dem in Abb. 4.1-3 gezeigten Schema: wir finden

- einen Warteschlangenverwaltungsmechanismus, der für die Entgegennahme der Nachrichten vom Auftragsgenerator bzw. von anderen Dateimanagern sowie für die Erzeugung von Auftragselementen verantwortlich ist,
- einen der Protokolleinheit äquivalenten Steuerungsteil, von dem aus der WVM aktiviert wird und der seinerseits wieder die Aussendung von Kontrollnachrichten an die restlichen Dateimanager veranlaßt.

Für den WVM sind zwei Alternativmodelle verfügbar, entsprechend der unterschiedlichen Handhabung von Auftragswarteschlangenelementen bei Einfach- und bei Mehrfachinitialisierung (vgl. 4.2.).

Die Abarbeitung der vom WVM nach Auftragsprioritäten unter Berücksichtigung der Verdrängungsmöglichkeiten (vgl. 3.1., 4.1.) erstellten Auftragswarteschlange durch die Protokolleinheit erfolgt derart, daß stets der an erster Position stehende Auf-

trag zum aktiven Auftrag wird, unabhängig davon, ob es sich um einen kritischen oder nicht-kritischen Auftrag handelt. Die für die eigentliche Zugriffsausführung (Lese- und/oder Schreiboperationen auf Sekundärspeicher) benötigte Zeit kann für kritische und nicht-kritische Zugriffe als Zufallszahl aus getrennten Verteilungen gezogen werden /43/.

Vergleichende Experimente mit der einstufigen und zweistufigen Variante des Koordinationsprotokolls werden durch das Dateimanagermodell ermöglicht, ohne daß die Realisierung beider Alternativen durch separate Teilmodelle erforderlich ist, da das zweistufige Protokoll die Elemente des einstufigen beinhaltet (vgl. 3.1.).

Die Arbeitsweise des Dateimanagersystems wird durch folgende Modellparameter bestimmt:

- Art der Auftragsinitialisierung (Einfach- oder Mehrfachinitialisierung)
- Protokolltyp (einstufig - zweistufig)
- Zahl der Dateimanager r
- Verteilungen der Zugriffsausführungszeiten

Die permanenten Modellkomponenten Auftragsgenerator, Kommunikationssystem und Dateimanagersystem (d.h. die einzelnen Dateimanager) werden im Simulationsmodell einheitlich durch Prozeßobjekte (Prozeß steht hier im Sinne des Prozeßbegriffs in SIMULA, vgl. /12/) realisiert. Die Interaktionen zwischen Auftragsgenerator und Dateimanagern sowie zwischen den Dateimanagern selbst sind nur auf dem Umweg über die Aktivierung des Prozeßobjekts Kommunikationssystem unter Aufruf einer dafür vorgesehenen Prozedur - entsprechend dem Aufruf einer Elementarfunktion sende Nachricht (vgl. 2.2.) - möglich.

Im Gegensatz zu den permanenten Modellkomponenten sind die temporären Komponenten des Simulationsmodells nicht als Prozeßobjekte realisiert. Vielmehr werden Nachrichten und Auftragselemente als Datenstrukturen bei Bedarf von den permanenten Komponenten erzeugt, manipuliert und wieder vernichtet. Der Aufbau der in den temporären Komponenten abgelegten In-

formationen sei hier kurz skizziert:

Nachrichten:
Bei den Nachrichten, dem Aufbau nach stets gleich strukturiert, sind zwei Typenklassen zu unterscheiden:

- externe, vom Auftragsgenerator erzeugte Nachrichten,
- interne, für die Inter-Dateimanager-Kommunikation verwendete Nachrichten.

Der Unterschied liegt in der für die Ermittlung der Kommunikationsverzögerung herangezogenen Verteilung, die für beide Klassen über entsprechende Modellparameter getrennt angebbar ist und so z.B. die privilegierte Bedienung interner Nachrichten im Kommunikationssystem zu berücksichtigen gestattet. Die Ziehung der Verzögerungszeit wird bei der Erzeugung eines Nachrichtenobjektes vorgenommen, was zulässig ist, da das Kommunikationssystem im Modell mit jeder Nachricht nur einmal beaufschlagt wird.

Bei den externen Nachrichten werden zwei Typen unterschieden: Nachrichten zur Beschreibung kritischer Anforderungen und Nachrichten zur Beschreibung unkritischer Anforderungen. Die in den externen Nachrichten enthaltenen Informationen wurden bereits bei der Erläuterung der Auftragsgeneratorfunktionen beschrieben.

Bei den internen Nachrichten gibt es drei verschiedene Typen, entsprechend den aus der Beschreibung des zweistufigen Protokolls (vgl. 3.1.) als aufwendigster Protokollvariante zu entnehmenden Erfordernissen: Die Typen A,B und E. Neben der Kennzeichnung des Nachrichtentyps umfassen interne Nachrichten sämtliche Informationen, die zur Zuordnung der Nachrichten zu den bezogenen Aufträgen erforderlich sind.

Auftragselemente:
Auftragselemente dienen der Beschreibung der vom Dateimanagersystem zu bedienenden Aufträge und werden sowohl für kritische als auch unkritische Zugriffsaufträge erstellt. Unkritische Aufträge werden, da sie lokal abgewickelt werden, jeweils nur durch ein einziges Element im gesamten Dateimanagersystem repräsentiert, während kritische Aufträge in jeder Auftragswarte-

schlange des Dateimanagersystems durch je ein Element dargestellt sind.

Der Aufbau der Elemente ist für beide Auftragstypen gleichartig; die Datenstruktur umfaßt neben der Kennzeichnung des Auftragstyps (kritisch/unkritisch) die in 4.1. detailliert beschriebenen Informationen, die, soweit angebracht, auch auf die Beschreibung unkritischer Aufträge übertragen werden. Daneben enthalten die Auftragselemente alle zur Erfassung der Bedienungscharakteristika des Dateimanagersystems erforderlichen Zeitangaben, wie

- Generierungszeitpunkt,
- Ankunftszeit beim Dateimanager,
- Zeitpunkt des Bearbeitungsbeginns,
- Ende der Bearbeitung.

5.3. Experimententwurf

Beim nun zu diskutierenden Experimententwurf bestehen die wesentlichen Aufgaben in der Festlegung der zu messenden Größen und der Art ihrer Ermittlung, in der Auswahl der zu variierenden Modellparameter (factorial design /33/) und der Vereinbarung geeigneter Wertebereiche für diese Parameter, sofern es sich um kontinuierlich veränderbare Parameter - quantitative Faktoren - handelt, im Gegensatz zu qualitativen Faktoren mit diskretem Wertebereich.

Die zu untersuchenden Größen wurden bereits in 5.1. erörtert; zur Durchführung der entsprechenden Messungen werden für alle Experimente an einem der Dateimanager des Überwachungssystems "Meßstellen" eingerichtet, mit deren Hilfe es möglich ist, die Ermittlung der einzelnen Kenngrößen für die Bearbeitung kritischer Zugriffe wie folgt vorzunehmen:

Bearbeitungsdauer: Zeitdauer, gemessen von der ersten Aktivierung eines Auftragselementes bis zu seiner Vernichtung nach beendeter Bearbeitung.

Auftragswartezeit: Zeitspanne von der Ankunft einer die Generierung eines Auftragselementes bewirkenden Nachricht bis zu der durch den beschriebenen Auftrag bewirkten ersten Aktivierung der Protokolleinheit.

Länge der Auftragswarteschlange: Erfassung der Zu- und Abgänge der Auftragswarteschlange sowie der Zeiten zwischen je zwei Änderungen des Warteschlangenbestandes.

Konsistenzherstellung: Abzählung der umgekehrten Anordnungen kritischer Aufträge in dem den Dateimanager verlassenden Auftragsstrom.

Zeitdifferenz: maximale Differenz der Aktivierungszeitpunkte der Protokolleinheiten durch die sich auf den gleichen Auftrag beziehenden Auftragselemente in den beteiligten Dateimanagern.

Bei der Auswahl der zu variierenden Parameter konzentriert man sich auf die Parameter, deren Veränderung einen signifikanten Einfluß auf die interessierenden Simulationsresultate zeigt. Der Experimententwurf kann daher in den meisten Fällen nur sukzessive, unter Zuhilfenahme von Simulationsprobeläufen, zur endgültigen Auswahl der zu verändernden Modellparameter führen.

Die zunächst aufgrund der in 5.1. erläuterten Zielsetzung ausgewählten qualitativen Faktoren sind:

a) Die Variante des Koordinationsverfahrens, entsprechend den möglichen Kombinationen der 3 Alternativen (vgl. Tabelle 4.2-1).

b) Die Zahl der Dateimanager.

Ebenfalls zu den qualitativen Faktoren gehören die die Übertragungsverzögerungen der Nachrichten bestimmenden Verteilungen sowie die Verteilungen der für die Durchführung des eigentlichen kritischen Zugriffs im kritischen Abschnitt benötigten Zeiten. Für die Übertragung von Nachrichten konstanter Länge (engl. packet) über komplexe Kommunikationsnetze stellt zwar die Normalverteilung, wie Senger in /39/ zeigt, eine brauchbare Approximation der Verteilungen der Verzögerungszeiten dar; da das Kommunikationssystem im hier diskutierten Simulationsmodell jedoch vornehmlich die Aufgabe hat, eine Störung der Reihenfolge des nach Generierungszeiten vom Auftragsgenerator geordneten Auftragsstromes zu verursachen, sowie eine möglichst große Streuung der Übertragungszeiten innerhalb vorgebbarer Intervalle für interne Nachrichten zu bewirken, genügt eine

Gleichverteilung, um die Koordinationsverfahren unter strengen Voraussetzungen vergleichen zu können.

Aus Gleichverteilungen entnommen werden auch die Zugriffsausführungszeiten, was zumindest für solche Zugriffe, die den Transfer eines einzelnen Datensatzes zwischen Rechner und Hintergrundspeicher betreffen, eine gute Übereinstimmung mit der Realität bedeutet (vgl. /43/).

Die beim Experimententwurf berücksichtigten quantitativen Faktoren sind die Parameter der verwendeten Verteilungen:

c) Die Ankunftsrate λ_{tot} sowie die Zusammensetzung γ des Auftragsankunftsstromes

d) Mittelwert und Schwankungsbereich der Übertragungsverzögerungen für externe Nachrichten $\bar{T}_{UP}$ bzw. $\Delta T'_{UP}$ und für interne Nachrichten $\bar{T}_{ik}$ bzw. $\Delta T'_{ik}$

Mittelwerte und Schwankungsbereiche der Zugriffsausführungszeiten wurden für alle durchgeführten Experimente konstant gehalten.

Bei der Festlegung der Wertebereiche der quantitativen Faktoren wurde von folgenden Voraussetzungen ausgegangen:

- Die aus den Verteilungen resultierenden Übertragungsverzögerungen sollen bei allen Versuchen in einer Größenordnung liegen, die die Vernachlässigung der Bearbeitungszeiten der Warteschlangenverwaltungsmechanismen und der Protokolleinheiten zulassen. Diese Bedingung muß entsprechend auch bei der Festlegung der Zugriffsausführungszeiten berücksichtigt sein.
- Die Schwankungsbereiche der Übertragungsverzögerungen sollen bezüglich des Auftragsankunftsstromes in der gleichen Größenordnung wie die mittlere Zwischenankunftszeit liegen, um einen möglichst hohen Grad der Konsistenzstörung (Störung der geordneten Reihenfolge) des beim Dateimanagersystem eintreffenden Auftragsstromes zu erreichen. Aus $\Delta T'_{UP}$ und $\bar{T}_{UP}$ ergibt sich der Wertebereich für die Übertragungsverzöge-

rungen T_{UP_i}, i=1(1)r der externen Nachrichten als das Intervall

$$(5.2\text{-}4) \qquad \bar{T}_{UP} - \frac{1}{2}\,\Delta T'_{UP} \leq T_{UP_i} \leq \bar{T}_{UP} + \frac{1}{2}\,\Delta T'_{UP}$$

Entsprechendes gilt für den Wertebereich der Übertragungsverzögerungen der internen Nachrichten T_{ik}, $i \neq k$, k,i=1(1)r.

- Kontrollnachrichten (d.h. interne Nachrichten) sollen durch je einen Informationsblock, der als eine Einheit (engl. message-packet) /13/ über das Kommunikationssystem übertragbar ist, realisiert werden können. Bei der Übertragung sollen sie eine bevorzugte Behandlung - verglichen mit externen Nachrichten - genießen, so daß für Kontrollnachrichten der Interdateimanager-Kommunikation gegenüber externen Nachrichten eine im Mittel kürzere Übertragungsdauer angenommen werden kann.
- Ein stationäres Verhalten des Modells wird durch Berücksichtigung der Bedingung

$$(5.2\text{-}5) \qquad 0 < \rho < 1 \quad \text{mit } \rho = \frac{\lambda_{tot}}{\mu_{tot}}$$

$$\frac{1}{\mu_{tot}} = \text{mittlere Bearbeitungsdauer für Aufträge}$$

für die Verkehrsintensität ρ gewährleistet.

Auf die Untersuchung des Einflusses von variablen Karenzzeitintervall-Längen (vgl. 3.1.) bei zweistufigen Verfahren wurde verzichtet; die Länge von T_K wurde generell mit 0 angenommen, um möglichst strenge Anforderungen an die konsistenzherstellenden Eigenschaften der auf zweistufigen Protokollen basierenden Verfahren zu stellen.

5.4. Durchführung der Experimente und Diskussion der Resultate

Die Experimente wurden in drei Gruppen eingeteilt, die die Untersuchung der konsistenzherstellenden Eigenschaften und die Ermittlung der Leistungskenngrößen der verschiedenen Koordinationsverfahren (vgl. 5.1.) für die Bearbeitung kritischer Zugriffe in Abhängigkeit von folgenden Einflüssen zum Ziele hatten:

Gruppe 1:

Variation der Zahl der Dateimanager r im Bereich $1 \leq r \leq r_{MAX}$ bei festgehaltenem Schwankungsbereich $\Delta T'_{UP} \approx \frac{1}{\lambda}$ der Übertragungsverzögerungen der externen Nachrichten. Der Maximalwert r_{MAX} wird durch den vertretbaren Rechenaufwand bestimmt.

Gruppe 2:

Veränderung der Störung des Auftragsstromes durch Variieren des Schwankungsintervalls $\Delta T'_{UP}$ im Bereich

$$(5.4\text{-}1) \qquad 0 \leq \Delta T'_{UP} < 2 \cdot \frac{1}{\lambda}$$

bei vorgegebener Dateimanagerkonfiguration r=const. > 1. Die Auswahl von r orientiert sich dabei an den Ergebnissen der Experimente der Gruppe 1.

Gruppe 3:

Veränderung der Zwischenankunftszeit $\frac{1}{\lambda_{tot}}$ des Auftragsstromes derart, daß sich eine Änderung des Verkehrskoeffizienten ρ bis zur Grenze des stationären Bereiches

$$(5.4\text{-}2) \qquad \rho_{max} \approx 1 > \rho > 0 \quad \text{mit } \rho = \frac{\lambda_{tot}}{\mu_{tot}(\lambda)}$$

ergibt. Die reziproke Bedienungsdauer μ_{tot} für kritische Aufträge war dabei als Funktion der Ankunftsrate λ kritischer Aufträge zu berücksichtigen. Weiter ist bei diesen Experimenten ebenfalls eine feste Dateimanagerkonfiguration (r=const.) sowie eine signifikante Störung des Auftragsankunftsstromes ($\Delta T'_{UP} \approx \frac{1}{\lambda}$)

vorgegeben.

Die einzelnen, bei fester Parameterkonstellation durchgeführten Experimente wurden nach jeweils 2000 ausgeführten kritischen Aufträgen abgebrochen. Zur Vermeidung des Einflusses von Abhängigkeiten aufeinanderfolgender Auftragsbearbeitungen bei der Ermittlung der Varianz der gemessenen Leistungskenngrößen wurde nach dem Verfahren von Conway /8/ vorgegangen: die Meßreihen der an 2000 Aufträgen gemessenen Werte wurden in Intervalle mit jeweils 100 Werten eingeteilt und die aus den Intervallen resultierenden Mittelwerte dann zur Berechnung des Gesamtmittelwertes und der Varianzabschätzung herangezogen. Die Ergebnisse der Varianzabschätzungen dienten der Berechnung der in den Darstellungen angegebenen 90% Vertrauensintervalle für die Mittelwerte.

Mit Hilfe des auf die Wartezeiten und Warteschlangenlängen angewendeten Trendtests (vgl. /2/) wurden die Stationarität des Modellverhaltens und das Abklingen der Einschwingphase überprüft. Es zeigte sich, daß die Verwerfung der ersten vier Intervalle (s.o.) in allen Fällen ausreichte, um den Einfluß der Einschwingphase auf die Meßergebnisse zu eliminieren.

Weiterhin wurde, um zu überprüfen, ob die Einteilung der gemessenen Zeitserie in Intervalle zu jeweils 100 Aufträgen zur Aufhebung der Abhängigkeitseinflüsse genügte, die Autokorrelationsfunktion ermittelt. Bei allen Versuchen, außer im Grenzbereich $\rho \approx 1$ war die Autokorrelation über einen Abstand von mehr als 50 Aufträgen hinweg vernachlässigbar.

Das Konsistenzherstellungsvermögen der einzelnen Koordinationsverfahren, d.h. also die Fähigkeit, einen in seiner Reihenfolge der kritischen Aufträge durch Überholvorgänge beim Nachrichtentransport gestörten Auftragsstrom dennoch in der richtigen Reihenfolge abzuarbeiten, wurde durch Abzählung der im Auftragsstrom nach Bearbeitung durch die Dateimanager noch vorhandene Zahl der umgekehrten Anordnungen ermittelt.

Sei $Z_1, Z_2, \ldots, Z_m$ die beobachtete Bearbeitungsreihenfolge der kritischen Aufträge, sei ferner

$$h_{ij} = \begin{cases} 1 \text{ wenn } ID(Z_i) > ID(Z_j) & i<j \\ 0 \text{ sonst} \end{cases}$$

wenn $ID(Z_k)$ die Identifikation von Z_k darstellt. Die Zahl INV der umgekehrten Anordnungen (Inversionen) ist dann gegeben durch

$$(5.4\text{-}3) \qquad INV = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} h_{ij}$$

Experimente der ersten Gruppe:

Die in den Abbildungen 5.4-1 a) bis c) dargestellten Simulationsresultate zeigen den Verlauf der normierten Zahl der Inversionen für die 8 untersuchten Koordinationsverfahren in Abhängigkeit von der Zahl der im System vorhandenen Dateimanager bei verschiedenen Graden der Störung des Auftragsankunftsstromes, entsprechend drei verschiedenen Schwankungsbereichen $\Delta T'_{UP}$ für die Übertragungsverzögerungen externer Nachrichten:

a) $\Delta T'_{UP} = 0{,}5 \cdot \frac{1}{\lambda}$

b) $\Delta T'_{UP} = \frac{1}{\lambda}$

c) $\Delta T'_{UP} = 1{,}9 \cdot \frac{1}{\lambda}$

Bezugspunkt der Normierung ist die jeweils für das Verfahren EEF am Ein-Dateimanagersystem gemessene Zahl der Inversionen.

Die Messungen bestätigen zunächst die erwartete Überlegenheit der zweistufigen Protokolle bezüglich des Konsistenzherstellungsvermögens. Für das Ein-Dateimanagersystem reduzieren sich die 8 Verfahren auf zwei: Mehrfachinitialisierung ist in diesem Falle mit der Einfachinitialisierung identisch, die Protokolleinheit funktionslos. Die Auswirkung der prioritätsorientierten

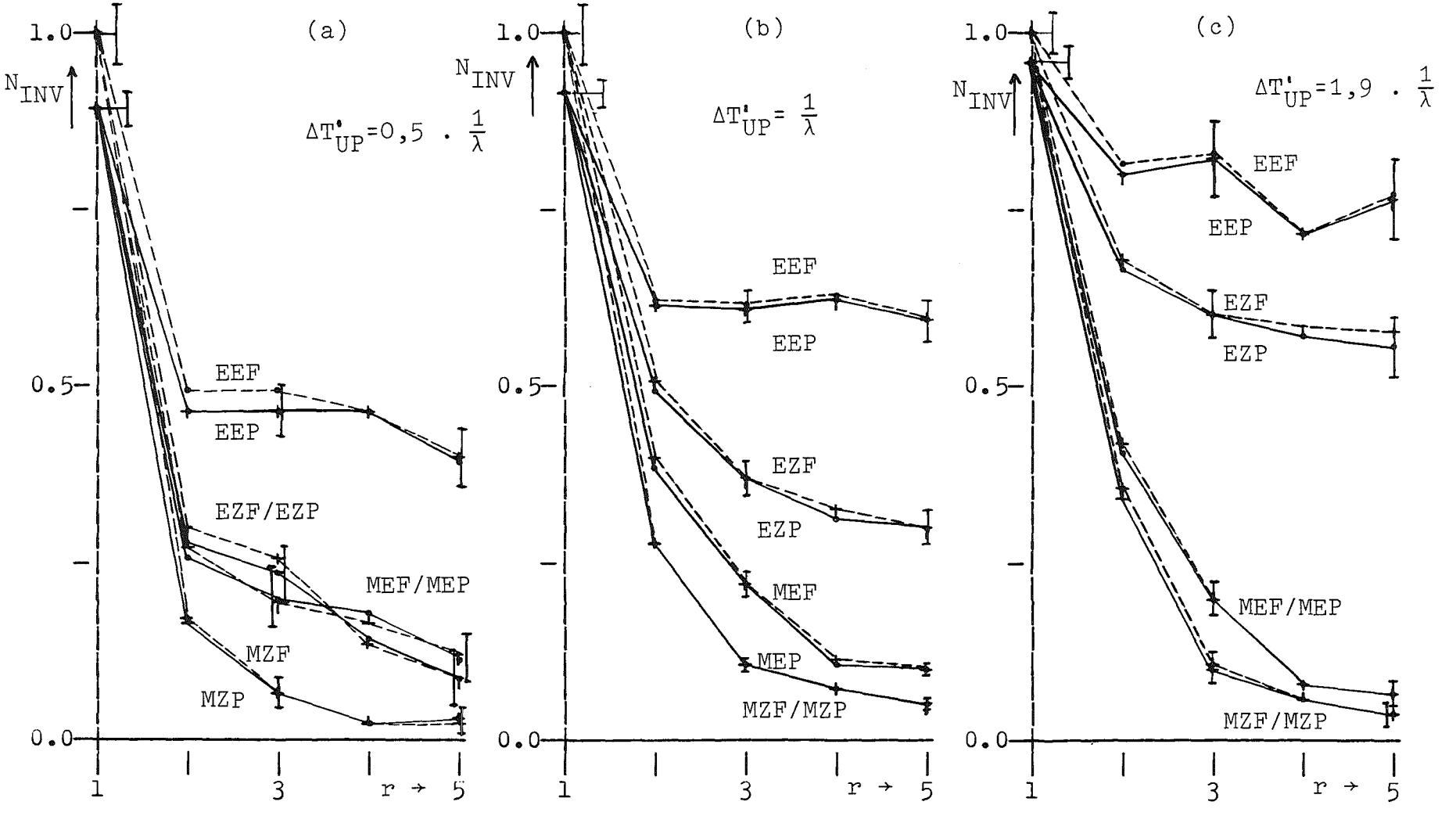


Abbildung 5.4-1 Normierte Zahl der Inversionen N_{INV} in Abhängigkeit von der Zahl r der Dateimanager bei unterschiedlich gestörtem Auftragsankunftsstrom für konstante Zwischenankunftszeit $\frac{1}{\lambda} = 16 \cdot T_{ik}$

Nachrichtenwarteschlangenabarbeitung ist, ausgenommen beim Ein-Dateimanagersystem, wo diese Technik neben dem prioritätsorientierten Aufbau der Auftragswarteschlange die einzige Möglichkeit zur Erreichung einer konsistenten Auftragsbearbeitungsreihenfolge darstellt, nahezu bedeutungslos. Dieser Effekt ist unabhängig von dem die "Unordnung" des Auftragsankunftsstromes bestimmenden Parameter $\Delta T'_{UP}$ zu beobachten. Da sich der Einfluß der Nachrichtenwarteschlangenabarbeitung auf die Leistungskenngrößen generell als insignifikant erwies, wurden die Alternativen der Abarbeitungsdisziplin bei den Experimenten der Gruppe 2 und 3 nicht berücksichtigt. Für die verbliebenen Varianten der Koordinationsverfahren werden im weiteren sinngemäß die Kurzbezeichnungen EE, EZ, ME und MZ (vgl. Tab. 4.2-1) verwendet.

Bemerkenswert ist die Verbesserung des Konsistenzherstellungsvermögens mit zunehmender Zahl der Dateimanager im System. Dies ist auf folgende Gründe zurückzuführen: Mit der Zahl der Dateimanager nimmt die Wahrscheinlichkeit einer längeren Bearbeitungsdauer für kritische Aufträge zu, da die für die Synchronisationen benötigte Zeitspanne sich an der Länge der für die Übertragung der Kontrollnachrichten benötigten Zeit orientiert. Durch die länger dauernde Testphase wird die Wahrscheinlichkeit für Auftragsverdrängungen erhöht. Zusätzlich aber kommt es durch die verlängerte Auftragsbearbeitungsdauer zu einer Zunahme der mittleren Auftragswarteschlangenlänge und damit zu einer verbesserten Prioritätsberücksichtigung innerhalb der Auftragswarteschlange.

Die geringere Zahl der Inversionen bei Mehrfachinitialisierung im Vergleich zur Einfachinitialisierung ist ebenfalls auf die Verlängerung der mittleren Bearbeitungsdauer zurückzuführen: Die Ausführung des kritischen Zugriffs wird von einem Dateimanager erst veranlaßt, wenn die auf diesen Zugriffsauftrag bezogene externe Nachricht empfangen worden ist (vgl. 4.2.).

Ein weiterer Vergleich der Abbildungen 5.4-1 a), b) und c) läßt mit wachsender Größe des Schwankungsbereiches $\Delta T'_{UP}$ der Verzögerungszeit externer Nachrichten ein Nachlassen des Konsistenzherstellungsvermögens speziell bei den Verfahren mit

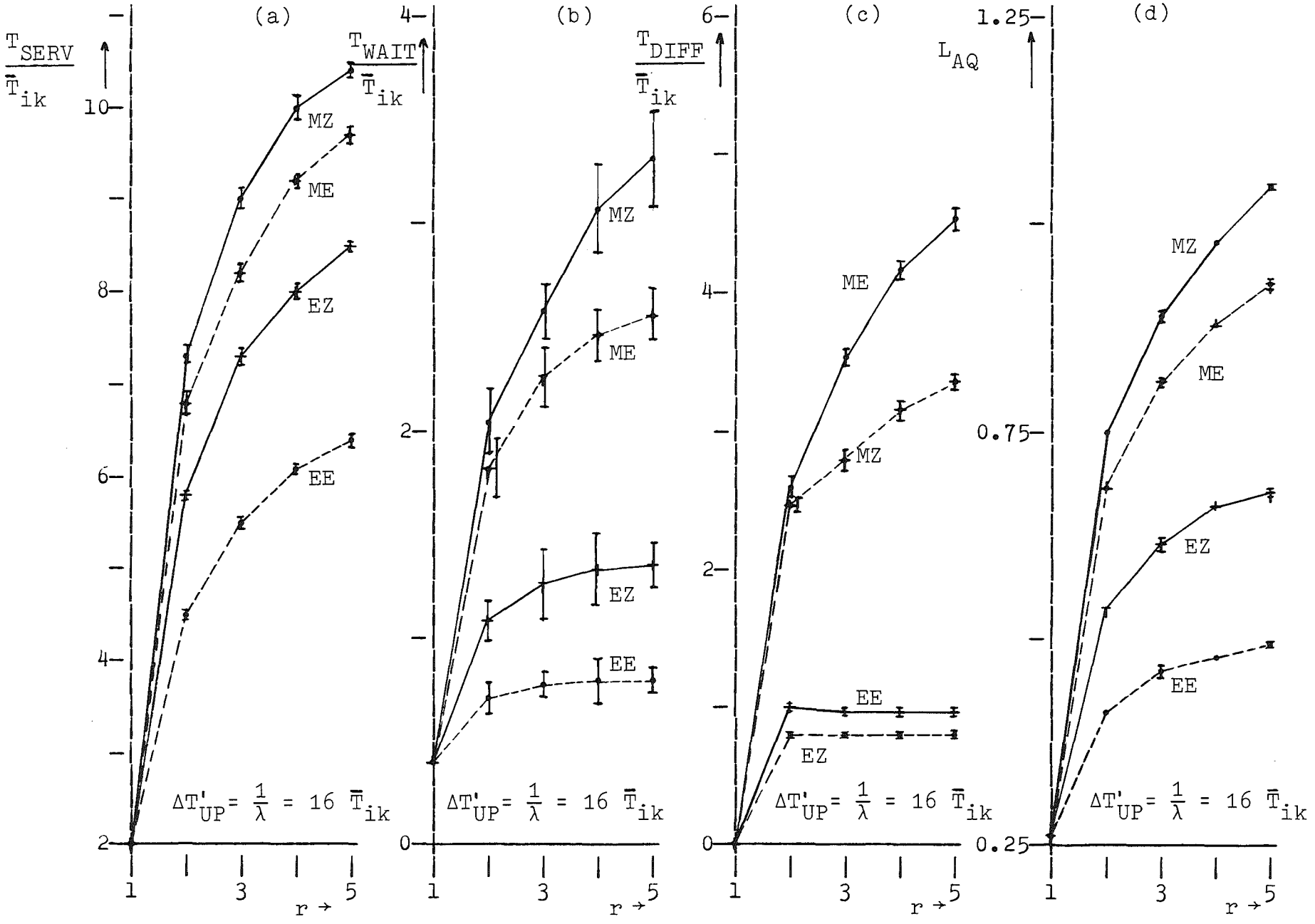


Abbildung 5.4-2 Verlauf der mittleren Bearbeitungsdauer T_{SERV}, Auftragswartezeit T_{WAIT}, Zeitdifferenz T_{DIFF} und der mittleren Länge L_{AQ} der Auftragswarteschlange in Abhängigkeit von der Zahl der Dateimanager r

Einzelinitialisierung erkennen. Eine Begründung dafür ist in der weitgehend von $\Delta T'_{UP}$, wie die Resultate weiterer Experimente zeigen, unabhängigen mittleren Bearbeitungsdauer dieser Verfahren zu suchen: Während $\frac{1}{\mu}$ konstant bleibt und damit die im Mittel zur Konsistenzherstellung verfügbare Zeit $T_{kon} < \frac{1}{\mu}$, wächst auf der anderen Seite monoton als Funktion von $\Delta T'_{UP}$ die Wahrscheinlichkeit, daß auf einen zur Zeit t' initialisierten Auftrag zur Zeit $t'+T_{kon}+\Delta t$ ein Auftrag höherer Priorität folgt, der nicht mehr zur Verdrängung des bereits initialisierten Auftrags führen kann.

Der vermutete Verlauf der mittleren Auftragsbearbeitungsdauer, Auftragswartezeit und Warteschlangenlänge wird durch die in Abb. 5.4-2 wiedergegebenen Messungen bestätigt.

Aus den Meßergebnissen der Versuche der ersten Gruppe kann geschlossen werden, daß das gegenüber den einstufigen Verfahren bessere Konsistenzherstellungsvermögen zweistufiger Verfahren bei Systemen, die mehr als zwei Dateimanager umfassen, nicht durch eine proportionale Einbuße der an der mittleren Bearbeitungsdauer bzw. Verweilzeit von Aufträgen im Überwachungssystem gemessenen Leistungsfähigkeit erkauft werden muß:

Für $\Delta T'_{UP} = \frac{1}{\lambda}$ erhalten wir die in Tab. 5.4-1 wiedergegebenen Vergleichswerte, die eindeutig die Überlegenheit der zweistufigen Verfahren erkennen lassen, wenn Konsistenzherstellungsvermögen und mittlere Dauer der Bearbeitung kritischer Aufträge gleich bewertet werden.

Verfahren	Dateimanager	Vergl. mit Verfahren	Verb. Konsistenzh.	Zunahme Bedzeit	Zunahme Verweilz.
EZF EZP	3	EEF EEP	40%	30%	35%
EZF EZP	5	EEF EEP	50%	33%	36%
MZF MZP	3	MEF MEP	50%	10%	11%
MZF MZP	5	MEF MEP	50%	8%	12%

Tabelle 5.4-1

Experimente der zweiten Gruppe:

Aufgrund der bei den Leistungskenngrößen in Abhängigkeit von der Zahl der Dateimanager zu beobachtenden Sättigungscharakteristik einerseits (vgl. Abb. 5.4-1 und 5.4-2) und zur Vermeidung eines zu hohen Rechenzeitaufwandes andererseits wurde für die Experimente der Gruppe 2 die Zahl der Dateimanager im Überwachungssystem auf drei festgelegt.

Abbildung 5.4-3 zeigt den Verlauf der normierten Zahl der Inversionen, bezogen auf die beim Ein-Dateimanagersystem für das Verfahren mit FIFO Nachrichtenabarbeitung beobachtete Inversionszahl, in Abhängigkeit von der Größe des Schwankungsbereiches $\Delta T'_{UP}$.

Das bereits bei den Experimenten der Gruppe 1 deutlich werdende Nachlassen des Konsistenzherstellungsvermögens der Verfahren mit Einzelinitialisierung bei größer werdenden $\Delta T'_{UP}$, erkenntlich an der raschen Annäherung an den Wert 1, wird bestätigt. Weiter sieht man, daß für Werte von $\Delta T'_{UP} < 0{,}5 \cdot \frac{1}{\lambda}$ der Einfluß der durch "spät" eintreffende externe Nachrichten erhöhten Bearbeitungsdauer und Wartezeit bei Verfahren mit Mehrfachinitialisierung auf das Konsistenzherstellungsvermögen verschwindet; das Konsistenzherstellungsvermögen wird in diesem Bereich vorrangig durch den Typ des Koordinationsprotokolls bestimmt. Der den Abbildungen 5.4-4 a) und b) zu entnehmende, für die verschiedenen Verfahren unterhalb $\Delta T'_{UP} = 0{,}5 \cdot \frac{1}{\lambda}$ nur noch geringfügig differierende Verlauf der Bearbeitungs- und Wartezeiten bestätigt dies; die Güte der Konsistenzherstellung durch verschiedene Verfahren innerhalb der gleichen Protokollklasse wird durch die mittlere Dauer der Auftragswartezeiten bestimmt.

Die mittleren Wartezeiten der Verfahren mit Mehrfachinitialisierung durchlaufen im Bereich $0 \leq \Delta T'_{UP} \leq 0{,}5 \cdot \frac{1}{\lambda}$ ein Minimum. Zu diesem Minimum kommt es, wenn mit wachsendem $\Delta T'_{UP}$ die Wahrscheinlichkeit wächst, daß die Ankunft einer externen Nachricht zur Initialisierung eines kritischen Auftrags bereits auf ein durch Fremdinitialisierung erzeugtes Auftragselement des gleichen Auftrags trifft. Dies ist der Fall, wenn für den Erwar-

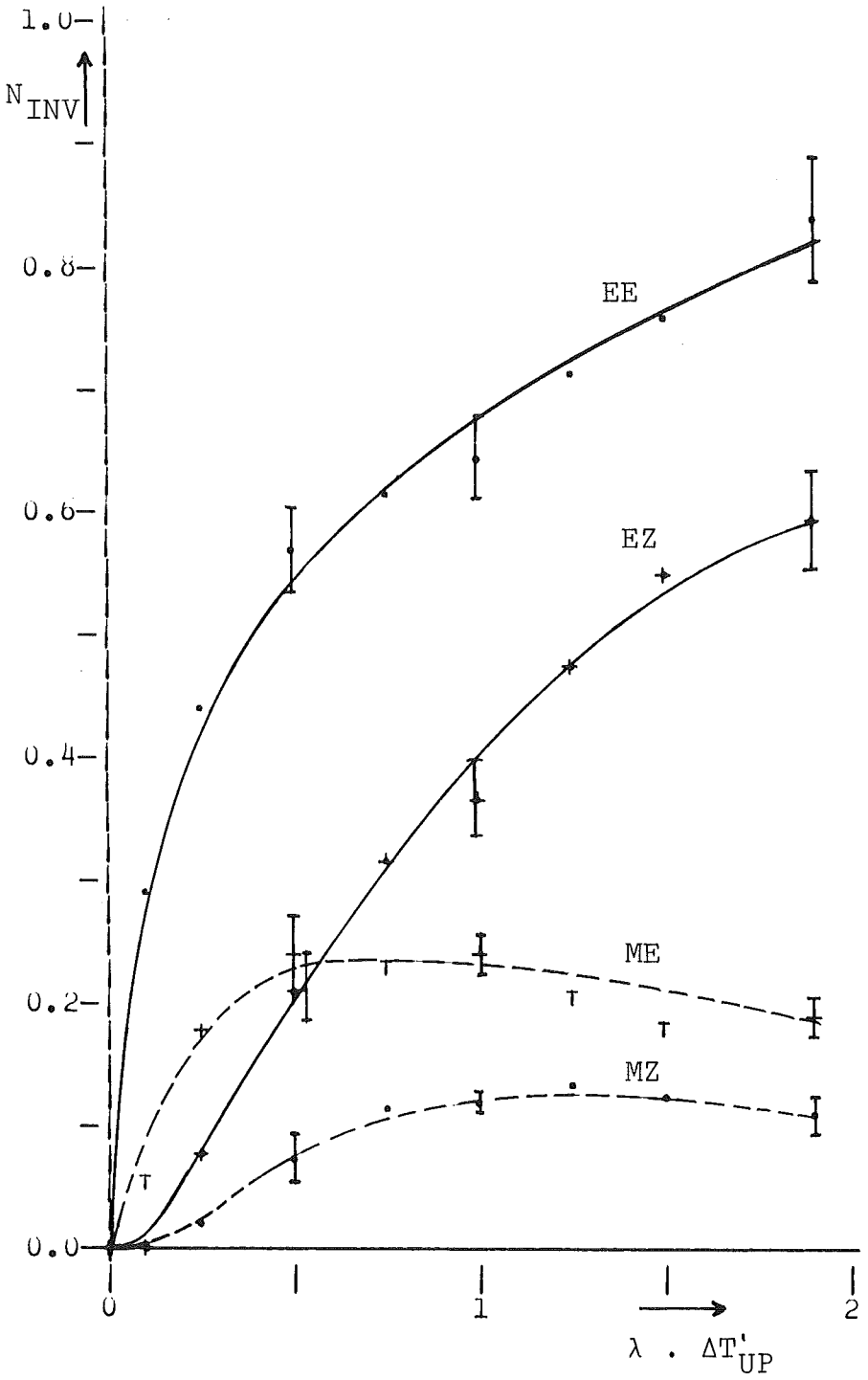


Abbildung 5.4-3 Normierte Inversionszahl N_{INV} bei einem 3-Dateimanager-System in Abhängigkeit von der Größe des Schwankungsbereichs $\Delta T'_{UP}$ der Übertragungsverzögerung externer Nachrichten für $\frac{1}{\lambda} = 16 \cdot \overline{T}_{ik}$

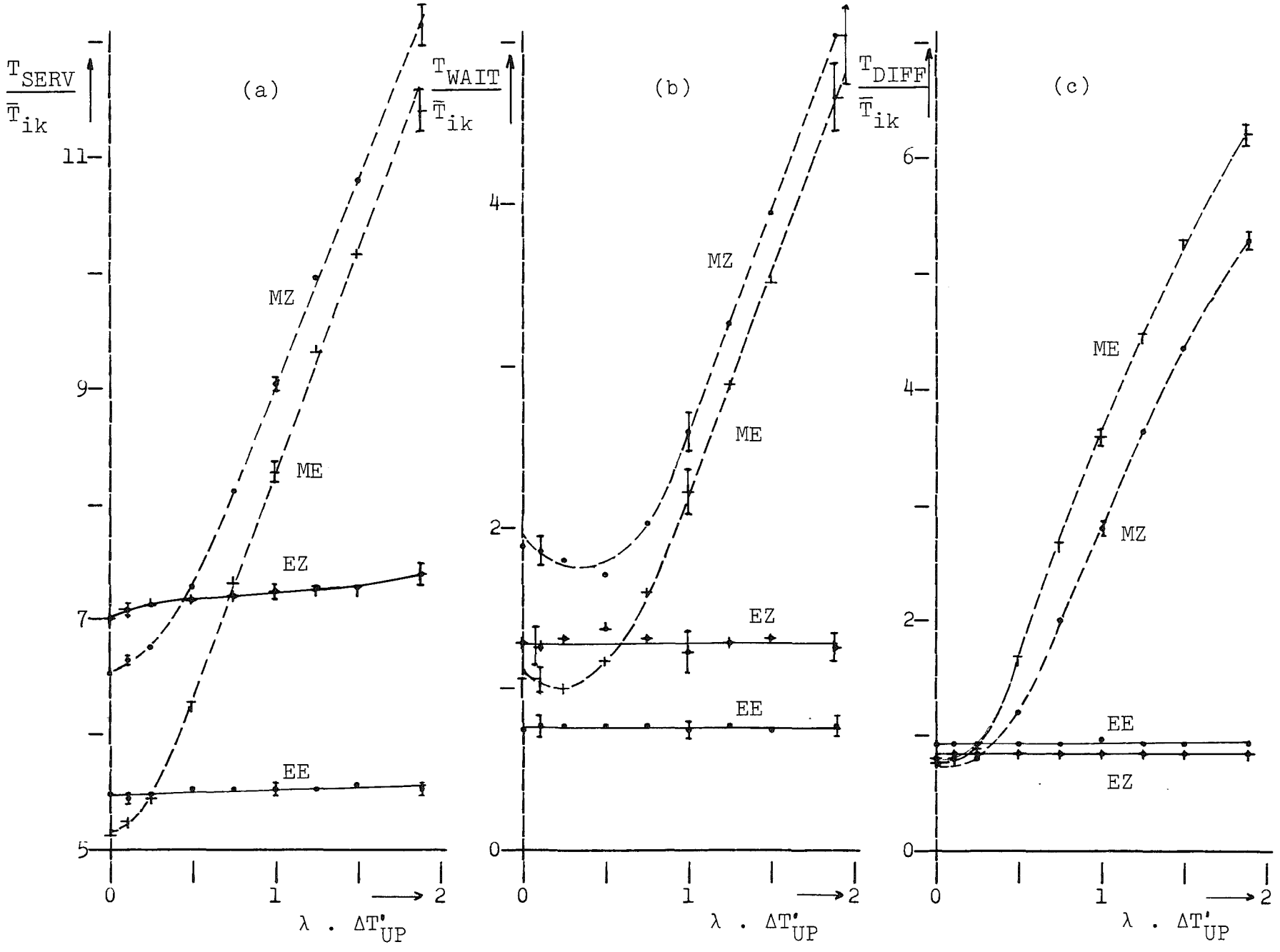


Abbildung 5.4-4 Verlauf von T_{SERV}, T_{WAIT} und T_{DIFF} in Abhängigkeit von der Größe des Schwankungsbereichs $\Delta T'_{UP}$ bei r=3 und $1/\lambda = 16\ \bar{T}_{ik}$

tungswert der maximalen Differenz der Initialisierungszeiten t_i^z für kritische Aufträge Z bei den beteiligten Dateimanagern gilt

$$(5.4\text{-}4) \qquad E(T_{TEST}) = E(\underset{i\varepsilon I}{Max}\{t_i^z\} - \underset{i\varepsilon I}{Min}\{t_i^z\})$$

$$\text{mit } I = \{1,\ldots,r\}$$

T_{TEST} gibt die Dauer der Testphase an.
Bei den im Experiment gegebenen Verteilungen wird (5.4-4) für ein Drei-Dateimanagersystem zu

$$(5.4\text{-}4a) \qquad E(T_{TEST}) = \frac{1}{2}\Delta T'_{UP}$$

$E(T_{TEST})$ liegt, wie man in Abb. 5.4-4a den Schnittpunkten der Kurven EZ/MZ und EE/ME entnimmt, für zweistufige Verfahren bei $3{,}25 \cdot \bar{T}_{ik}$, für einstufige Verfahren bei $2 \cdot \bar{T}_{ik}$.

Bei vergleichbaren Warte- sowie Bearbeitungszeiten sind die mit Mehrfachinitialisierung arbeitenden Verfahren den Verfahren mit Einfachinitialisierung, bei gleichem Typ des verwendeten Protokolls, im Bereich $\Delta T'_{UP} < 0{,}25 \cdot \frac{1}{\lambda}$ geringfügig überlegen, sowohl bezüglich der Konsistenzherstellung als auch in Bezug auf die Zeitdifferenz bei der Aktivierung des gleichen Zugriffsauftrags (siehe Abb. 5.4-4c).

Experimente der dritten Gruppe:

Die Versuche wurden ebenfalls an einer Drei-Dateimanagerkonfiguration durchgeführt und dienten der Untersuchung des Verhaltens der Koordinationsverfahren im Grenzbereich der Stationarität. Zur Vermeidung eines zu großen Aufwandes wurden die Untersuchungen auf Verfahren mit Einfachinitialisierungen beschränkt (EE und EZ). Der Störungsgrad des Auftragsankunftsstromes wurde bei allen Experimenten durch $\Delta T'_{UP} = \frac{1}{\lambda}$ bestimmt.

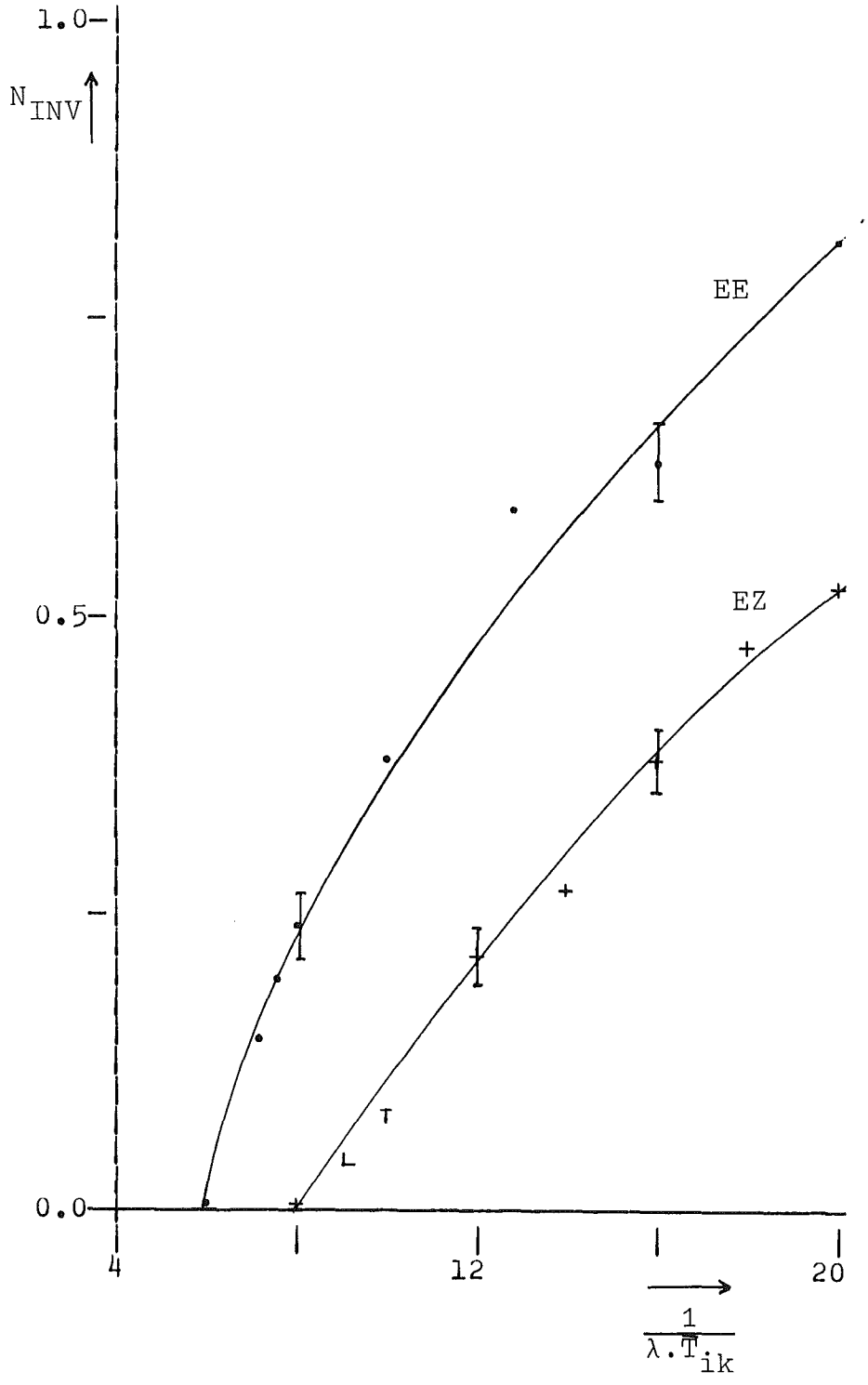


Abbildung 5.4-5 Normierte Inversionszahl N_{INV} in Abhängigkeit von der Zwischenankunftszeit kritischer Aufträge $\frac{1}{\lambda}$ bei r=3 und $\Delta T'_{UP} = \frac{1}{\lambda}$

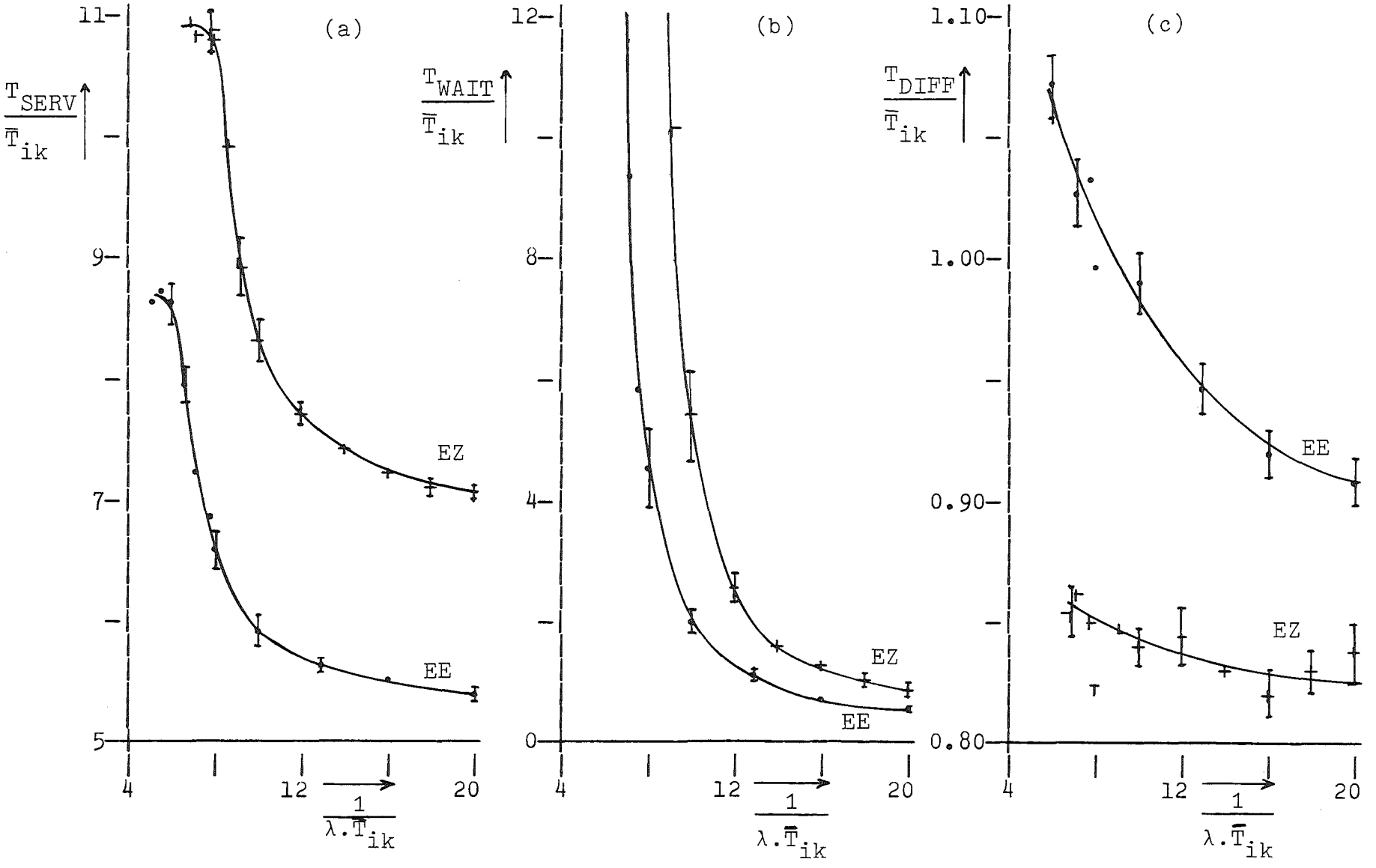


Abbildung 5.4-6 Mittlere Bearbeitungsdauer T_{SERV}, Wartezeit T_{WAIT} und Zeitdifferenz T_{DIFF} in Abhängigkeit von der Zwischenankunftszeit kritischer Aufträge $\frac{1}{\lambda}$ bei r=3 und $\Delta T'_{UP}=1/\lambda$

Abbildung 5.4-5 zeigt den gemessenen Verlauf der Zahl der Inversionen in Abhängigkeit von der Zwischenankunftszeit für die Verfahren EE und EZ. Unterhalb einer Zwischenankunftszeit von $\frac{1}{\lambda} = 6{,}0 \cdot \bar{T}_{ik}$ bei EE und $8{,}0 \cdot \bar{T}_{ik}$ bei EZ wurden keine Inversionen mehr festgestellt. Das Erreichen einer vollständigen Konsistenzherstellung macht sich in einem Abflachen der Meßkurve der mittleren Bearbeitungszeit für kleine Zwischenankunftszeiten bemerkbar, wie aus Abb. 5.4-6a zu ersehen. Wie man der Darstellung weiter entnimmt, liegt der Bereich, in dem die Bedienungszeit von der Ankunftsrate λ unabhängig wird, bereits im nichtstationären Bereich: die Verkehrsintensität ρ nimmt für das Verfahren EZ bei $\frac{1}{\lambda} \approx 9 \cdot \bar{T}_{ik}$ und für EE bei $\frac{1}{\lambda} \approx 7 \cdot \bar{T}_{ik}$ den Wert 1 an; erkenntlich wird das Einsetzen des instabilen Verhaltens am plötzlichen steilen Ansteigen der Auftragswartezeiten für Zwischenankunftszeiten $\frac{1}{\lambda} < 12 \cdot T_{ik}$ für EZ und $\frac{1}{\lambda} < 9{,}5 \cdot T_{ik}$ für EE, entsprechend einem Verkehrskoeffizienten von $\rho \approx 0{,}65$.

Die starke Zunahme der mittleren Bearbeitungsdauer für kritische Aufträge im Bereich kleiner Werte für $\frac{1}{\lambda}$ läßt sich durch die Inanspruchnahme der konsistenzherstellenden Eigenschaften beider Verfahrensvarianten und die daraus sich ergebenden Auftragsverdrängungen erklären.

Unter Zusammenfassung der wesentlichsten Ergebnisse der Simulationsexperimente kann man folgende Wertung der Verfahren zur Koordination kritischer Zugriffe vornehmen:

- Die sich nur durch verschiedene Abarbeitungsdisziplin der Nachrichtenwarteschlangen unterscheidenden Verfahren zeigen kein signifikant unterschiedliches Verhalten bezüglich des Verlaufs der Leistungskenngrößen in Abhängigkeit von der Zahl der Dateimanager und dem Grad der Störung des Auftragsankunftsstromes. Wegen des geringeren Implementierungsaufwandes sind jedoch Verfahren mit FIFO-Abarbeitung der Nachrichtenwarteschlangen vorzuziehen.
- Werden Konsistenzherstellungsvermögen und Bearbeitungsdauer der Zugriffsaufträge gleich gewichtet, so sind die auf der Basis zweistufiger Koordinationsprotokolle aufgebauten Ver-

fahren den mit einstufigem Koordinationsprotokoll vorzuziehen, wenn die mittlere Dauer der Testphase in der Größenordnung der mittleren Schwankung der Übertragungsverzögerung externer Nachrichten liegt.

- Im Bereich geringer Konsistenzstörung des Auftragsankunftsstromes ($\Delta T'_{UP} < \frac{1}{4\lambda}$) sind die Verfahren mit Mehrfachinitialisierung denen mit Einfachinitialisierung bezüglich des Konsistenzherstellungsvermögens, der Verweilzeit der Zugriffsaufträge im Dateimanagersystem und der Gleichzeitigkeit der Zugriffsausführung mindestens gleichwertig.
- Für die Verfahren mit Einfachinitialisierung wurde bei einer Störung der Konsistenz des Auftragsankunftsstromes durch $\Delta T'_{UP} = \frac{1}{\lambda}$ die vollständige Konsistenzherstellung im Grenzbereich des stationären Verhaltens ($\rho \approx 1$) erreicht.

6. Zusammenfassung

Die Zielsetzung dieser Arbeit bestand in der Entwicklung von Verfahren zur Koordination kritischer Zugriffe zu verteilten Datenbanken in Rechnernetzen unter der Randbedingung einer dezentral organisierten Überwachung.

Ausgangspunkt war die Tatsache, daß die Verteilung der Komponenten einer Datenbank als verwandte Dateien über zu einem Verbund zusammengeschlossene Rechner zunehmend Interesse findet. Die Gründe dafür liegen teils in der durch verteilte Komponenten ermöglichten Verringerung der Gesamtkosten und Verbesserung der Effizienz, teils in der durch redundant realisierte Datenbankkomponenten erreichbaren erhöhten Zuverlässigkeit und Verfügbarkeit von Datenbanksystemen.

Mit der Verteilung von Datenbankkomponenten über mehrere Rechner in einem Rechnernetz tritt das Problem der Koordination der Zugriffe in den Vordergrund, deren unkontrollierte Durchführung die Konsistenz der abgespeicherten und gewonnenen Information in Frage stellt. Methoden zur Lösung des Koordinationsproblems für die Durchführung dieser kritischen Zugriffe existieren zwar für nicht verteilte Datenbanken, ihre Übertragung auf verteilte Datenbanken bedingt jedoch das Vorhandensein einer zentralen Überwachungsinstanz.

Die Nachteile einer zentralen Überwachung liegen in dem durch sie geschaffenen Engpaß, sowohl bezüglich der Zugriffsausführung, als auch bezüglich der Zuverlässigkeit. Zudem widersprechen zentrale Instanzen dem sich immer mehr durchsetzenden Prinzip der dezentralen Betriebsorganisation von Rechnernetzen.

Es lag daher nahe, nach Verfahren zu suchen, die die Möglichkeiten der dezentralen Koordination kritischer Zugriffe zu verteilten Datenbanken eröffneten.

Bei der Entwicklung dieser Verfahren wurde davon ausgegangen, daß die Abwicklung der Zugriffskoordination durch unabhängige, über das Rechnernetz verteilte Systemprozesse - die Dateimanager - erfolgt, indem diese untereinander durch Interkommunikation ihre Einzelaktionen in geeigneter Weise aufeinander

abstimmen. Das System dieser Prozesse stellt, mit dem durch die Koordinationsverfahren festgelegten Aufbau der Dateimanager, ein dezentral organisiertes Überwachungssystem dar, welches auch bei Ausfall einzelner Prozesse in der Lage ist, von Benutzerprozessen übermittelte Aufträge auf Durchführung kritischer Zugriffe zu bearbeiten.

Die Basis für die zwischen den Dateimanagern erforderliche Kommunikation lieferten die heute bereits in Rechnerverbundnetzen verfügbaren "Elementarfunktionen" einer erweiterten Interprozeßkommunikation, die Dialoge zwischen im Netz beliebig verteilten Prozessen ermöglichen. Ausgehend von einem spezifizierten Minimalsatz dieser "Elementarfunktionen" wurde zunächst der Kern der Koordinationsverfahren als problemorientiertes Kommunikationsprotokoll unter Zuhilfenahme formaler Beschreibungselemente entwickelt, wobei die Kommunikationsprotokolle alle globalen Vereinbarungen über die zur Koordination notwendigen lokalen Aktionen der Dateimanager beinhalten.

Mit Hilfe der Beschreibungsmethoden wurden zwei Typen von Koordinationsprotokollen unter Verwendung gemeinsamer Bausteine erstellt: ein einfacher, als "einstufig" bezeichneter Koordinationsmechanismus sowie ein "zweistufiges" Protokoll, mit besonderen, an den Grad der Störung der geordneten Reihenfolge eines Auftragsankunftsstromes anpaßbaren, konsistenzherstellenden Eigenschaften.

Die Koordinationsprotokolle allein genügen jedoch nicht zur vollständigen Festlegung des zur Zugriffskoordination notwendigen Aufbaus der Dateimanager; durch die Protokolle wird lediglich die für die Dialogführung mit den anderen Dateimanagern verantwortliche "Protokolleinheit" eines Dateimanagers spezifiziert. Zusätzlich muß ein Mechanismus vorhanden sein, der die Verwaltung der dem Dateimanager zugeordneten Nachrichten- und Auftragswarteschlangen übernimmt.

Durch Kombination der zwei Protokolltypen mit vier unterschiedlich arbeitenden Warteschlangenverwaltungsmechanismen konnten 8 verschiedene Koordinationsverfahren zusammengestellt werden,

die die Festlegung des erforderlichen Dateimanageraufbaus gestatten.

Die erarbeiteten Verfahren wurden einer vergleichenden Untersuchung ihrer Leistungsfähigkeit unterzogen. Von Interesse waren dabei speziell die qualitativen Unterschiede der bei gleichem Auftragsprofil für die Koordination und Durchführung kritischer Zugriffe im Mittel benötigten Zeit, der Auftragswartezeiten, der Zeitdifferenz der Zugriffsinitialisierung sowie des Konsistenzherstellungsvermögens. Die Untersuchungen wurden anhand eines dafür entwickelten Simulationsmodelles vorgenommen. Die Experimente zeigten, als eines der wesentlichsten Resultate, daß mit dem zweistufigen Koordinationsprotokoll als Kern der Verfahren eine ausgezeichnete Wiederherstellung konsistenzgestörter Auftragsströme erzielt werden kann, ohne daß dafür ein nennenswerter Leistungsrückgang in Kauf genommen werden muß, was diese Verfahren speziell für Realzeitsysteme interessant macht.

Schlußwort

Die beschriebenen Verfahren zur Überwachung verteilter Datenbanken in Rechnernetzen wurden im Rahmen eines übergeordneten Forschungsvorhabens erarbeitet, das sich mit der Untersuchung möglicher Organisationsformen für Mehrrechnersysteme zum Einsatz für Prozeßlenkungsaufgaben befaßt. Die gewonnenen Resultate stellen einen Beitrag zu dem Teilbereich des Vorhabens dar, der die Entwicklung von Konzepten zur Erhöhung der Ausfallsicherheit, Verfügbarkeit und Effizienz von Mehrrechnersystemen zur Prozeßlenkung zum Ziel hat. Weiterführende Arbeiten auf diesem Gebiet werden die Gelegenheit bieten, die gewonnenen Ergebnisse im Rahmen von Implementierungen auf Prozeßrechnersystemen zu validieren und die Verfahren unter Berücksichtigung der Überlegungen zum fehlertoleranten Aufbau von Kommunikationsprotokollen weiter auszubauen.

An dieser Stelle sei Herrn Prof. G. Krüger besonders gedankt, der mir die Durchführung der erforderlichen Untersuchungen ermöglichte und wertvolle Anregungen bei der Anfertigung dieser Arbeit gab.

Gleichfalls danke ich Herrn Prof. H. Wettstein für wesentliche Ratschläge und kritische Anmerkungen; interessante Hinweise verdanke ich auch der Diskussion mit den Herren Dr. H. Beilner und Prof. H.H. Nagel.

Bei der Lösung von Teilproblemen und einer übersichtlichen Darstellung des Textes waren mir vor allem die Herren Dipl.-Inf. O. Drobnik, Dr. J. Nehmer, Dr. G. Petrich, Dipl.-Inf. F. Schumacher und Dipl.-Inf. R. Senger - allesamt Mitarbeiter des IDT - behilflich, denen ich hier für ihre Unterstützung danke.

Literatur

/1/ Bell, C.G., Habermann, A.N., McCredie, J., Rutledge, R., Wulf, W.
Computer Networks
Computer Science Review, 1969, Carnegie Mellon Univ., pp. 30-49

/2/ Bendat, J.S., Piersol, A.G.
Measurement and analysis of random data
John Wiley & Sons, New York, London, Sidney, 1966

/3/ Burke, P.J.
The Dependence of Sojourn Times in Tandem M/M/s Queues
Operations Research 17, 1969, pp. 754-755

/4/ Carr, S.C., Crocker, S.D., Cerf, V.G.
HOST-HOST communication protocol in the ARPA network
AFIPS Conference Proceedings Vol. 36,
Spring Joint Computer Conference, 1970, pp. 589-597

/5/ Casey, R.G.
Allocation of copies of a file in an information network
AFIPS Conference Proceedings Vol. 40,
Spring Joint Computer Conference, 1972, pp. 617-625

/6/ Chu, W.W.
Optimal File Allocation in a Multiple Computer System
IEEE Transactions on Computers, Vol. C-18, No. 10,
October 1969, pp. 885-889

/7/ Coffman Jr., E.G., Denning, P.J.
Operating Systems Theory
Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1973, pp. 31-82

/8/ Conway, R.W.
Some Tactical Problems in Digital Simulation
Management-Science, Vol.10, No.1, Oct. 1963, pp. 47-61

/9/ Courtois, P.J., Heymans, F., Parnas, D.L.
Concurrent Control with "Readers" and "Writers"
Communications of the ACM, Oct. 1971, Vol.14,
No. 10, pp. 667-668

/10/ Crocker, S.D., Heafner, J.F., Metcalfe, R.M., Postel, J.B.
Function-oriented protocols for the ARPA computer network
AFIPS Conference Proceedings Vol. 40,
Spring Joint Computer Conference, 1972, pp. 271-279

/11/ Dadda, L., LeMoli, G.
An Introduction to Computer Networks
IRIA Proceedings of the 1. European Workshop on
Computer Networks, Arles, 1973

/12/ Dahl, O.J., Myhrhaug, B., Nygaard, K.
Common Base Language
SIMULA information, Norwegian Computing Center,
Oslo, 1970

/13/ Davies, D.W.
Principles of packet switching
IRIA Proceedings of the 1. European Workshop on
Computer Networks, Arles, 1973

/14/ Dijkstra, E.W.
Co-operating Sequential Processes
Technological University, Eindhoven, 1965

/15/ Dijkstra, E.W.
The Structure of "THE"-Multiprogramming System
Communications of the ACM, May 1968, Vol.11,
No. 5, pp. 341-347

/16/ Engles, R.W.
A Tutorial on Data-Base Organization
Annual Review in Automatic Programming, Vol.7,
Part 1, pp. 1-64, 1972

/17/ Farber, D., Larson, K.C.
The Structure of the Distributed Computing System
University of California, Irvine, 1972

/18/ Gilbert, P., Chandler, W.J.
Interference Between Communicating Parallel Processes
Communications of the ACM, June 1972, Vol.15, No. 6,
pp. 427-437

/19/ Habermann, A.N.
Synchronization of Communicating Processes
Communications of the ACM, March 1972, Vol.15,
No. 3, pp. 171-176

/20/ Hansen, P.B.
The Nucleus of a Multiprogramming System
Communications of the ACM, April 1970, Vol.13,
No. 4, pp. 238-250

/21/ Hansen, P.B.
A Comparison of Two Synchronizing Concepts
Acta Informatica 1, 1972, pp. 190-199

/22/ Hoffmann, H.J.
Syntax-Directed Communication Control - Some Implications of a Design Approach
IBM Research Report RZ 421, April 1971

/23/ Holler, E.
Betriebsmittelvergabe in heterogenen Rechnernetzen bei dezentralisierter Netzwerksteuerung
Nachrichtentechnische Fachberichte Band 44, NTG-GI-Tagung, Darmstadt 1972, S. 96-105

/24/ Holler, E.
Files in Computer Networks
IRIA Proceedings of the 1. European Workshop on Computer Networks, Arles, 1973

/25/ Jackson, J.R.
Networks of Waiting Lines
Operations Research 5, 1957, pp. 518-521

/26/ Jaffe, J.
Linked Probabilistic Automata
Mathematical Biosciences 7, 1970, pp. 191-204

/27/ McKay, D.B., Karp, D.P.
Protocol for a Computer Network
IBM Systems Journal, No.1, 1973, pp. 94-105

/28/ Kleinrock, L.
Survey of Analytical Methods in Queuing Networks
Courant Computer Science Symposium 3, Computer Networks, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1972, pp. 185-205

/29/ LeMoli, G.
A Theory of Colloquies
IRIA Proceedings of the 1. European Workshop on Computer Networks, Arles, 1973

/30/ Mezzalira, L., Schreiber, F.
A proposal for a formal description of colloquies as a form of interactions of sequential machines
Instituto di Elettrotecnica ed Elettronica del Politecnico di Milano, Laboratorio di Calcolatori, Rapporto interno n. 73-7, 1973

/31/ Meister, B., Müller, H.R., Rudin, H.R.
On the Optimization of Message Switching Networks
IEEE Transactions on Communications, Vol. COM-20, No.1, Febr. 1972, pp. 8-14

/32/ Merten, E.
Zugriffssynchronisation bei mehrfach geführten Dateien in Rechnernetzen
Diplomarbeit, Universität Karlsruhe, März 1973

/33/ Mihram, G.A.
Simulation, Statistical Foundations and Methodology
Academic Press, New York and London, 1972

/34/ Muntz, R.R.
Poisson Departure Processes and Queuing Networks
IBM Research Report RC 4145, Dec. 1972

/35/ Murphy, J.E.
Resource-allocation with interlock detection in a multi-task system
AFIPS Conference Proceedings Vol. 33
Fall Joint Computer Conference, 1968, pp. 1169-1176

/36/ Nehmer, J.
Dispatcher-Elementarfunktionen für symmetrische Mehrprozessor-DV-Systeme
Dissertation, Universität Karlsruhe, 1973

/37/ Pouzin, L.
Network architectures and components
IRIA Proceedings of the 1. European Workshop on Computer Networks, Arles, 1973

/38/ Schumacher, F.
Simulationsmodell eines Rechnerverbundsystems: Experimententwurf und Validation
Fachgespräche Methodik der rechnergestützten Simulation, Gesellschaft für Kernforschung, Bericht KFK 1845, 1973, S. 319-331

/39/ Senger, R.
Auftrags-Wartezeiten als Maß für die Bedienungsqualität eines Rechnerverbundnetzes
Diplomarbeit, Universität Karlsruhe, Dez. 1972

/40/ Shoshani, A., Bernstein, A.J.
Synchronization in a Parallel-Accessed Data Base
Communications of the ACM, Nov. 1969, Vol.12, No. 11, pp. 604-607

/41/ Silk, D.J.
Queuing model for message-switching networks with constant-length messages
Proc. IEE, Vol.116, No.11, Nov. 1969, pp. 1821-1826

/42/ Stutzman, B.W.
Data Communication Control Procedures
Computing Surveys, Vol.4, No.4, Dec. 1972,
pp. 197-220

/43/ Teroy, J.T., Pinkerton, T.B.
A Comparative Analysis of Disk Scheduling Policies
Communications of the ACM, March 1972, Vol.15, No.3,
pp. 177-184

/44/ Walden, D.C.
A System for Interprocess Communication in a
Resource Sharing Computer Network
Communications of the ACM, April 1972, Vol.15, No.4,
pp. 221-230

/45/ Wettstein, H.
The Implementation of Locking Operations for Critical
Sections in Various Environments
University of Karlsruhe, Germany, 1973

/46/ Whitney, V.K.M.
A Study of Optimal File Assignment and Communication
Network Configuration in Remote-Access Computer
Message Processing and Communication Systems
SEL Technical Report No.48, Systems Engry. Lab,
Dept. of Elect. Engrg., University of Michigan,
Sept. 1970 (Ph.D Dissertation)

/47/ Wohlrapp, E.
Auftragsbeziehungen in Betriebssystemen
Elektronische Rechenanlagen, Heft 2, 1973, S. 66-72

/48/ Zimmermann, M., Elie, M.
Vers une approche systematique des protocoles
sur un reseau d'ordinateurs
Reseau Cyclades, Juillet 1973, SCH 512