

MITTEILUNGSBLATT

DES REGIONALEN
RECHENZENTRUMS ERLANGEN

HERAUSGEBER F. WOLF

Einsatz des ISO-Transaktionsdienstes
zur Echtzeitkommunikation
in verteilten Systemen

Gabriele Dobler

Nr. 64 - ERLANGEN - SEPTEMBER 1993

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Regionalen Rechenzentrum der Universität Erlangen-Nürnberg in der von Herrn Dr. Peter Holleczeck geleiteten Abteilung für Kommunikationssysteme. Ihm bin ich für seine Anregungen und die Unterstützung, die mir die Anfertigung dieser Arbeit neben projektbezogenen Tätigkeiten eröffnete, zu besonderem Dank verpflichtet.

Herr Prof. Dr. Fridolin Hofmann gab mir die Möglichkeit, diese Arbeit an seinem Institut durchzuführen. Für seine aktive Betreuung und seine Bereitschaft zur kritischen Diskussion möchte ich ihm danken.

Herrn Prof. Dr. Klaus Meyer-Wegener gilt mein Dank für das meiner Arbeit entgegengebrachte Interesse, seine Bereitschaft zu Gesprächen, die mir wertvolle Hinweise gaben und dafür, daß er sich trotz des engen zeitlichen Rahmens zur Durchsicht der Arbeit und zur Übernahme des Zweitgutachtens bereit erklärte.

Danken möchte ich auch Herrn Michael Slopianka, Frau Bettina Kauth und allen weiteren Studenten, die mich während der Anfertigung ihrer Studien- oder Diplomarbeiten oder im Rahmen ihrer Tätigkeit als studentische Hilfskraft durch Diskussion und bei der Durchführung der praktischen Arbeiten tatkräftig unterstützt haben.

Frau Sibylle Schweizer und Herrn Dr. Christian Andres danke ich für sorgfältiges Korrekturlesen.

Für die stetige Hilfsbereitschaft und die moralische Unterstützung in schwierigen Situationen, die mir mein Mann Stefan entgegenbrachte, sowie seine Bereitschaft zur fachlichen Diskussion und zum Korrekturlesen möchte ich mich herzlich bedanken.

Inhaltsverzeichnis

0	Kurzfassung	8
1	Einleitung	9
2	Verteilte Transaktionsverarbeitung unter Echtzeitanforderungen	15
2.1	Verteilte Transaktionsverarbeitung	15
2.1.1	Charakterisierung einer verteilten Transaktion	15
2.1.2	Das Kommunikationsmodell konkurrierender Subtransaktionen	17
2.1.3	Commit-Protokolle für verteilte Transaktionen	19
2.2	Verteilte Echtzeitverarbeitung	24
2.2.1	Charakterisierung von Echtzeitanwendungen	24
2.2.2	Schwache Synchronisation von Uhren	27
2.2.3	Starke Uhrensynchronisation	28
2.3	Transaktionsverarbeitung unter Echtzeitbedingungen in einem verteilten System	30
3	Konzepte zur Beschreibung temporaler Beziehungen	32
3.1	Kausale Zeitstrukturen	32
3.2	Aktivitäten sequentieller Prozesse	34
4	Das ISO-TP-Modell	37
4.1	Räumliche Strukturierung einer TP-Anwendung	37
4.1.1	Das TP-Modell gegenüber dem Modell konkurrierender Subtransaktionen	37
4.1.2	Der TP-Dialogbaum einer TP-Anwendung	40
4.1.3	Transaktionsbereiche einer TP-Anwendung	42
42	Sequentielle Bearbeitungsabläufe in einem Transaktionsbereich	45

1.2.1	Kommunikationsphasen einer verteilten Transaktion . .	46
1.2.2	Transaktionskette	48
1.2.3	Unverkettete Transaktionen	49
5	Kausale Zeitstrukturen des TP-Basis-Standards	52
5.1	Kausal-, Zeitstruktur der TP-Schnittstelle eines Transaktionsh, r<idl('s	53
5.1.1	Aufbau eines Transaktionsbereiches für unverkettete Transaktionm1	54
5.1.2	Aufbau eines Transaktionsbereiches für eine Transaktionskette	56
5.1.3	Positiver Abschluß einer Transaktion	57
5.1.4	Negativer Abschluß einer Transaktion	61
5.2	Verfeinerung der kausalen Zeitobjekte des TP-Service-Providers	63
5.2.1	Verfeinerung der Zeitobjekte für den Aufbau eines Transaktionsbereiches	65
5.2.2	Verfeinerung der Zeitobjekte für den positiven Transaktionsabschluß	65
5.3	Kommunikationsintervalle eines Transaktionsbereiches	71
5.3.1	Das Datentransferintervall eines Transaktionsprozesses	72
5.3.2	Das Commit-Phase-1-Intervall eines Transaktionsprozesses	74
5.3.3	Das Commit-Phase-2-Intervall eines Transaktionsprozesses	75
6	Zusammenhänge der Kommunikationsaktivitäten eines Transaktionssystems	78
6.1	Die Menge der Kommunikationsaktivitäten	78
6.1.1	Aktivitäten eines Kommunikationsmanagers	79
6.1.2	Aktivitäten des Kommunikationssystems	81

6.2	Beschreibung der Kommunikationsphasen durch Aktivitätskompositionen	82
6.2.1	Verknüpfung der Aktivitäten eines Kommunikationsmanagers	81
6.2.2	Die Überlagerung von Warteaktivitäten eines Kommunikationsmanagers	88
6.3	Interpretation von Aktivitäten für Monoprozessor-Subsysteme	92
6.3.1	Lokale Abschätzung der Dauer interner Aktivitätskompositionen	92
6.3.2	Einfluß der Übertragungszeiten auf die Wartezeiten	96
7	Grenzen der Echtzeitfähigkeit von ISO-TP	99
7.1	Berücksichtigung von Zeitanforderungen an der TP-Schnittstelle	100
7.2	Berücksichtigung von Zeitanforderungen im TP-Protokoll	103
7.3	Grenzen des Commit-Verfahrens.	107
8	Ein Ansatz zur Behandlung zeitkritischer TP-Anwendungen	109
8.1	Zielsetzung	109
8.2	Voraussetzungen	112
8.3	Lokale Begrenzung des Kommunikationsverhaltens während des Datentransferintervalles	114
8.4	Die Basis für den Transfer zeit.bezogener Daten	117
8.5	Zeitüberwachung des Commit-Phase-1-Intervalles	121
8.6	Abschätzung der Dauer der Commit-Phase-2	125
8.7	Konsequenzen aus der Unterstützung von Zeitbedingungen	130
8.8	Grenzen des Ansatzes	133

9 Realisierung eines ISO-TP-Kommunikationsmanagers für zeitkritische Transaktionen	135
9.1 Entwicklung und Einbettung des Kommunikationsmanagers	136
9.1.1 Grobstruktur des realisierten Kommunikationsmanagers	137
9.1.2 Implementation des Kommunikationsmanagers	140
9.2 Lokale Behandlung von Zeitvorgaben	141
9.2.1 Erweiterung der TP-Schnittstelle	143
9.2.2 Zeitüberwachung im Kommunikationsmanager	144
9.2.3 Begrenzung von kommunikationsabhängigen Zeiträumen durch die Subtransaktion	146
9.2.4 Bewertung der gewonnenen Information	148
9.3 Realisierung eines Protokolls zur prozeßübergreifenden Behandlung von zeitbezogenen Daten	150
9.3.1 Abwicklung des TPTiming-Protokolls	150
9.3.2 Konfliktlösung für das TPTiming-Protokoll	152
9.3.3 Einbindung der TPTiming-Implementation in den Kommunikationsmanager	154
9.3.4 Bewertung des TPTiming-Protokolls	156
10 Ausblick	158
Literaturverzeichnis	162
Index	166

Verzeichnis der Abbildungen

1	Kommunikationsmodell konkurrierender Subtransaktionen . . .	19
2	graphische Symbole zur Darstellung von kausalen Zeitstrukturen	33
3	TP-Schnittstelle	38
4	TP-Dialogbaum	40
5	Transaktionsbereiche in einem Dialogbaum	43
6	Transaktionskette	48
7	unkettete Transaktionsfolge	50
8	Aufbau eines Transaktionsbereiches für unkettete Transaktionen	55
9	Erweitern eines ererbten Transaktionsbereiches	57
10	kausale Abhängigkeit beim Transaktionsabschluß	59
11	Commit-Vorbereitung während der Datentransferphasen . . .	60
12	Kausale Zeitstruktur des TP-Rollback-Dienstes	62
13	Verfeinerung der Zeitobjekte für den Aufbau eines Transaktionsbereiches	66
14	Beispiel des Protokollablaufs beim Transaktionsabschluß, Teil 1.	67
15	Beispiel des Protokollablaufs beim Transaktionsabschluß, Teil 2.	68
16	Verfeinerung der vorgezogenen Commit-Vorbereitung	69
17	Toleranzen synchroner Bearbeitungszeiträume bei Begrenzung des Pufferverhaltens	115
18	Projektionen zur Ermittlung der Werte der Basismatrix	119
19	Projektionen zur Übertragung der maximalen Wartezeit an den Vorgängerprozeß	123
20	Abschätzung des maximalen Zeitbedarfs eines Nachfolgerprozesses	127

21	Interne Strukturierung des Kommunikationsmanagers durch PASS-Prozesse	139
22	PASS-Kommunikationsstruktur des Kommunikationsmanagers mit Zeitüberwachung	142

Verzeichnis der Tabellen

1	Bezugszeitpunkte der TP-Dienste an der TP-Schnittstelle . . .	54
2	Bezugszeitpunkte der TP-Dienste im TP-Service-Provider. . .	64
3	Liste der Aktivitäten	84

0 Kurzfassung

Mit der Entwicklung eines Standards zur Kommunikationsunterstützung verteilter Transaktionen in einer heterogenen Umgebung öffnet sich der Bereich der Transaktionsverarbeitung für ein weites Feld von Anwendungen. Für die Kommunikationsunterstützung von verteilten Anwendungen, die eine zeitlich korrekte Abwicklung von Transaktionen erfordern, ist die vom ISO-Standard für Distributed Transaction Processing (ISO-TP) gebotene Funktionalität nicht ausreichend. Die Abwicklung des zentral koordinierten Commit-Protokolls, welches die Einigung aller an einer verteilten Transaktion beteiligten Prozesse auf einen gemeinsam getragenen Transaktionsabschluß erlaubt, hat für die Prozesse unbestimmte Wartezeiten auf Nachrichten zur Folge. Solche unbegrenzten Verzögerungen sind in einer Transaktion, deren korrekte Abwicklung von zeitlichen Faktoren abhängt, nicht tragbar. Der hier entwickelte Ansatz zur Berücksichtigung von verteilten Zeitbedingungen gestattet eine mit den Transaktionseigenschaften verträgliche, zeitlich korrekte Bearbeitung verteilter Transaktionen in einer Echtzeitumgebung. Die Behandlung von Zeitanforderungen ist zunächst auf solche Maßnahmen beschränkt, die auf den an der Transaktion mitwirkenden Rechnern der verteilten Umgebung lokal durchführbar sind. Der Ausschöpfung aller einem Prozeß zur Verfügung stehenden Möglichkeiten zur Gewährleistung der Einhaltung von Zeitbedingungen wird Vorrang gegenüber einer verteilten Behandlung eingeräumt, um anwendungsspezifischen Konfliktlösungen nicht vorzugreifen. Die Erweiterung dieser Möglichkeiten zur Zeitbegrenzung durch ein zusätzliches Protokoll greift deshalb erst während des Transaktionsabschlußverfahrens, wenn keine anwendungsspezifische Kommunikation mehr möglich ist. Dieses Protokoll beruht auf der schrittweisen Übertragung der Verantwortung für die Einhaltung von Zeitbedingungen an den Koordinatorprozeß des Commit-Protokolls. Dadurch läßt sich eine lückenlose Überwachung der Zeitanforderungen aller an der Transaktion beteiligten Prozesse erreichen. Bei Verletzung einer Zeitanforderung kann der Transaktionsabbruch oder eine anwendungsspezifische Reaktion durch den Prozeß eingeleitet werden, dem die Verletzung der Zeitbedingung signalisiert wird.

1 Einleitung

Die Entwicklung verteilter Systeme in der Datenverarbeitung war ein grundlegender Schritt zur Erschließung neuer Anwendungsbereiche. Damit wurden Anwendungen erfaßt, für die durch die Vorgabe einer räumlichen Struktur eine verteilte Verarbeitung unvermeidbar ist. In einigen Fällen wird durch Strukturierung einer Anwendung gemäß ihren Anforderungen der Einsatz hoch spezialisierter Hardware und Peripherie begünstigt, so daß die Vorteile einer verteilten Verarbeitung gegenüber den Nachteilen der Dezentralisierung überwiegen. Der Begriff "verteilt System", der dieser Entwicklung zugrundeliegt, wird aus der in [Tan92] angegebenen Definition übernommenen:

" A distributed system. is one that runs on a collection of machines that do not have shared memory, yet looks to its users like a single computer."

Die Zuordnung einer Anwendung zu einem Anwendungsbereich erfolgt in Abhängigkeit von den vorliegenden Anforderungen. Letztere müssen durch geeignete Systemunterstützung erfüllt werden. Für verteilte Systeme besteht ein Teil dieser Unterstützung in der Bereitstellung anwendungsorientierter Kommunikationsprotokolle. Die Standardisierung herstellerunabhängiger Kommunikationsdienste erleichtert den Einsatz heterogener verteilter Systeme und leistet damit einen Beitrag zur Erweiterung des Spektrums verteilter Anwendungen.

Im Zuge dieser Entwicklung wurde für verteilte Anwendungen, die konkurrierend auf globale Datenbestände oder Betriebsmittel zugreifen, ein ISO¹-Standard, Distributed-Transaction-Processing, (ISO-TP), [TPM90], [TPS90] und [TPP90], zur Kommunikationsunterstützung für den Bereich der verteilten Transaktionsverarbeitung ausgearbeitet. Damit wird das Ziel der Transaktionsverarbeitung, die zusammenhängende Ausführung mehrerer Operationen als atomarer Bearbeitungsschritt, in einem verteilten System realisierbar. Das ISO-TP-Protokoll ermöglicht den Erhalt eines konsistenten verteilten Datenbestandes und berücksichtigt dabei insbesondere Ausfälle von Kommunikationsverbindungen und Rechnern des verteilten Systems. Zur

¹ISO:International Standard Organization

ordnungsgemäßen Bearbeitung einer verteilten Transaktion müssen die Rechner des verteilten Systems neben der korrekten Abwicklung des transaktionsorientierten Kommunikationsprotokolls lokale Transaktionsunterstützung gewährleisten. Aus diesem Zusammenhang leitet sich der für diese Arbeit grundlegende Begriff des verteilten Transaktionssystems ab:

Verteiltes Transaktionssystem: Ein verteiltes System, dessen Rechner lokale Transaktionsunterstützung anbieten und transaktionsorientierte Kommunikationsmechanismen bereitstellen, realisiert ein verteiltes Transaktionssystem.

In [MW86] werden Transaktionssysteme unter den Begriff der Echtzeitsysteme eingeordnet und gegenüber der Stapelverarbeitung abgegrenzt. Echtzeitsysteme heben sich aus der Menge der Datenverarbeitungssysteme durch die Berücksichtigung der Zeit als kritische Größe ab. Im Vordergrund steht die Erfüllung von Anforderungen der Anwendung bezüglich der Rechtzeitigkeit oder Gleichzeitigkeit von Operationen.

Ein Echtzeitsystem wird in [BW90] als:

"any information processing activity or system which has to respond to externally generated input stimuli within a finite and specified period"

definiert. Damit wird die Gewährleistung eines definierten Zeitverhaltens als wesentliches Kriterium für die Echtzeitfähigkeit eines Systems angegeben. Die Unterscheidung von harten und weichen Zeitanforderungen charakterisiert den Einfluß der Zeit auf die korrekte Abwicklung von Operationen durch das Echtzeitsystem.

Durch die Verletzung einer weichen Zeitanforderung sinkt die Relevanz der Ergebnisse der durchgeführten Operationen, ihre Korrektheit bleibt darnach unberührt. Ein Beispiel dafür ist die Ausführung einer Transaktion, die von einer Person interaktiv bedient wird. In diesem Zusammenhang treten Zeitanforderungen in Bezug auf das Antwortzeitverhalten des Transaktionssystems auf, die aus der begrenzten Geduld des menschlichen Bedieners resultieren, deren Überschreitung jedoch keinen Einfluß auf die Korrektheit des Ergebnisses besitzt.

Im Gegensatz dazu sind Ergebnisse von Operationen, die unter Verletzung von harten Zeitanforderungen ermittelt wurden, möglicherweise falsch. Ein Blick zu dem Anwendungsbereich von Prozeßrechensystemen, die die Steuerung technischer Prozesse übernehmen, läßt die harten zeitlichen Bedingungen erkennen. Für die Roboter einer Fertigungsstraße liegen z.B. definierte Zeitrahmenbedingungen zur Erfüllung ihrer Operationen vor. Eine Verletzung des Zeitrahmens an einer Station stört den zeitlichen Ablauf und kann dazu führen, daß das entstehende Produkt fehlerhaft ist.

Die Bearbeitung verteilter Anwendungen mit Echtzeitanforderungen wirft die Problematik der Synchronisation verteilter Uhren auf. Die Synchronisation der Uhren eines verteilten Systems beruht auf Kommunikationsmechanismen zwischen den Rechnern des verteilten Systems. Die derzeit bekannten Ansätze unterscheiden sich im Aufwand, in der Genauigkeit und im Umgang mit Ausfällen von Kommunikationsverbindungen und Rechnern.

Der Anspruch in der verteilten Transaktionsverarbeitung an die Echtzeitfähigkeit des verteilten Transaktionssystems orientiert sich an der Art der im einzelnen vorliegenden Zeitanforderungen. An dem Beispiel einer Reisebuchung in einem rechnergestützten Reisebüro lassen sich Zeitanforderungen mit unterschiedlichem Wirkungsbereich verdeutlichen.

Der Rechner eines Reisebüros sei mit Platzbuchungssystemen von Fluggesellschaften und Hotelketten zu einem verteilten Transaktionssystem verknüpft. Die Buchung einer Reise bildet eine verteilte Transaktion, in der die Wünsche des Kunden bezüglich Reiseziel und Aufenthaltszeitraum in entsprechenden Reservierungen umgesetzt werden. Der Anfangszeitpunkt einer Reisebuchung wird vom Kunden des Reisebüros bestimmt. Dies zeigt, daß für den Rechner, auf welchem die verteilte Transaktion initiiert wurde (in diesem Fall der Rechner des Reisebüros), eine lokale Zeitanforderung mit lokaler Reichweite bestehen kann. Durch die Öffnungszeiten des Reisebüros oder den Wunsch des Kunden, die Reisebuchung innerhalb seiner Mittagspause abzuschließen, sind Zeitanforderungen bezüglich der Dauer der gesamten Transaktion vorgegeben. Diese liegen nur in dem Rechner vor, in welchem die Transaktion initiiert wurde. Sie umfassen aber das Zeitverhalten der gesamten Transaktion und beziehen alle beteiligten Rechner mit ein. Die Betreiber der Rechenanlagen für Hotelreservierung und Flugbuchung können

ihre Dienstleistung über transaktionsorientierte Kommunikationsprotokolle mehreren Reisebüros gleichzeitig anbieten. Daraus können Konkurrenzsituationen zwischen mehreren verteilten Transaktionen, die verschiedene Kunden bedienen, entstehen. Im Interesse der Betreiber der Rechenanlagen liegt es, die Dauer der Beteiligung an einer Transaktion zu begrenzen und damit die in dieser Transaktion benutzten Datenbestände anderen Reisebüros möglichst bald wieder zugänglich zu machen. In diesem Fall unterliegt die Transaktion verteilten unkoordinierten Zeitanforderungen.

Mit dem Beispiel einer Reisebuchung wurde gezeigt, daß in einer verteilten Transaktion die Behandlung verteilter weicher Zeitanforderungen wünschenswert ist. An dem aus [SLW89] übernommenen Beispiel einer chemischen Fabrik wird die Notwendigkeit der Behandlung von harten Zeitanforderungen deutlich.

Das Beispiel beschreibt eine chemische Fabrik, in der die Fertigstellung eines Produktes auf einem Fließband erfolgt. Auf jeden auf dem Fließband transportierten Rohling werden nacheinander zwei chemische Substanzen aufgetragen. Das Aufbringen beider Substanzen muß innerhalb von 10 Sekunden abgeschlossen sein, um den gleichmäßigen Transport beizubehalten. Wenn nur eine Aktion innerhalb der geltenden Zeitvorgaben ausgeführt werden konnte, muß eine Neutralisierung eingeleitet werden. Das Auftragen der Substanzen wird von je einem Prozeß überwacht. Ein weiterer Prozeß übernimmt die Zeitüberwachung. Als Anforderung ergibt sich in diesem Beispiel die atomare Ausführung beider Aufträge innerhalb definierter Zeitgrenzen.

Die Berücksichtigung von verteilten Zeitanforderungen bildet die Voraussetzung für die frühzeitige Erkennung von daraus resultierenden Konflikten. Die in verteilten Transaktionssystemen vorgegebenen Maßnahmen beim Ausfall von Rechnern oder Kommunikationsverbindungen sind auf den Erhalt der Transaktionseigenschaften abgestimmt. Die Behandlung dieser Ausfallsituationen begrenzt die Möglichkeiten zur Gewährleistung eines definierten Zeitverhaltens bei der Abwicklung einer verteilten Transaktion.

Ziel dieser Arbeit ist es, die Behandlung verteilter Zeitanforderungen in dem durch den ISO-TP-Standard definierten transaktionsorientierten Kommunikationsprotokoll vorzusehen und die Grenzen der Echtzeitfähigkeit abzustecken. Damit soll eine Kommunikationsgrundlage für die Berücksichtigung

von Zeitanforderungen in verteilten Transaktionssystemen geschaffen werden.

In Kapitel 2 wird die Zielsetzung von Transaktionsanwendungen und Anwendungen aus dem Bereich der Echtzeitverarbeitung erläutert und auf die besonderen Bedingungen in einer verteilten Umgebung eingegangen. Für beide Bereiche werden übliche Verfahren zur Unterstützung verteilter Anwendungen aufgeführt.

Für die Beschreibung zeitkritischer Abläufe werden in Kapitel 3 zwei Ansätze vorgestellt, die über räumliche Grenzen hinweg die Darstellung zeitlicher Zusammenhänge ermöglichen.

Die Einordnung des ISO-TP-Standards in den Bereich der verteilten Transaktionsverarbeitung wird in Kapitel 4 vorgenommen. Die Beschreibung der dem Standard zugrundeliegenden Modellvorstellungen wurde auf die zur Analyse des Zeitverhaltens notwendigen räumlichen und zeitlichen Strukturierungskonzepte beschränkt.

In Kapitel 5 sind die für die Anwendung zur Kommunikation notwendigen Systemdienste und die vorliegenden Reihenfolgebestimmungen angegeben. Die Beschreibung der daraus ableitbaren kausalen Beziehungen erfolgt unter Verwendung des ersten, in Kapitel 3 vorgestellten Ansatzes. Auf der Grundlage dieses Beschreibungsmodells werden die zur Realisierung der Systemdienste vorgegebenen Bearbeitungsschritte durch eine Verfeinerung der kausalen Zusammenhänge herausgearbeitet. Mit einer Gliederung der sich ergebenden Strukturen in kausale Zeitintervalle unter Berücksichtigung der räumlichen Verteilung schließt das Kapitel.

In Abschnitt 6 werden aufbauend auf der Intervallstruktur die von einem System zu erbringenden Leistungen charakterisiert. Im Gegensatz zur systemübergreifenden Analyse im vorangehenden Kapitel wird hier der Blick auf die Einbettung einer Implementierung in ein System der verteilten Umgebung gerichtet. Die Darstellung greift auf den zweiten, in Kapitel 3 vorgestellten Ansatz zurück. Das Zeitverhalten einer Implementation wird durch die Einbettung ins System, den internen Ablauf und externe Abläufe charakterisiert. In die Bewertung des Zeitbedarfs für externe Abläufe gehen die Übertragungszeiten ein.

Die Grenzen der Echtzeitfähigkeit des ISO-TP-Standards zeigen sich für die Anwendung in der fehlenden Berücksichtigung von Zeitanforderungen. In Kapitel 7 werden daher zunächst die zur Zeitüberwachung im Vergleich der Kommunikation vorliegenden Informationsdienste zwischen Systemunterstützung und Anwendung aufgezeigt. Die Grenzen, die sich innerhalb einer TP-Implementierung, wie der durch den Standard vorgeschriebenen Definition des Übertragungsprotokolls ergeben, werden anschließend betrachtet. Die in einer verteilten Umgebung vorliegenden, prinzipiell unlösbaren Problemstellungen werden dabei ausgeklammert und im 7.7. Abschnitt diskutiert.

In Kapitel 8 wird ein Ansatz zur Unterstützung der Behandlung von zeitbezogenen Daten in einer verteilten Transaktion vorgestellt. Dieser Ansatz verfolgt zwei Ziele. Zum einen wird die zeitliche Bewertung der in Kapitel 5 eingeführten kausalen Zeitintervalle in der Anwendung ermöglicht. Zum anderen wird die Verantwortung für die Einhaltung von Zeitanforderungen in die Systemdienste verlagert.

Der Ansatz wurde für eine vorgegebene Klasse von Zeitanforderungen realisiert. Kapitel 9 beschreibt die Struktur der vorliegenden Implementation und gibt einen Einblick in die Realisierung von Verfahren zur Behandlung von Zeitanforderungen.

Die Arbeit schließt mit einem Ausblick, der weitere offene Gesichtspunkte aufzeigt und auf Ansätze verweist, die weitere interessante Aspekte für die hier betrachtete Problemstellung behandeln.

2 Verteilte Transaktionsverarbeitung unter Echtzeitanforderungen

Die Anforderungen von Anwendungen aus den Bereichen der Transaktionsverarbeitung und der Echtzeitverarbeitung weisen nur geringfügige Gemeinsamkeiten auf. Die Überschneidung der Anforderungsprofile besteht in kurzen Antwortzeiten für eine Transaktion gegenüber definiertem Zeitverhalten eines Echtzeitsystems und in der Gewährleistung einer hohen Verfügbarkeit. Mit der Entwicklung verteilter Anwendungen entsteht in beiden Bereichen eine Abhängigkeit von dem zur Nachrichtenübertragung eingesetzten Kommunikationssystem. Die Zielvorstellungen und grundlegende Verfahren zur Unterstützung von verteilten Anwendungen aus beiden Bereichen werden in den folgenden Abschnitten vorgestellt.

2.1 Verteilte Transaktionsverarbeitung

Mit dem Begriff der Transaktionsverarbeitung verbindet sich die Vorstellung, mehrere Operationen als unteilbare Aktion auszuführen. Der Zugriff auf die von den Operationen betroffenen Objekte ist dabei mehreren Anwendungen gleichermaßen möglich, sodaß zum Erhalt der Konsistenz der gegenseitige Ausschluß gewährleistet sein muß. Zur Charakterisierung der Transaktionsverarbeitung in einer verteilten Umgebung wird zunächst der Begriff der verteilten Transaktion erläutert. Im Anschluß daran wird auf ihre Einbettung in ein verteiltes Transaktionssystem eingegangen. Dabei steht der Kommunikationsaspekt im Vordergrund. Mit der Vorstellung von Kommunikationsmechanismen, die zur Gewährleistung der Transaktionseigenschaften in verteilten Transaktionen notwendig sind, schließt das Kapitel.

2.1.1 Charakterisierung einer verteilten Transaktion

Der Einführung des Begriffes "verteilte Transaktion" wird zunächst die Definition des Transaktionsbegriffes aus [GR93] vorangestellt:

"A **transaction** is a collection of operations on the physical and abstract. application state."

Die Festlegung der **von** einer Transaktion berührten Objekte und die Art und Reihenfolge der auszuführenden Operationen liegt im Verantwortungsbereich der Anwendung. Die Anforderungen an Transaktionsverarbeitung bestehen in der Gewährleistung der ACID-Rules, deren Definition ebenfalls aus [GR93] übernommen wurde:

"Atomicity. A transaction's changes to the state are atomic: either all happen or none happen. These changes include database changes, messages, and actions on transducers."

"Consistency. A transaction is a correct transformation of the state. The actions taken as a group do not violate any of the integrity constraints associated with the state. This requires that the transaction be a correct program."

"Isolation. Even though transactions execute concurrently, it appears to each transaction, T, that others are executed either before T or after T, but not both."

"Durability. Once a transaction completes successfully (commits), its changes to the state survive failures."

Der Begriff der verteilten Transaktion schränkt den bereits eingeführten Transaktionsbegriff auf die folgende Definition ein:

Verteilte Transaktion: Eine verteilte Transaktion besteht aus einer Menge von Operationen auf Objekten, die von verschiedenen Rechnern eines verteilten Transaktionssystems verwaltet werden.

Zu der Vorgabe von Art und Reihenfolge der Operationen durch die Anwendung kommt für verteilte Transaktionen die Untergliederung der Menge der Operationen in Teilmengen hinzu, wobei die Operationen einer Teilmenge auf einem Rechner des verteilten Transaktionssystems lokal ausgeführt werden. Darauf aufbauend wird nun der Begriff "Subtransaktion" eingeführt:

Subtransaktion: Eine Subtransaktion setzt sich aus einer Teilmenge der anwendungsspezifischen Operationen einer verteilten Transaktion zusammen, die auf einem Rechner des verteilten Transaktionssystems lokal ausgeführt werden.

Damit lässt sich eine verteilte Transaktion als Menge von Subtransaktionen beschreiben. Die Subtransaktionen sind als anwendungsspezifische Programme zu betrachten, die zum Zweck der Abwicklung einer verteilten Transaktion miteinander kommunizieren. Den dabei einzusetzenden Kommunikationsmechanismen liegt eine Anordnung der Subtransaktionen in einer Kommunikationsstruktur zugrunde. Eine verteilte Transaktion wird von einer transaktionsinitiierenden Subtransaktion eröffnet. Sie kann über Kommunikationsdienste andere anwendungsspezifische Programme als Knoten in die Kommunikationsstruktur mit einbeziehen, die nun wiederum Subtransaktionen der verteilten Transaktion verkörpern. Durch entsprechendes Verhalten der neu integrierten Subtransaktionen lässt sich die Kommunikationsstruktur zu einem gerichteten azyklischen Graphen mit der transaktionsinitiierenden Subtransaktion als Wurzel erweitern. Die anderen Subtransaktionen sind als der Wurzel untergeordnete Knoten in den Graphen eingebunden. Die gerichteten Kanten repräsentieren die Kommunikationsverbindungen, wobei die Richtung der Kante nur die Ordnung der Knoten in der Kommunikationsstruktur verdeutlicht aber beide Kommunikationsrichtungen zulässt.

2.1.2 Das Kommunikationsmodell konkurrierender Subtransaktionen

Analog zu der Konkurrenzsituation beim Zugriff auf gemeinsame Objekte durch verschiedene auf einen Rechner beschränkte Transaktionen ergeben sich in der verteilten Transaktionsverarbeitung Konkurrenzsituationen zwischen den von einem Rechner unterstützten Subtransaktionen. Wenn ein Rechner mehrere Subtransaktionen der selben verteilten Transaktion unterstützt, entsteht eine Konkurrenzsituation innerhalb der verteilten Transaktion. Solche Situationen lassen sich anhand systemweit eindeutiger Transaktionsbezeichner erkennen, die für alle Subtransaktionen derselben verteilten Transaktion übereinstimmen. Die oben beschriebene gerichtete Kommunikationsstruktur bildet die Grundlage für die Weitergabe des im Wurzelknoten generierten Transaktionsbezeichners an alle untergeordneten Knoten. Eine Verklemmung kann dann in dem betroffenen Subsystem durch lokale Maßnahmen behandelt werden.

Durch das nachfolgend eingeführte Kommunikationsmodell konkurrierender

Subtransaktionen soll die Konkurrenzsituation und von Pines Rechnen unterstützten Subtransaktionen in Bezug auf Kommunikationsdienste verdeutlicht werden. Es beschreibt ein verteiltes Transaktionssystem als Verteiltes System, dessen Rechner über ein Kommunikationssystem verbunden sind. Das Kommunikationssystem läßt eine hierarchische Vermaschung zu. Jeder Rechner realisiert ein Subsystem des verteilten Transaktionssystems und stellt damit die zur verteilten Transaktionsverarbeitung notwendige Systemunterstützung bereit.

Dieses Modell orientiert sich an dem in [CR91] definierten X/Open distributed Transaction Processing Model. Letzteres grenzt innerhalb eines Subsystems ein

„anwendungsspezifisches Programm vom

Transaction-Manager, von dem

- Resource-Managern und vom

Communication-Manager

ab. Damit werden die lokale Transaktionsunterstützung, der Zugriff auf Betriebsmittel und der Zugriff auf das Kommunikationssystem als Systemdienste einer Subtransaktion zur Verfügung gestellt. Die Konkurrenzsituation wird dabei nicht erfaßt.

Der in Abbildung 1 dargestellte Ausschnitt Pines' für ein Transaktionssystem zeigt das Kommunikationsmodell konkurrierender Subtransaktionen. Es bezieht alle von einem Subsystem unterstützten Subtransaktionen mit ein, abstrahiert aber vom Zugriff auf lokale Betriebsmittel und die lokale Transaktionsunterstützung. Damit wird der konkurrierende Zugriff auf das Communication Management hervorgehoben. Die beiden dargestellten Subsysteme unterstützen je zwei bezüglich eines Kommunikationsmanagers konkurrierende Subtransaktionen.

Kommunikationsmanager: Der Kommunikationsmanager bietet innerhalb eines Subsystems die Kommunikationsdienste an und regelt

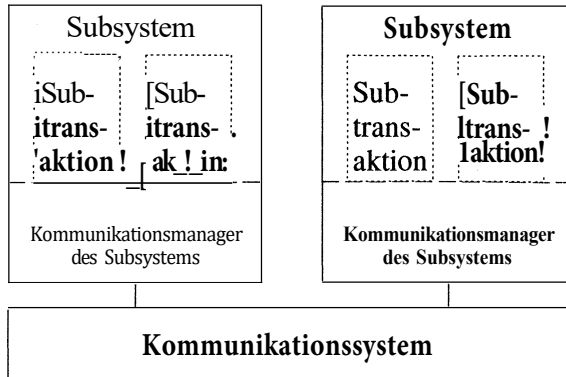


Abbildung 1: Kommunikationsmodell konkurrierender Subtransaktionen

den Zugang zum Kommunikationssystem. Er stellt insbesondere geeignete Kommunikationsmechanismen bereit, die die korrekte Terminierung verteilter Transaktionen ermöglichen. Sie beinhalten verteilte Einigungsprotokolle, Rücksetzverfahren und Recoverymechanismen.

2.1.3 Commit-Protokolle für verteilte Transaktionen

Für den Abschluß einer Transaktion ergeben sich aus der Forderung nach atomarer Ausführung aller Operationen und konsistenzerhaltender Übergänge genau zwei Terminierungszustände:

- Mit dem positiven Ausgang wird die erfolgreiche Vollendung aller Operationen und damit eine Überführung aller betroffenen Objekte in einen konsistenten Folgezustand erreicht.
- Beim negativen Abschluß wird die Transaktion ohne Wirkung auf die durch die Operationen berührten Objekte verlassen.

In einer auf einen Rechner beschränkten Transaktion wird die Entscheidung über den Transaktionsabschluß durch ein lokales Commit-Verfahren ermittelt, an welchem die anwendungsspezifische Komponente, der Transaction-Manager und alle involvierten Resource-Manager beteiligt sind.

Der positive Transaktionsabschluß setzt dabei die Übereinstimmung aller genannten Komponenten zur positiven Vollendung voraus. Wenn mindestens eine Komponente den positiven Ausgang ablehnt, wird der negative Abschluß durchgesetzt.

In einer verteilten Transaktion dehnt sich das Commit-Verfahren ihrerseits auf an der Transaktion beteiligten Subsysteme aus. Die Kommunikationsmanager der Subsysteme stellen ein Commit-Protokoll bereit, welches die Ermittlung und Durchsetzung einer globalen Terminierungsentscheidung unterstützt. Die Kommunikationsmanager werden damit zum einen in die Ermittlung der lokalen Bereitschaft zur Terminierung auf den zugehörigen Subsystemen einbezogen. Zum anderen ermöglichen sie die Verknüpfung der lokalen Terminierungsaussagen aller Subsysteme. Durch fortlaufende Protokollierung des Commit-Zustandes in jedem Subsystem auf stabilen Speicher wird die Fehlertoleranz für Ausfälle von Subsystemen und Ausfälle von Verbindungen einschließlich Netzwerkpartitionierungen erreicht, sofern keine protokollierte Information verloren geht.

Die Koordination des Commit-Protokolls bildet eine zentrale Aufgabe innerhalb einer Transaktion. Sie wird von einer im folgenden als Commit-Koordinator bezeichneten Subtransaktion wahrgenommen.

Gegenüber den rein lokalen Entscheidungskriterien, die für eine auf einen Rechner beschränkte Transaktion vorliegen, ergibt sich bei verteilten Transaktionen die Möglichkeit, daß jede Subtransaktion ihre Bereitschaft zur Terminierung von lokalen Bedingungen *und* vom Terminierungszustand anderer Subtransaktionen abhängig macht, [Bü89]. Die lokale Bereitschaft eines Subsystems zur Terminierung hängt unter anderem von der lokal unterstützten Subtransaktion ab und kann daher mittelbar den Zustand anderer Subtransaktionen einbeziehen.

Jedes Subsystem durchläuft 3 Terminierungsabschnitte:

1. Beschluß und Verkündung der lokalen Bereitschaft zur Terminierung
2. Ermittlung der globalen Terminierungsentscheidung
3. Lokale Verwirklichung der globalen Entscheidung

In den Kommunikationsprotokollen zur Unterstützung des Transaktionsabschlusses ist die Dreiteilung meist nicht vollständig erkennbar. Die einzelnen Abschnitte lassen sich zum Teil lokal, d.h. ohne Kommunikationsaktionen verwirklichen. Nachfolgend werden zwei Ansätze verteilter Commit-Verfahren vorgestellt.

Das **2-Phasen-Commit-Protokoll** verwirklicht die drei Terminierungsabschnitte in zwei Kommunikationsphasen.

Abschnitt 1: Die Subtransaktion, die das Commit-Verfahren für die gesamte verteilte Transaktion initiiert und daher als Commit-Initiator-Subtransaktion bezeichnet wird, fordert alle anderen Subtransaktionen durch eine Nachricht zur Ermittlung der lokalen Terminierungsaussage auf. Das Commit-Koordinator-Subsystem wartet auf die Nachrichten aller in die Transaktion involvierten Subsysteme, die deren lokal ermittelte Bereitschaft zum Transaktionsabschluß signalisieren. Alle Subtransaktionen, die eine Aufforderung empfangen haben und die Commit-Initiator-Subtransaktion leiten die lokale Ermittlung der Terminierungsbereitschaft ein, deren Ergebnis anschließend an den Commit-Koordinator übermittelt wird.

Abschnitt 2: Im Commit-Koordinator wird die globale Entscheidung abhängig von den Mitteilungen aller anderen Subsysteme und der lokalen Bereitschaft bestimmt. Die Entscheidung wird lokal bekannt gemacht und allen betroffenen Subsystemen übermittelt. In jedem dieser Subsysteme wird die Entscheidung aus der vom Commit-Koordinator zugestellten Nachricht ermittelt und lokal verbreitet.

Abschnitt 3: Jedes Subsystem verwirklicht die globale Entscheidung lokal. Durch Übermittlung einer Bestätigung an den Commit-Koordinator wird die Durchsetzung der globalen Entscheidung von der Subtransaktion bekanntgegeben. Der Commit-Koordinator erkennt die Terminierung der gesamten Transaktion.

Die Kommunikationsphasen ergeben sich aus zwei Anfrage-Antwort-Zyklen, die in jedem Subsystem mit Ausnahme des Commit-Initiator-Subtransaktion unterstützenden Subsystems und des Commit-Koordinator-Subsystems durchlaufen werden. Der erste Abschnitt verwirklicht die erste

Kommunikationsphase des 2-Phasen-Commit-Protokolls. f)j,, Abschnitt.t." 2 und 3 werden zur zweiten KommunikationsphasP zusammengefaßt.

Das 3--Phasen-Commit-Protokoll verwirklicht. di" <lr'i 'f,-rminicrllngsa.b-schnitte in drei Kommunikationsphasen.

Abschnitt 1: Dieser Abschnitt entspricht dPm Abschnitt. 1 des 2-Phasen-Commit-Protokolls.

Abschnitt 2: Das Commit-Koordinator-Subsystem berechnet die globale Entscheidung und übermittelt sie an alle anderen Subsysteme. Danach wartet es darauf, daß alle Subsysteme den Empfang der Entscheidung bestätigen. Jedes Subsystem mit Ausnahme des Commit-Koordinators empfängt die globale Entscheidung als Nachricht vom Commit-Koordinator. Es bestätigt zunächst dem Commit-Koordinator die Kenntnis der Entscheidung und wartet dann auf die Aufforderung zur Entscheidungsdurchsetzung.

Abschnitt 3: Der Commit-Koordinator setzt an alle anderen Subsysteme eine Aufforderung zur Durchsetzung der Entscheidung ab und wartet auf die Bestätigungsmeldungen der Entscheidungsdurchsetzung der Subsysteme. Außerdem setzt er die globale Entscheidung auch lokal durch. Die anderen Subsysteme empfangen die Aufforderung zur Entscheidungsdurchsetzung und setzen nach der lokalen Verwirklichung eine Bestätigung an den Commit-Koordinator ab. Mit dem Empfang aller Bestätigungen erkennt der Commit-Koordinator den Abschluß der gesamten Transaktion.

Die beiden vorgestellten Verfahren unterscheiden sich im Kommunikationsaufwand und in den Möglichkeiten zur korrekten Terminierung einzelner Subtransaktionen beim Ausfall von Subsystemen und Verbindungen.

Der Kommunikationsaufwand des 3-Phasen-Commit-Protokolls besitzt gegenüber dem 2-Phasen-Commit-Protokoll einen zusätzlichen Kommunikationszyklus, der alle Subtransaktionen erfaßt.

Das drei-Phasen-Commit-Protokoll erlaubt die Terminierung für alle intakten Subtransaktionen, wenn keine Partitionierung des verteilten Transaktionssystems eingetreten ist, die eine Partitionierung der verteilten Trans-

aktion nach sich zieht. Das 2-Phasen-Commit-Protokoll kann dagegen eine Blockierung funktionsfähiger Subtransaktionen verursachen, die auf das Eintreffen der globalen Entscheidung warten, wenn während der Commit-Protokollabwicklung ein Subsystem ausfällt. An dieser Stelle wird insbesondere für das 2-Phasen-Commit-Protokoll die Forderung nach hoher Verfügbarkeit aller Komponenten eines verteilten Transaktionssystems deutlich. Die in einer Transaktion bearbeiteten Objekte unterliegen einer Zugriffssperre für konkurrierende Subtransaktionen. Der Ausfall einer am Transaktionsabschluß beteiligten Komponente kann zu einer nicht begrenzbaren Belegungsdauer der Objekte führen, die die Abwicklung anderer Transaktionen verhindert.

Zum Nachziehen des Transaktionsabschlusses in den ausgefallenen Komponenten kann in beiden Protokollen das Erfragen der globalen Entscheidung bei einem beteiligten intakten Subsystem notwendig sein.

Neben diesen beiden grundlegenden Verfahren sind weitere Variationen möglich, die eine Optimierung des Nachrichtenflusses erlauben. Die zentrale Einleitung des 2- oder 3-Phasen-Commit-Protokolls durch eine ausgewählte Subtransaktion, die den Commit-Initiator realisiert, kann z.B. durch eine dezentrale Commit-Einleitung ersetzt werden. Die Subtransaktionen übertragen dann ihre Terminierungsbereitschaft unaufgefordert an den Commit-Koordinator.

Weitere Optimierungen des Nachrichtenflusses ergeben sich in zentralen Commit-Protokollen für eine mehrstufige hierarchische Anordnung der Subtransaktionen in einer Kommunikationsstruktur. Die dezentrale Einleitung des Transaktionsabschlusses geht in diesem Fall von den Subtransaktionen der untersten Hierarchieebene aus. Die Nachrichten einer Hierarchiestufe, die an den Commit-Koordinator gerichtet sind, werden von den Subsystemen der nächst höheren Ebene gesammelt und vor der Weiterleitung nach oben unter Einbeziehung der eigenen lokalen Terminierungsbereitschaft reduziert. Mitteilungen, die vom Commit-Koordinator ausgehen, werden in jeder Hierarchieebene vervielfältigt und dann an die Subsysteme der nächst tieferen Ebene verteilt.

Weitere Varianten und Optimierungen werden in [MSF83], [Bü89] und [SLW89] vorgestellt.

2.2 Verteilte Echtzeitverarbeitung

Ziel der Echtzeitverarbeitung ist die Bearbeitung von Operationen innerhalb eines vorgegebenen Zeitrahmens. Echtzeitsysteme listen für Anwendungen mit dieser Problemstellung Unterstützung durch Gewährleistung eines definierten Zeitverhaltens.

Im nächsten Abschnitt werden die Anforderungen an Echtzeitsysteme präzisiert. Die Eigenschaften eines Echtzeitsystems, welches den Anforderungen gerecht wird, leiten sich daraus ab.

Die Problematik der Interpretation von zeitbezogenen Angaben in einer verteilten Umgebung macht den Einsatz von Synchronisationsmechanismen notwendig. Unter der Annahme, daß jedem Prozessor eine lokale Uhr zugeordnet ist, sind Zeitbedingungen für die lokal zu bearbeitenden Operationen relativ zur lokalen Uhr kontrollierbar. Alle darüber hinausgehenden Zeitüberwachungen beruhen entweder auf Annahmen über den Gleichlauf der Uhren oder auf einem Synchronisationsverfahren für verteilte Uhren. Zwei grundlegende Ansätze werden abschließend vorgestellt.

2.2.1 Charakterisierung von Echtzeitanwendungen

Echtzeitanwendungen haben die rechnergestützte Verarbeitung von Ereignissen, die einem durch die reale Umwelt geprägten Ablauf angehören, zum Ziel. Die rechnergestützte Bearbeitung der Ereignisse bildet die Grundlage für die Einleitung einer situationsbezogenen Reaktion. Die Anforderungen an die Systemunterstützung betreffen die Fähigkeit, eine mit dem vorgegebenen Ablauf schritthaltende Bearbeitung der Ereignisse zu gewährleisten.

Der zeitliche Verlauf einer Echtzeitanwendung ist somit durch äußere Einflüsse bestimmt. Der Zeitraum für die rechtzeitige Bearbeitung der Ereignisse läßt sich in einer Echtzeitanwendung durch Angabe zeitlicher Bedingungen beschreiben. Diese Bedingungen liegen in der Form

- absoluter Angaben im direkten Bezug zu einer physikalischen Uhr und / oder

- relativer Angaben, die zeitliche Abstände zu den bezüglich einer physikalischen Uhr ermittelbaren Eintrittszeitpunkten von Ereignissen beschreiben,

vor. Der gewünschte Ausführungszeitraum kann durch absolute oder relative Angaben von

- oberen Schranken, die den spätest möglichen Zeitpunkt für ein Ereignis vorgeben, und / oder
- untere Schranken, die den frühest möglichen Eintrittszeitpunkt festlegen,

definiert sein. Durch die logische Verknüpfung mehrerer Zeitbedingungen lassen sich Zeitrahmen für die Bearbeitung zeitkritischer Abschnitte abstecken. Die Definition der Randbedingungen für eine zeitlich synchronisierte Abwicklung von Operationen wird ebenfalls durch Angabe von mehreren Zeitbedingungen möglich.

Wesentlich für die Funktionsfähigkeit einer Echtzeitanwendung ist die Vorhersehbarkeit des Zeitverhaltens in den definierten zeitkritischen Abschnitten [Her91]. Die Vorhersehbarkeit wird durch Echtzeitplanung erreicht. Als Parameter gehen in diese Planung die von den Anwendungen vorgegebenen Zeitanforderungen ein.

Die Schnelligkeit eines Systems bildet im Zusammenhang mit Echtzeitverarbeitung einen Parameter, der der Berechnung einer minimalen Bearbeitungsdauer für eine Operation zugrunde liegt. Daraus läßt sich nur erkennen, ob die Ausführung einer Operation, deren Start einer unteren und deren Ende einer oberen Schranke unterliegen, zwangsläufig eine der beiden Schranken verletzt. Der minimale Zeitbedarf wird neben diesem Parameter von der Planungsstrategie beeinflusst, der die in einem Echtzeitsystem konkurrierenden Operationen unterworfen sind. Mit der Fristenplanung und der Spielraumplanung werden zwei grundlegende in [Her91] erläuterte Verfahren zur Echtzeitplanung vorgestellt.

Aus der Fristenplanung ergibt sich eine Ausführungsreihenfolge für konkurrierende zeitkritische Operationen nach ihren Zielzeitpunkten. Unter Berücksichtigung der minimalen Ausführungsdauer läßt sich der spätest mögliche

Startzeitpunkt einer zeitkritischen Operation ermitteln. Wenn die, Ordnung nach Zielzeitpunkten zu einer Verletzung des spätest, möglichsten Anfangszeitpunktes für eine zeitkritische Operation führt, ist, die rechtzeitige Ausführung nach dieser Strategie nicht möglich.

Unter Berücksichtigung des Spielraumes, der sich aus der minimalen Ausführungszeit einer Operation und dem vorgegebenen Zeitrahmen ergibt, läßt sich eine flexiblere Planung durchführen. Für Operationen, die nur einer unteren Schranke unterliegen, ergibt sich ein unbegrenzter Spielraum. Wenn in die Planung der Spielraum zeitkritischer Operationen eingeht, lassen sich Operationen mit einem Puffer Zeitrahmen gegenüber Operationen mit einem großzügig dimensionierten Zeitrahmen bevorzugt behandeln.

Neben der zur Unterstützung von Echtzeitanforderungen gewählten Planungsstrategie spielt die Verfügbarkeit der Komponenten des Echtzeitsystems eine entscheidende Rolle. Der Ausfall einer Komponente darf nicht zu unvorhersehbaren Folgen für die Anwendung führen. Für wesentliche Komponenten muß daher durch Fehlertoleranzmaßnahmen, wie z.B. Redundanz, die Ausfallsicherheit gewährleistet werden.

Abhängig von der Qualität der Zeitanforderungen einer Echtzeitanwendung ergeben sich mehr oder weniger schwerwiegende Konsequenzen aus ihrer Verletzung. Das Echtzeitsystem muß daher die vorgegebenen Zeitgrenzen kontrollieren und bei Verletzung einer Zeitbedingung die situationsbezogene Bearbeitung durch die Anwendung gestatten. In der Echtzeitplanung sollte neben den Zeitanforderungen auch eine Priorität berücksichtigt werden, in der sich die Härte der Zeitanforderungen und damit die aus ihrer Verletzung resultierenden Konsequenzen niederschlagen.

Die Abwicklung verteilter Echtzeitanwendungen setzt definiertes Zeitverhalten für die beteiligten Rechner und für den Informationsaustausch voraus. Der Nachrichtenfluß muß demnach durch minimalen und maximalen Zeitbedarf eingrenzbar sein. Eine wahrscheinlichkeitstheoretische Beschreibung des Übertragungsverhaltens erlaubt keine definitive Aussage über das Zeitverhalten und ist im Zusammenhang mit Echtzeitanforderungen nicht ausreichend. Die Interpretation der Übertragungsgrenzwerte muß für die in die Kommunikation einbezogenen Rechner möglich sein. Bereits von der Übertragung einer Nachricht sind zwei Rechner eines verteilten Systems betroffen, die die Be-

wertung der Übertragungsdauer nach unterschiedlichen Uhren durchführen können. Durch den Einsatz von Synchronisationskonzepten für verteilte Uhren läßt sich in jedem Rechner eine Interpretation von Zeitaussagen anderer Rechner erreichen.

2.2.2 Schwache Synchronisation von Uhren

Die schwache Synchronisation von Uhren wird durch Konstruktion logischer Uhren in Abhängigkeit von physikalischen Uhren und dem Nachrichtentransfer erreicht. Schwache Synchronisation basiert auf einer vorgegebenen Kommunikation, die im allgemeinen nicht durch regelmäßige zeitliche Abläufe bestimmt ist und keine zusätzlichen Synchronisationsnachrichten enthält.

Für die logischen Uhren wird der Gleichlauf mit einer maximalen Abweichung gefordert. Die maximale Abweichung logischer Uhren bildet eine Funktion über der Zeit. Die Differenz der Zeitwerte zweier beliebiger synchronisierter logischer Uhren darf zu keinem Zeitpunkt den Funktionswert überschreiten. Zur Nachbildung der Monotonie der Zeit wird die Forderung erhoben, daß eine logische Uhr nie zurückgesetzt werden darf. Daraus leitet sich ein unbegrenzter Wertebereich für logische Uhren ab.

Die physikalischen Uhren verschiedener Prozessoren sind voneinander unabhängig und können auseinanderdriften. Für korrekte physikalische Uhren wird vorausgesetzt, daß die Abweichung von der realen Zeit durch eine a priori bekannte Funktion beschreibbar ist.

Zur Synchronisation der logischen Uhren wird die Kommunikation der Prozesse benutzt. Den Sende- und Empfangsereignissen werden logische Zeitwerte zugeordnet, die mit der Reihenfolge der Ereignisse verträglich sind. Die schwache Synchronisation der Uhren basiert auf der Annahme, daß jeder Prozeß für jede Verbindung zu seinen Kommunikationspartnern eine obere und eine untere Schranke für die Übertragungszeit kennt.

In [Dol86] wird nachgewiesen, daß für die Synchronisation von Uhren durch das Übertragungsverhalten des Netzwerks eine untere Schranke vorgegeben ist. Die Kommunikation der Prozesse wird dabei nur zur Initialisierung der Uhren eingesetzt. Der Nachweis wird für idealisierte Bedingungen geführt,

die jedem Prozeß beim Start die Benachrichtigung aller bereits aktiven Nachbarprozesse auferlegen. Durch Weiterleitung dieser Nachrichten werden sukzessive alle Prozesse über den Initialzeitpunkt des neuen Prozesses informiert. Die untere Grenze der Synchronisation hängt von der minimalen Latenz der Nachrichten ab.

Außerdem wird gezeigt, daß die Synchronisation auf ein physikalisches Einigungsverfahren zurückzuführen ist. Die Synchronisation läßt sich dann erreichen, wenn mindestens eine zwei Drittel Mehrheit aller Prozesse korrekt arbeitet, d.h. Zugriff auf eine physikalische Uhr besitzt und die Spezifikation erfüllt. Zudem müssen diese Prozesse durch das Netzwerk verbunden sein.

Die schwache Synchronisation bietet sich für die Hardware eines verteilten Systems an, wenn sie kein gemeinsames Protokoll zur starken Synchronisation unterstützen. Ein weiterer Vorteil der schwachen Synchronisation liegt in der Vermeidung expliziter Synchronisationsnachrichten.

2.2.3 Starke Uhrensynchronisation

Für die starke Synchronisation von Uhren in einer verteilten Umgebung kommen Protokolle zum Einsatz, die in regelmäßigen Abständen eine Nachführung der Rechneruhren ermöglichen. Dabei werden Synchronisationsnachrichten ausgetauscht. Zu unterscheiden sind Master-Slave-Mechanismen, die die Zeit eines Referenzrechners verteilen, und gleichberechtigte Einigungsverfahren, bei denen mehrere Rechner eine synchronisierte Zeit bestimmen. Der Aufwand zur Synchronisation von Uhren liegt zwischen einfachen Broadcast-Protokollen, die eine periodische Synchronisation von Uhren durch einen Zeit-Server erreichen, und komplexen Synchronisationsprotokollen, die die Netzlaufzeit berücksichtigen und die Synchronisationsabstände variieren.

Die Grenze der Synchronisation durch diese Protokolle wird ebenso wie im Fall der schwachen Synchronisation durch das Übertragungsverhalten des Netzwerkes beeinflusst. Durch regelmäßiges Versenden von Synchronisationsnachrichten werden Abschnitte definiert, in welchen die Aktualisierung der Uhren vorgenommen wird. Die maximale Abweichung der Uhren als Funk-

tion über der Zeit läßt sich auf jeden Abschnitt projizieren. Aus der Begrenzung der Länge eines Zeitabschnittes ergibt sich ein absoluter Wert der maximalen Abweichung.

Der Ausfall einer Synchronisationsnachricht bewirkt eine Vereinigung zweier Zeitabschnitte und geht neben anderen z.B. durch das Netzwerk verursachten Schwankungen in die Fehlerabschätzung eines Synchronisationsverfahrens mit ein. Starke Synchronisationsmechanismen sind ebenso wie schwache Synchronisationsverfahren von Rechnerausfällen und der Gefahr der Netzwerkpartitionierung betroffen.

Der Einsatz synchronisierter Rechneruhren erlaubt, eine Kontrolle der bearbeiteten Prozesse bezüglich der synchronisierten Zeit. Die Definition des Zeitverhaltens eines Prozesses relativ zu dieser Zeit wirft neue Probleme auf. Für eine Berechnung, die in einem Prozessor durchgeführt wird, ist die Angabe der notwendigen Taktzyklen möglich. Unter Bezug auf eine Rechneruhr, die sich an den Taktzyklen orientiert, ergibt sich eine direkte Proportionalität zwischen der minimalen Bearbeitungszeit und den dafür benötigten Zyklen. Die Bearbeitungszyklen sind aber im allgemeinen nicht proportional zu dem minimalen Zeitbedarf bezüglich der synchronisierten Uhr.

An der Universität Erlangen/Nürnberg wurden am Lehrstuhl für Betriebssysteme (IMMD IV) Untersuchungen bezüglich der Synchronisation von Rechneruhren unter Einsatz des Network Time Protocol (NTP) durchgeführt, [Kar93]. Als Ergebnis dieser Studien wurde für Rechner in einem lokalen Netz eine Synchronisation von weniger als 2 Millisekunden angegeben. Für die Synchronisation von Rechnern, die ans deutsche Wissenschaftsnetz (WIN) angebunden sind, wurde ein Wert von unter 80 Millisekunden ermittelt. Der Unterschied der beiden Werte ist auf die unterschiedliche Laufzeit in LANs gegenüber WANs zu sehen.

Für die sichere Abwicklung einer verteilten Echtzeitanwendung müssen die angegebenen Erfahrungswerte durch Worst-Case-Werte ersetzt werden, um definiertes Zeitverhalten bezüglich der synchronisierten Uhr zu garantieren.

2.3 Transaktionsverarbeitung unter Echtzeitbedingungen in einem verteilten System

Eine Anwendung, deren korrekte Abwicklung an die Unterstützung von Transaktionen und an die Einhaltung von zeitlichen Bedingungen geknüpft ist, stellt hohe Ansprüche an die Systemunterstützung. Neben der Gewährleistung der für Transaktionsverarbeitung grundlegenden ACID-Eigenschaften ist die rechtzeitige Ausführung der durch die Transaktion vorgegebenen Operationen gefordert.

In der Transaktionsverarbeitung wird die Fortsetzung der Verarbeitung nach dem Ausfall einer Komponente des Transaktionssystems durch Protokollierung von Transaktionszuständen auf stabilen Speicher erreicht. Eine Verzögerung, die sich aus dem Austausch oder der Reparatur ergeben kann, hat keinen Einfluß auf die korrekte Erfüllung der Transaktionen. Der Ausfall einer Komponente eines Echtzeitsystems darf dagegen nicht zur unvorhersehbaren Verzögerung der Bearbeitung führen. Reparaturzeiten dürfen keine Verletzung der zeitlich korrekten Funktionserfüllung nach sich ziehen. Die Funktion von Komponenten, deren Ausfall zu undefiniertem Systemverhalten führt, wird deshalb durch Redundanz garantiert.

Ein System, welches Transaktionen mit Echtzeitanforderungen unterstützt, muß den Spezifikationen beider Bereiche genügen. Es muß die Verletzung einer Zeitbedingung erkennen und die Möglichkeit der situationsbezogenen Reaktion durch die Anwendung erlauben, ohne die ACID-Eigenschaften zu gefährden.

Die Unterstützung verteilter Anwendungen erfordert geeignete Kommunikationsmechanismen, die zum einen die ACID-Eigenschaften, zum anderen die Zeitbedingungen berücksichtigen. Die Commit-Protokolle, die den Erhalt der ACID-Eigenschaften ermöglichen, wurden in Abschnitt 2.1.3 vorgestellt. Sie beinhalten keine Unterstützung in Bezug auf die Überwachung von Zeitbedingungen einer verteilten Anwendung. In einer verteilten Umgebung, die Transaktionsverarbeitung unter Echtzeitbedingungen unterstützt, ist daher während der Abwicklung des Commit-Protokolls die rechnerübergreifende Behandlung von Zeitbedingungen gefordert.

Die Echtzeitverarbeitung einer Transaktion in einem verteilten System setzt

neben dem definierten Zeitverhalten für alle mitwirkenden Rechner auch zeitlich begrenzbares Übertragungsverhalten voraus. Die Nachrichtenlaufzeit für einen quittierten Nachrichtenfluß zwischen zwei Rechnern eines verteilten Systems läßt sich auf dem initiierenden Rechner ermitteln, wenn die Beantwortung unmittelbar nach Empfang der Nachricht erfolgt. Die Ergebnisse einer Meßreihe von quittierten Nachrichten sind allerdings statistischer Natur. Aus der kürzesten und der längsten Übertragungszeit lassen sich Schätzwerte für die geforderte minimale und maximale Übertragungszeit berechnen. Die Güte dieser Schätzwerte hängt neben der Anzahl der berücksichtigten Übertragungen von den in die Berechnung einfließenden Parametern ab, die sich aus der Verkehrsstatistik einer Verbindung ergeben können.

Aus der Beschreibung des 2-Phasen-Commit-Protokolls geht hervor, daß die Aufforderung zur lokalen Entscheidungsfindung sowie die Mitteilung der globalen Entscheidung bestätigt werden. Im 3-Phasen-Commit-Protokoll kommt die ebenfalls bestätigte Übermittlung der Durchsetzungsaufforderung hinzu. Die Quittungen dieser Nachrichten werden jedoch nicht unmittelbar nach ihrem Empfang abgesendet. Sie werden durch lokale Verarbeitung, in einer hierarchischen Kommunikationsstruktur zusätzlich durch die Weiterleitung der Information nach unten und das Warten auf die Bestätigung, verzögert. Durch Einführung einer zusätzlichen unmittelbaren Antwort auf eine der aufgezählten Nachrichten des Commit-Protokolls wäre die Ermittlung von Erfahrungswerten für die Laufzeit von Nachricht einschließlich ihrer Bestätigung in den Rechnern des verteilten Transaktionssystems möglich.

Die Übertragungsrichtung der zusätzlich eingeführten Quittung stimmt mit der erwarteten Bestätigung überein, sodaß sich aus dieser Ergänzung keine zusätzlichen Blockierungen ergeben können. Das Ausbleiben der neu eingeführten Quittung wird beim Eintreffen der Bestätigung erkannt. Die Länge der Nachrichten im Commit-Protokoll ist als unabhängig von der jeweiligen Transaktion zu betrachten, wenn sie keine anwendungsspezifischen Daten beinhaltet. Für diesen Fall lassen sich die ermittelten Meßwerte als unabhängig von den zugrundeliegenden Transaktionen bezeichnen.

3 Konzepte zur Beschreibung temporaler Beziehungen

Für die Beschreibung des Einflusses der Kommunikation auf das Zeitverhalten eines verteilten Systems werden zwei ereignisorientierte Ansätze herangezogen. Beide erlauben die Darstellung von

- beliebigen prozeßinternen Beziehungen zwischen Ereignissen,
- Wechselwirkungen zwischen Ereignissen verschiedener Prozesse.

Beide Modelle basieren auf der Annahme, daß jeder Prozeß durch eine Menge von Ereignissen und Relationen über der Ereignismenge charakterisierbar ist.

3.1 Kausale Zeitstrukturen

Für die Beschreibung des ISO-TP-Dienstes in Kapitel 5 wird auf die in (Obe90] eingeführte Beschreibung kausaler Zeitstrukturen und kausaler Zeitintervalle zurückgegriffen. Grundlegende Begriffe dieses Ansatzes und der graphischen Aufbereitung werden nachfolgend vorgestellt.

Bezugssystem : Eine kontinuierliche Zeitachse bildet das Bezugssystem für die Einordnung der Ereignisse.

Bezugszeitpunkt : Jedem Ereignis eines Prozesses wird ein Bezugszeitpunkt zugeordnet, der einem Zeitpunkt auf einer kontinuierlichen Zeitachse entspricht.

Zeitobjekt : Ein Zeitobjekt beschreibt einen beliebigen zeitabhängigen Sachverhalt, der durch ein Anfangsereignis und ein Endeereignis begrenzt ist. Um die Modellierung von Zeitobjektfolgen zu gewährleisten, wurde ein Zeitobjekt als halboffenes Intervall aufgefaßt, welches den Bezugszeitpunkt des Anfangsereignisses beinhaltet, den des Endeereignisses aber ausklammert.

Kausale Zeitstruktur : Eine kausale Zeitstruktur basiert auf kausaler Abhängigkeit der Bezugszeitpunkte des Ereignisraumes. Ein Beispiel für die kausale Abhängigkeit zweier Ereignisse bildet die Beziehung zwischen dem Endeereignis eines Zeitobjektes und seinem Anfangsereignis. Das Endeereignis kann nur nach dem Anfangsereignis eintreten. Durch Zuordnung der kausalen Bezugszeitpunkte zu den verschiedenen Fliegssystemen wird die Wechselwirkung zwischen räumlich verteilten Prozessen erfasst.

kausale Zeit-Relationen : In kausalen Zeitstrukturen sind folgende Relationen definiert:

- Vorgängerrelation
- Nachfolgerrelation
- Hierarchien der Ereignisabfolge

Kausale Zeitintervalle : In einer kausalen Zeitstruktur bildet die Menge aller Paare von Ereignissen, die durch eine Folge kausaler Zeitobjekte in Beziehung stehen, die Menge der kausalen Zeitintervalle.

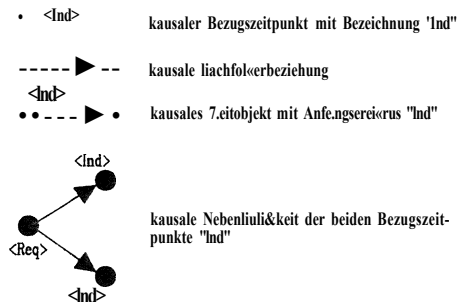


Abbildung 2: graphische Symbole zur Darstellung von kausalen Zeitstrukturen

Aus den Ereignisbeziehungen lassen sich kausale Zusammenhänge zwischen Zeitobjekten ableiten. Mit Hilfe der Definition der Verfeinerung von Zeitobjekten wird der Aufbau einer hierarchischen Ablaufbeschreibung ermöglicht.

Kausale Zeitstrukturen definieren eine partielle Ordnung auf dem Ereignisraum. Jeder in einer Zeitstruktur festgelegte Zeitpunkt tritt genau einmal ein. Alternative Abläufe sind innerhalb einer Zeitstruktur nicht darstellbar. Grenzen einer kausalen Zeitstruktur ergeben sich aus der Darstellung alternativer und zyklischer Abläufe. In [Obe90] wurde durch die Einführung kausaler Zeitstrukturen in Petrinetze die Beschreibung sehr einfacher ermöglicht.

Für die graphische Veranschaulichung von Zeitstrukturen werden in Abbildung 2 Symbole vorgestellt, die Ereignisse als Punkte und Relationen als Pfeile zwischen zwei Ereignispunkten beschreiben. Durch graphische Nachbildung von Zeitstrukturen entsteht ein gerichteter Graph.

Für die Darstellung kausaler Beziehungen zur Veranschaulichung der TP-Dienste wird auf den in diesem Kapitel vorgestellten Ansatz zurückgegriffen. Auf den Einsatz von Petrinetzen zur Beschreibung komplexer Wechselwirkungen mehrerer Zeitstrukturen wird aus Gründen der Übersichtlichkeit verzichtet. Das nachfolgend vorgestellte Spezifikationsmodell erscheint für die Beschreibung dieser Zusammenhänge angemessen.

3.2 Aktivitäten sequentieller Prozesse

Das Zeitverhalten eines Kommunikationsmanagers zur Realisierung der TP-Dienste wird mit dem in [And89] vorgestellten Ansatz zur Beschreibung eines sequentiellen Prozeßablaufs durch Aktivitäten dargestellt. Dieser Ansatz basiert auf dem Begriff der virtuellen Zeit, die als aufzählbare wohlgeordnete Menge von Zeitpunkten eingeführt wird. Der zeitliche Abstand zwischen zwei Zeitpunkten wird als Zeitdauer interpretiert. Zur Beschreibung der Beziehung zwischen den Ereignissen eines Prozesses wird der Begriff der Aktivität definiert.

"Eine Aktivität ist ein Paar $A=(e,d)$, wobei e ein Element der Ereignismenge eines Prozesses ist und d den Abstand zum nächsten Zeitpunkt angibt, zu dem wieder ein Ereignis eintreten kann."

Eine Aktivität wird durch ihr Startereignis und eine Zeitdauer modelliert. Für die Beschreibung der Zusammenhänge zwischen Aktivitäten wird jedem

Startereignis ein Zeitpunkt einer logischen Uhr zugeordnet. Die Konstruktion einer logischen Uhr orientiert sich an einem in [Lam78] vorgestellten Verfahren. Darin wird auf der Grundlage einer totalen Ordnung der Prozesse eine totale Ordnung aller Ereignisse des Ereignisraumes angegeben, die mit den kausalen und den zeitlichen Wechselwirkungen im Ereignisraum verträglich ist. Jeder Prozeß p_i ordnet jedem Ereignis e_i eine Ereignismenge eine Zahl zu und realisiert damit eine lokale Uhr C_i , die die Reihenfolge der Ereignisse erfaßt. Diese lokalen Uhren sind unabhängig von der physikalischen Zeit und müssen den beiden nachfolgenden Bedingungen genügen:

- " Für je zwei Ereignisse e_1, e_2 aus der Ereignismenge eines Prozesses P_i gilt: wenn a vor b eintritt, dann ist $C_i(a) < C_i(b)$.
- " Für ein Sendeereignis a aus der Ereignismenge des Prozesses P_i und das entsprechende Empfangsereignis b aus der Ereignismenge des Prozesses P_j gilt: $C_i(a) < C_j(b)$

Für zwei kausal nebeneinanderliegende, zeitlich unvergleichbare Ereignisse a, b wird für den Fall der Gleichheit der lokalen Uhren, $C_i(a) = C_j(b)$ die Ordnung der Ereignisse a und b aus der totalen Ordnung der Prozesse abgeleitet. Diese logische Totalordnung ermöglicht eine Ordnung zweier Ereignisse, die mit der kausalen Beziehung der Ereignisse verträglich ist. Aus den beiden Zeiten lassen sich aber keine Rückschlüsse auf die kausale Beziehung der Ereignisse ziehen.

[And87] verzichtet auf die totale Ordnung der Prozesse und konstruiert eine logische Uhr als karthesisches Produkt lokaler Uhren. Die Komponenten des Uhrvektors werden bei der Kommunikation von Prozessen so berechnet, daß

„Zwei kausal nebeneinanderliegende, zeitlich unvergleichbare Ereignisse a aus der Ereignismenge des Prozesses P_i und b aus der Ereignismenge des Prozesses P_j sind durch folgende Eigenschaften charakterisiert:

- a und alle dem Ereignis a nachgeordneten Ereignisse des Prozesses P_i sind nicht Sendeereignisse zu einem Empfangsereignis, welches mit b oder einem b vorangehenden Ereignis aus der Ereignismenge des Prozesses P_j identisch ist.
- b und alle dem Ereignis b nachgeordneten Ereignisse des Prozesses P_j sind nicht Sendeereignisse zu einem Empfangsereignis, welches mit a oder einem a vorangehenden Ereignis aus der Ereignismenge des Prozesses P_i identisch ist.

bei asynchroner Kommunikation die Uhrkomponente für das Empfangsereignis einen größeren Wert erhält als die Komponente für das Sendeereignis. In [SM91] wird gezeigt, daß kausale Beziehungen zwischen Ereignissen durch Vektoruhren charakterisierbar sind. Jedem Ereignis aus der Ereignismenge ist ein eindeutiger Vektor zugeordnet. Aus den Vektorzeiten zweier Ereignisse läßt sich die kausale Beziehung erkennen. Mit Hilfe der Vektorzeit ist die kausale Vorgeschichte eines Ereignisses als Matrix darstellbar.

4 Das ISO-TP-JVfodeH

11/r f.0 TP-z1.tldmd lv;chricht P;nen Komrnnnlk?ttionsdienst zur Erg.tel-
luna und W.-rvrdtong finrr kcmplPXEn Kommnnkali-H;c;struktur für verteilte
Anwendungen. Innerhalb einer Kommunikationsstruktur bietet TP Unter-
stützung für ;,orte,lt,r l rai,s,aktio*,sw-rarbeitung an. Eirff **Transaktion** Wird
in [TPM90] wie folgt definiert:

...A 'r;Jsi\ctiun i; " sei. of rdate<l operations characterized by
four properties: 2Jonlicity, consistency, isolation, durability'.

Uies< D(filitorl. entspricht dñll in Kapitel 1 anp;gebfñen Transaktionsbe-
f;ñif und verw!ist auf dit dort aufgefñhrfñ AC:ID-Rul;S.

De Trennilf!; von kormntfñkatiunsuntPrst.ützencln und trcInsaktionsbezoge-
r(n) nsÜ,ñll (dfñwt den Standard auch für vfrteiltc AInwendungen, die keine
r-rall;aktio!lsv,rarl<-itung durchführen. Die beiden folgenden Kapitel stellen
di• r-itunidl{11 StruktHrc11 und diñ zeitliche Glied<-rllug von Abläufen inner-
i,,ñl, di•s,r Strnktun-!1 vor,

4.1 Räumliche Strukturierung einer TP-Anwendung

l);. 1<onnnnullikationslnterstütZ,1111g von ISO-TP beruht auf der Integra-
tion einer Anwendung ia eine OSI-Umgebung mit vollständigem ISO-
Protokollturm. Das [SO-TP-Modell abstrahiert von den 7 Schichten des
ISO-Refrcrcnzmodelis und definiert auf dieser Grundlage die Kommunika-
tionsstruktur und die Tumsa,ktionsbereiche einer TP-Anwendung.

4.1.1 Das TP-Modell gegenüber dem Modell konkurrierender Subtransaktionen

Die Integration einer TP-Anwendung ill ein verteiltes Transaktionssystem
setzt eine Prozeßstruktur voraus. Das TP-Modell verbirgt die reale verteilte
Umgebung unter dem Begriff des TP-Service-Providers. Abbildung 3 stellt
das TP-Modell einer verteilten TP-Anwendung graphisch dar. Die grund-
legenden Begriffe dieses Modells und ihre Zusammenhänge werden in der
nachfolgenden Aufzählung erläutert.

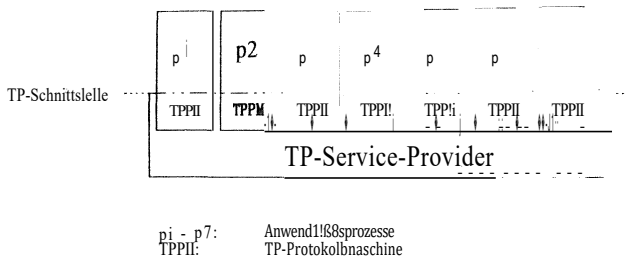


Abbildung 3: TP-Schnittstelle

Anwendungsprozesse: Es wird eine Gliederung der Anwendung in Anwendungsprozesse vorausgesetzt, die im allgemeinen von verschiedenen Subsystemen eines verteilten Transaktionssystems unterstützt werden und daher nicht über gemeinsamen Speicher verfügen.

TP-Service-Provider: Die Prozesse werden durch den subsystemübergreifenden TP-Service-Provider verknüpft, der die Kommunikationsgrundlage für die Anwendungsprozesse in Form von Dienstleistungen bereitstellt.

TP-Schnittstelle: Der TP-Service-Provider verbirgt die zur Realisierung der Dienste notwendigen Protokolle und bietet pro Anwendungsprozeß eine lokale Ausprägung der TP-Schnittstelle an.

TP-Dienste: Die vom TP-Service-Provider bereitgestellten TP-Dienste regeln den Informationsfluß zwischen den Prozessen der Anwendung durch Nachrichtentransfer.

TP-Dienst-Primitive: Für jeden TP-Dienst ist eine Folge von Dienstprimitiven definiert, die als Schnittstellensignale zwischen einem Prozeß der TP-Anwendung und dem TP-Service-Provider übergeben werden.

TP-Protokollmaschine (TPPM): Jede TP-Protokollmaschine übernimmt für genau einen Anwendungsprozeß die Abwicklung der Kommunikationsprotokolle. Der TP-Service-Provider enthält die TP-Protokollmaschinen aller Prozesse einer TP-Anwendung.

Der TP-Standard nimmt Bezug auf *eine verteilte Anwendung* und definiert die Kommunikationsgrundlage für die in diese Anwendung integrierten Prozesse. *Die konkurrierende Bearbeitung einzelner Anwendungsprozesse innerhalb eines Subsystems wird im TP-Standard nicht berücksichtigt.* Die Einordnung des TP-Modells in das Kommunikationsmodell konkurrierender Subtransaktionen³ wird durch die beiden folgenden Begriffsabbildungen wiedergegeben:

Kommunikationsmodell konkurrierender Subtransaktionen → TP-Modell

- Die Kommunikationsmanager eines verteilten Transaktionssystems realisieren alle TP-Protokollmaschinen einer verteilten TP-Anwendung.
- Das Kommunikationssystem eines verteilten Transaktionssystems übernimmt für TP-Anwendungen alle Aktionen der TP-Service-Provider mit Ausnahme der Funktionalität der TP-Protokollmaschinen. Es realisiert damit die Ebenen 1-6 des ISO-Referenzmodells für mehrere Anwendungen.
- Eine Subtransaktion implementiert alle Aktionen eines TP-Anwendungsprozesses mit Ausnahme der Aktionen der TP-Protokollmaschine.
- Ein Subsystem unterstützt mehrere TP-Anwendungsprozesse.

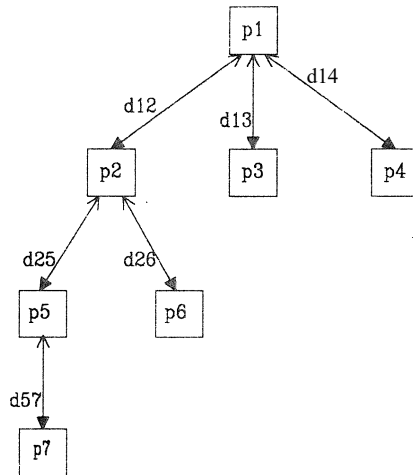
TP-Modell → Kommunikationsmodell konkurrierender Subtransaktionen

- Ein TP-Anwendungsprozeß ist genau einem Subsystem zugeordnet.
- Eine TP-Protokollmaschine ist genau einem Kommunikationsmanager zugeordnet.
- Der neben der TP-Protokollmaschine verbleibende Anteil eines TP-Anwendungsprozesses wird mit dem Begriff der Subtransaktion identifiziert.
- Der TP-Service-Provider mit Ausnahme der TP-Protokollmaschinen wird vom Kommunikationssystem getragen.

³Siehe Abbildung 1 auf Seite 19.

4.1.2 Der TP-Dialogbaum einer TP-Anwendung

Die mit TP erstellbare Kommunikationsstruktur fwschränkt sich auf die Verknüpfung von Anwendungsprozessen w eirwr Ballmstrukt.11r. TP-Di<nst.< sind nur für verbindungsorientiertf Kommunikatioll von Anw<nd1111gsproz7, sen definiert.



d12, d13, d14, d25, d26, d57 : Dialoge
 p1, p2, p3, p4, p5, p6, p7 : Anwendungsprozesse

Abbildung 4: TP-Dialogbaum

TP-Dialog: Eine Verbindung zwischen Anwendungsprozessen wird als TP-Dialog bezeichnet und von den beiden Partnerprozessen entsprechend den spezifischen Anforderungen konfiguriert. Die TP-Unterstützung wird durch Aushandeln der in [TPS90] definierten Funktionsklassen für jeden Dialog einzeln festgelegt. Für spezifische Kommunikationsanforderungen, die über die von TP erfaßte Funktionalität hinausgehen, läßt sich jeder Dialog durch Integration anderer ISO-Ebene-7-Dienste erweitern ([ALS89], [TPM90]).

Abbildung 4 zeigt als Beispiel einen Dialogbaum, in welchem die mit p1 bis p7 beschrifteten Rechtecke die Anwendungsprozesse symbolisieren. Die mit d12, d13, d14, d25, d26 und d57 gekennzeichneten Verbindungslinien zwischen je zwei Rechtecken stellen die Dialoge dar. Die beiden Kommunikationsrichtungen eines Dialoges sind durch Pfeile an den Enden der Verbindungslinien wiedergegeben. Der ausgefüllte Pfeil deutet die Dialogaufbaurichtung an.

Der Aufbau dieser Kommunikationsstruktur beginnt in einem durch die Anwendung festgelegten Initialprozeß, der den Wurzelknoten des Baumes und damit den Initialbaum bildet (in Abbildung 4 der Prozeß p1). Ausgehend von diesem Wurzelknoten kann jeder in den Dialogbaum integrierte Prozeß durch Absetzen von Dialogaufbauaufforderungen weitere bisher isolierte Prozesse in die Kommunikationsstruktur einbinden. Jeder Dialogaufbau erweitert den Dialogbaum um einen neuen Prozeßknoten und eine neue Dialogkante, die die Kommunikationsverbindung zwischen dem anfordernden und dem neuen Knoten bildet.

Graphentheoretisch wird ein Dialogbaum als zusammenhängender, gerichteter azyklischer Graph mit einer Menge von Anwendungsprozessen als Knoten, einer Menge von Dialogverbindungen als Kanten, einem ausgezeichneten Wurzelprozeß, einer Dialoginitiator-Funktion und einer Dialogakzeptor-Funktion beschrieben:

$$\text{Dbaum} = (A, D, r, i, a)$$

1 Dialogakzeptorfunktion

Dialoginitiatorfunktion

Wurzelprozeß

Dialoge

Anwendungsprozesse

Die Komponenten des Graphen besitzen folgende Eigenschaften:

- Die Menge der Anwendungsprozesse A ist endlich.
- Die Menge der Dialogverbindungen D ist endlich, es gilt:
 $1 \leq |A| \leq |D| \leq 1$.
- Die Funktionen i und a ordnen einem Dialog einen Dialoginitiatorprozeß bzw. einen Dialogakzeptorprozeß aus der Menge der Anwendungs-

prozesse zu: $i : D \succ A$ $a : D \succ A$.

Ein Dialog wird durch das Paar (Dialoginitiatorprozeß, Dialogakzeptorprozeß) eindeutig identifiziert.

- Der Dialogbaum-Wurzelprozeß ist in der Menge der Anwendungsprozesse enthalten, tritt aber nie als Dialogakzeptorprozeß auf:

$$r \in A, \forall p \in A: (p, r) \notin D.$$

- Jeder Anwendungsprozeß ist Initiator oder Akzeptor mindestens einer Dialogverbindung:

$$\forall p \in A \exists q \in A: (p, q) \in DV \vee (q, p) \in DV.$$

4.1.3 Transaktionsbereiche einer TP-Anwendung

TP unterstützt die nebenläufige Verarbeitung von verteilten Transaktionen für getrennte Transaktionsbereiche eines Dialogbaumes. Im Zusammenhang mit der Definition eines "Transaktionsbereiches" ist die Einführung weiterer Begriffe notwendig.

Transaktionsbereich: Jeder Transaktionsbereich umfaßt einen Teilbaum der Kommunikationsstruktur einer verteilten Anwendung. Aus der Abgrenzung von Transaktionsbereichen ergibt sich die Möglichkeit der parallelen Verarbeitung mehrerer Transaktionen in einem Dialogbaum.

Transaktionsprozeß: Jeder Anwendungsprozeß, der in einem Transaktionsbereich liegt, wird als Transaktionsprozeß bezeichnet.

Transaktionsverbindung: Jeder Dialog, der zwei Transaktionsprozesse eines Transaktionsbereiches verbindet, realisiert eine Transaktionsverbindung.

Rollen eines Transaktionsprozesses: Ein Transaktionsprozeß nimmt eine der drei Rollen

- Wurzelknoten, (w)
- Zwischenknoten, (zw)
- Blattknoten, (bl)

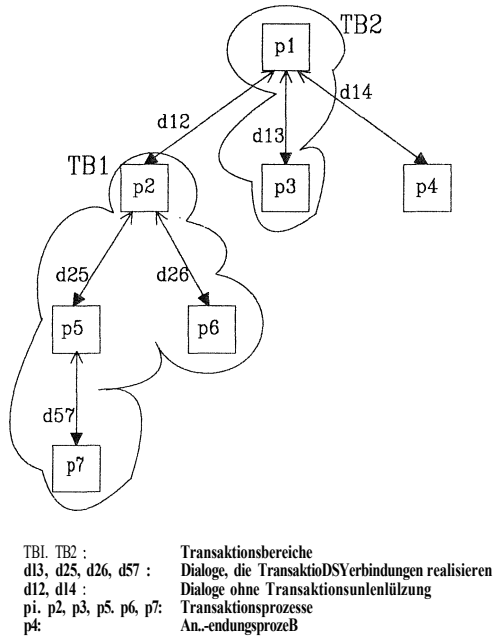


Abbildung 5: Transaktionsbereiche in einem Dialogbaum

ein, auf die im folgenden unter den eingeklammerten Abkürzungen Bezug genommen wird.

Abbildung 5 zeigt die Momentaufnahme eines Dialogbaumes einer verteilten Anwendung mit zwei Transaktionsbereichen TB1 und TB2, die zwei zeitlich nebenläufige, verteilte Transaktionen symbolisieren.

Graphentheoretisch wird ein Transaktionsbereich als zusammenhängender Teilgraph des Dialogbaumes beschrieben:

$$\text{Tbereich} = (P, V, w, i, a)$$

Transaktionsprozesse

Transaktionsverbindungen

Transaktionswurzelprozeß

Dialoginitiatorfunktion

hialogakzeptorfunktion

Für Transaktionsbereiche gelten folgende Eigenschaften:

- Die Menge der Transaktionsprozesse P eines Transaktionsbereiches bildet eine Teilmenge der Menge der Anwendungsprozesse A des Dialogbaumes. $P \subseteq A$;
- Die Menge der Transaktionsverbindungen V bildet eine Teilmenge der Dialoge D : $V \subseteq D$; $V(p, q) \in V : p \in P \wedge q \in P$.
- Zwei Transaktionsbereiche dürfen sich nicht überschneiden:
 $W, V' \subseteq D, V \cap V' = \emptyset$.
- Die Funktionen i und a werden aus der graphentheoretischen Definition des Dialogbaumes übernommen. Es gilt:
 $V(p, q) \in D \setminus V : i((p, q)) \notin P \vee a((p, q)) \notin P$.
- Die Wurzel eines Transaktionsbereiches liegt in der Menge der Transaktionsprozesse: $w \in P$.
 Die Wurzel eines Transaktionsbereiches kann nur als Akzeptorprozeß eines Dialoges auftreten, der keine Transaktionsverbindung realisiert:
 $\nexists p \in P : (p, w) \in V \wedge (\exists V' \subseteq D : (p, w) \in V')$.
- Alle Transaktionsprozesse eines Transaktionsbereiches sind an einer Transaktionsverbindung beteiligt:
 $\forall p \in P \exists q \in P : (p, q) \in V \vee (q, p) \in V$.

Für alle Transaktionsprozesse eines Transaktionsbereiches werden die direkten Vorgängerrelationen $prev$ und die direkten Nachfolgerrelationen $next$ definiert: $\forall p \in P$:

$$prev(p) = \{q \mid (q, p) \in V\} \quad (1)$$

$$next(p) = \{q \mid I(p, q) \in EV\} \quad (2)$$

Die Mächtigkeit der direkten Vorgängerrelation in einer Baumstruktur ist auf 1 begrenzt. Für die direkte Nachfolgerrelation eines Transaktionsprozesses läßt sich als obere Schranke nur die Mächtigkeit der Menge der Prozesse eines Transaktionsbereiches angeben.

$$I_{prev}(p) \leq 1; \quad I_{next}(p) \leq IV$$

Für einen Transaktionsprozeß p wird die Menge der (direkten und indirekten) Nachfolger und darauf aufbauend der Begriff des unterlagerten Teilbaumes definiert:

Menge der Nachfolger eines Transaktionsprozesses: Die Definition der Menge der direkten und indirekten Nachfolgerprozesse $next^*(p)$ eines Transaktionsprozesses p bildet die transitive Hülle über der Relation $next(p)$. Sie wird für die Einführung des Begriffes "unterlagerte Teilbaum" benötigt. Es gilt:

$$\begin{aligned} & \forall q \in next(p) \rightarrow q \in next^*(p); \\ & \forall q \in next^*(p), \forall r \in next(q) \rightarrow r \in next^*(p); \end{aligned} \quad (3)$$

Unterlagerte Teilbaum eines Transaktionsprozesses: Der einem Transaktionsprozeß unterlagerte Teilbaum bildet den zusammenhängenden Teilgraphen des Transaktionsbereiches, der den Transaktionsprozeß p alle seine (direkten und indirekten) Nachfolger $next^*(p)$ im Transaktionsbereich und alle zwischen diesen Transaktionsprozessen definierten Transaktionsverbindungen enthält.

4.2 Sequentielle Bearbeitungsabläufe in einem Transaktionsbereich

Im vorangehenden Kapitel wurde ein Transaktionsbereich als zusammenhängender Ausschnitt eines Dialogbaumes eingeführt. Neben der Möglichkeit, mehrere unabhängige Transaktionen in einem Dialogbaum nebenläufig auszuführen, sind in TP zwei Konzepte für die sequentielle Bearbeitung verteilter

Transaktionen vorgesehen. Sie erlauben die Wiederverwendung von Teilbereichen eines Transaktionsbaumes für die Bearbeitung unverketteter Transaktionen und Transaktionsketten. Als Grundelement der beiden Konzepte wird auf die Transaktion zurückgegriffen, deren Kommunikationsphasen zunächst erläutert werden.

4.2.1 Kommunikationsphasen einer verteilten Transaktion

Das Kommunikationsverhalten eines Transaktionsprozesses, der an einer erfolgreichen Transaktion mitwirkt, gliedert sich in eine Datentransferphase und zwei Phasen für die Abwicklung des Commit-Dienstes zum Abschluß der Transaktion. Die drei Phasen unterscheiden sich an der Kommunikationsschnittstelle durch Beschränkungen der angebotenen Dienste. Eine verteilte Transaktion, die zu einem **positiven Abschluß** kommt, durchläuft in jedem beteiligten Transaktionsprozeß alle drei Phasen. Die Kommunikationsphasen sind so definiert, daß sie sich nahtlos aneinanderfügen.

Die Datentransferphase eines Transaktionsprozesses :

In der Datentransferphase stehen einem Transaktionsprozeß neben den TP-Diensten alle weiteren beim Aufbau der einzelnen Dialoge ausgehandelten Kommunikationsdienste zur Verfügung. Damit ist der anwendungsspezifische Informationsaustausch gewährleistet. In dieser Phase kann der Transaktionsprozeß den Abbruch der verteilten Transaktion durch Aktivierung eines geeigneten TP-Dienstes erzwingen. Die Einleitung des Transaktionsabschlusses bleibt dem Wurzelprozeß des Transaktionsbereiches vorbehalten.

Die Commit-Phase-1 eines Transaktionsprozesses :

Diese Phase schließt in jedem Anwendungsprozeß unmittelbar an die Datentransferphase an. Sie charakterisiert den Zeitraum, in dem ein Transaktionsprozeß die anwendungsspezifische Kommunikation abgeschlossen und den positiven Abschluß der Transaktion befürwortet hat, die globale Entscheidung aber noch nicht kennt. Die Subtransaktion muß in dieser Phase noch mit dem Abbruch der Transaktion rechnen,

kann aber selbst keinen Transaktionsabbruch mehr einleiten. Diese Phase beinhaltet die Wartezeit auf die globale Entscheidung⁴.

Die Commit-Phase-2 eines Transaktionsprozesses :

Die Commit-Phase-2 schließt in jedem Transaktionsprozeß unmittelbar an die Commit-Phase-1 an. Mit dem Eintritt in diese Phase wird der positive Abschluß der Transaktion freigegeben. Ein Abbruch der Transaktion ist nicht mehr möglich. Mit dem Ende dieser Phase sind alle Kommunikationsaktionen des Transaktionsprozesses bezüglich der verteilten Transaktion abgeschlossen.

Der ISO-TP-Standard beruht auf der Abstraktion von lokalen Aktionen innerhalb eines Transaktionsprozesses. Nur das Kommunikationsverhalten der Transaktionsprozesse wird aufgegriffen. Aus den Kommunikationsphasen lassen sich nur eingeschränkte Rückschlüsse auf den Bearbeitungszeitraum der Transaktionsprozesse für eine verteilte Transaktion ableiten.

Der Anfang der Datentransferphase muß nicht den Start aller transaktionsbezogenen Aktionen widerspiegeln. Ein Wurzelprozeß kann z.B. die lokale Transaktionsverarbeitung lange vor der Eröffnung der Datentransferphase beginnen. Für alle anderen Transaktionsprozesse beginnt die Datentransferphase mit dem Zeitpunkt der Einbindung in den Transaktionsbaum. Das Ende der Commit-Phase-2 bezieht sich nur auf die Kommunikation eines Transaktionsprozesses. Innerhalb eines Transaktionsprozesses kann sich der Transaktionsabschluß hinauszögern. Nach der Protokollierung der globalen Entscheidung auf stabilen Speicher wird ihre Durchsetzung durch das lokale Subsystem garantiert. Eine Aussage über den Zeitpunkt der lokalen Vervollständigung ist darin nicht enthalten.

⁴Die mit Commit-Phase-1 bezeichnete *Kommunikationsphase* unterscheidet sich von dem im Bereich verteilter Datenbanken üblichen Verständnis der mit Commit-Phase-1 bezeichneten *Transaktionsphase*. Dort wird die Untergliederung in Transaktionsabschlußphasen zur Einordnung der Aktionen eines Subsystems vorgenommen. Die Wartezeit auf die globale Entscheidung trennt die beiden Transaktionsabschlußphasen, die sich unter Berücksichtigung des zeitlichen Ablaufs nicht aneinanderfügen.

4.2.2 Transaktionsketten

Bildet eine Transaktion ein Element aus einer Transaktionskette, so impliziert in jedem beteiligten Prozeß der Endzeitpunkt der Transaktion den Start der nachfolgenden Transaktion.

Der Transaktionsbereich einer verketteten Transaktion wird mit der ersten Transaktion der Kette aufgebaut. Jede weitere Transaktion erbt den Transaktionsbereich zum Zeitpunkt des Transaktionsabschlusses der vorangehenden Transaktion als initialen Transaktionsbereich. Eine Veränderung des Transaktionsbereiches kann nur im Zuge der Modifikation des Dialogbaumes erfolgen.

Während der Bearbeitungsphase läßt sich der Transaktionsbereich durch Eröffnen weiterer Dialoge zu neuen Anwendungsprozessen ausdehnen. Wenn der Transaktionsabschluß eingeleitet ist, kann der Abbau von Dialogen die Schrumpfung des Transaktionsbereiches herbeiführen. Abbildung 6 stellt als

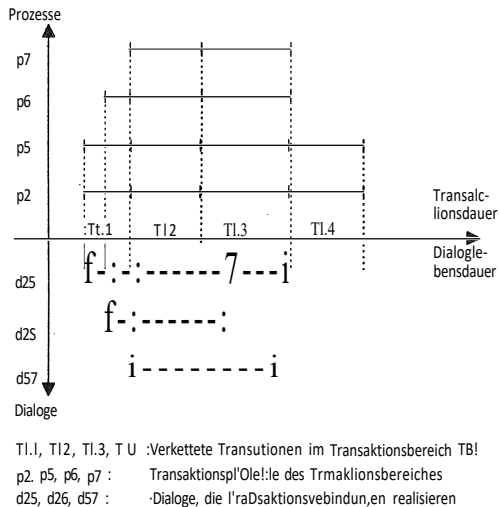


Abbildung 6: Transaktionskette

Beispiel ein Zeitdiagramm für eine Kette von Transaktionen T1.1 bis T1.4 dar. An der Transaktion T1.1 sind nur die beiden Prozesse p2 und p5 beteiligt. Die Transaktion T1.1 beginnt implizit mit dem Aufbau des Dialoges d25 als Transaktionsverbindung. Die Prozesse p6 und p7 werden durch den Aufbau der Dialoge d26 und d57 während der ersten bzw. am Anfang der zweiten Transaktion der Kette beteiligt. In der zweiten Transaktion der Kette hat der Transaktionsbereich die Form des in Abbildung 5 auf Seite 43 mit TBI bezeichneten Bereichs. Die Prozesse p6 und p7 werden durch Beenden der Dialoge d26 bzw. d57 aus der Kommunikationsstruktur der Anwendung und damit aus dem Transaktionsbereich TBI entfernt. Das Ende eines Dialoges tritt immer gleichzeitig mit dem Ende einer Transaktion ein.

Der Einsatz von Transaktionsketten ermöglicht die Unterstützung von Anwendungen, deren Aufgabe in der zusammenhängenden Abwicklung mehrerer Transaktionen unter durchgehender Beanspruchung von Betriebsmitteln besteht. Durch Verzicht auf die Freigabe der durchgängig benötigten Betriebsmittel am Ende einer Transaktion wird ggf. zu Lasten anderer Anwendungen der Fluß einer Transaktionskette unterstützt.

4.2.3 Unverkettete Transaktionen

Im Gegensatz zur Transaktionskette muß der Transaktionsbereich einer unverketteten Transaktion explizit aufgebaut werden. Der Transaktionsbereich kann durch Eröffnen neuer Dialoge oder durch Einbeziehen bestehender Dialoge angelegt werden. Zur Bearbeitung unverketteter Transaktionen stellt TP Dienste zur Verfügung, die ohne Modifikation der Kommunikationsstruktur den Aufbau eines Transaktionsbereiches erlauben. In der zweiten Transaktionsabschlußphase können Dialoge des Transaktionsbereiches beendet werden.

Mit dem Transaktionsabschluß wird der Transaktionsbereich einer unverketteten Transaktion aufgelöst, so daß jeder an der Transaktion beteiligte Prozeß in eine beliebige andere unverkettete Transaktion einbezogen werden kann.

Zwischen dem Ende einer Transaktion und dem Anfang der nächsten ergibt sich in jedem Prozeß ein Zeitraum unbestimmter Länge bis zur Bearbeitung

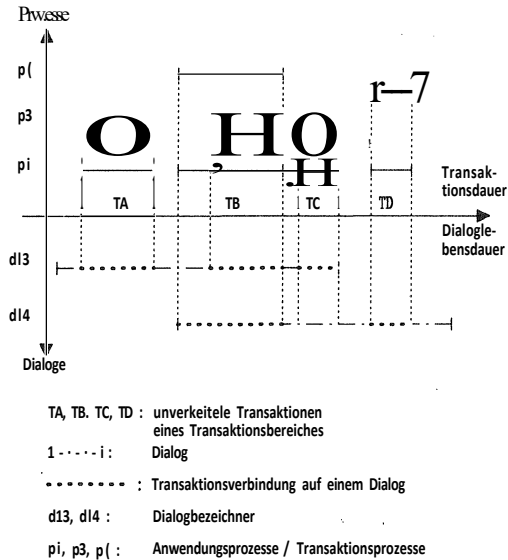


Abbildung 7: unverkettete Transaktionsfolge

einer anderen Transaktion. Als Beispiel wurde in Abbildung 7 ein Zeitverlauf für die Bearbeitung von 4 unverketteten Transaktionen dargestellt. Der bereits bestehende Dialog d13 wird als Transaktionsverbindung in die Transaktion TA aufgenommen, während der Dialog d14 beim Aufbau des Transaktionsbereiches der Transaktion TB erzeugt wurde. Der Prozeß p4 wurde damit an der Transaktion TB beteiligt. Während der Bearbeitung der Transaktion TC blieb der Dialog d14 erhalten, obwohl der Prozeß p4 in die Ausführung der Transaktion TC nicht einbezogen war. In der Transaktion TA hat der Transaktionsbereich die in Abbildung 5 mit TB2 bezeichnete Form. Der Abbau des Dialoges d13 erfolgt mit der Terminierung der Transaktion TC. Der Dialog d14 wurde erst nach dem Abschluß von TD abgebaut. Ein rechnergestütztes Reisebüro läßt sich als Anwendungsbeispiel für die Verarbeitung unverketteter Transaktionen anführen. Der Aufbau eines Transaktionsbereiches orientiert sich an den Wünschen eines Kunden. Entsprechende Reservierungen werden vom Reisebüro in qn unabhängigen Zielsystemen

beantragt. Nach der Bestätigung der Termine von allen in den Transaktionsbereich aufgenommenen Subtransaktionen kann das Reisebüro die Buchungen vornehmen und die Transaktion abschließen. Damit löst sich der Transaktionsbereich auf. Der Dialogbaum bleibt bestehen und kann so für den Aufbau eines Transaktionsbereiches für den nächsten Kunden genutzt werden.

5 Kausale Zeitstrukturen des TP-Basis-Standards

Aus dem Einsatz von TP-Diensten entstehen kausale Abhängigkeiten zwischen den Anwendungsprozessen eines Dialogbaumes. In diesem Abschnitt werden kausale Abhängigkeiten innerhalb eines Transaktionsbereiches dargestellt. Eine vollständige Behandlung aller aus dem Kommunikationsverhalten ableitbaren kausalen Abhängigkeiten ist nur für eine konkrete verteilte Transaktion möglich. Die Abstraktion vom anwendungsspezifischen Kommunikationsverhalten bewirkt eine Reduktion von der Menge aller möglichen TP-Dienste auf die Menge der zur Transaktionsabwicklung notwendigen TP-Aufrufe. Außerdem beschränken sich die hier erfaßten kausalen Zusammenhänge weitgehend auf erfolgreiche Transaktionen. Um die Ausbreitung der negativen Entscheidung im Fall eines Transaktionsabbruchs zu veranschaulichen, wird nur ein möglicher Ablauf als kausale Zeitstruktur dargestellt. Eine vollständige Angabe aller Variationen bringt keine zusätzlichen Erkenntnisse über den zeitlichen Verlauf des Rollback-Verfahrens, welches zur Verbreitung der negativen Entscheidung eingesetzt wird. Die aufgeführten Einschränkungen lassen sich wie folgt zusammenfassen:

Ziel dieses Kapitels ist die Darstellung der kausalen Beziehungen der TP-Dienste, die in den Transaktionsprozessen eines Transaktionsbereiches die Abwicklung einer erfolgreichen Transaktion begrenzen.

In die eingeschränkte Menge der in einer Transaktion notwendigen Kommunikationsdienste fallen die Dienste zum Aufbau eines Transaktionsbereiches, TP-Begin-Dialogue und TP-Begin-Transaction, die Dienste TP-Prepare und TP-Commit zum erfolgreichen Abschluß einer Transaktion und TP-Rollback zum Transaktionsabbruch. Unter Berücksichtigung der im Standard festgelegten Protokollabwicklung lassen sich die kausalen Zusammenhänge an der TP-Schnittstelle in verfeinerte kausale Zeitstrukturen zerlegen. Aus der Verknüpfung der verfeinerten Zeitstrukturen lassen sich kausale Zeitintervalle für die 3 Phasen eines Transaktionsprozesses ableiten. Für die drei nachfolgenden Abschnitte werden die in Kapitel 3 eingeführten Bezugssysteme

einer verteilten Transaktion so folgt festgelegt, daß alle Transaktionsprozesse verschiedenen Subsystemen zugeordnet sein können⁵:

Bezugssysteme einer verteilten Transaktion: Für eine verteilte Transaktion entspricht die Abfolge der Transaktionsprozesse eines Transaktionsbereiches der Abfolge der in einer kausalen Zeitstruktur zu berücksichtigenden Bezugssysteme.

5.1 Kausale Zeitstrukturen der TP-Schnittstelle eines Transaktionsbereiches

Die aus einem TP-Dienst resultierenden kausalen Beziehungen sind durch eine im TP-Standard festgelegte Ordnung über den zugehörigen Primitiven definiert.

Kausale Bezugszeitpunkte der TP-Schnittstelle: Die Dienstprimitive bilden die an der TP-Schnittstelle erkennbaren kausalen Bezugszeitpunkte.

Tabelle 1 listet diese Dienste und die zugehörigen Dienstprimitive auf. Die eingeklammerten Dienstprimitive sind optional. Die Dienstprimitive grenzen Zeitobjekte ein, die entweder in einer Subtransaktion oder im TP-Service-Provider bearbeitet werden. Die Zuordnung ergibt sich aus dem Endeereignis eines Zeitobjektes. Ein Primitiv vom Typ "Indication" bzw. "Confirmation" beendet ein Zeitobjekt des TP-Service-Providers. Er übergibt damit einer Subtransaktion Informationen des Dialogpartners am Ende des die Übertragung repräsentierenden Zeitobjektes oder zeigt ihr intern ermittelte Erkenntnisse an. Ein Primitiv vom Typ "Request" bzw. "Response" beendet ein Zeitobjekt einer Subtransaktion. Sie übergibt Information an den TP-Service-Provider und legt damit den Anfangszeitpunkt eines von ihm zu bearbeitenden Zeitobjektes fest.

⁵Für zwei Transaktionsprozesse, die vom selben Subsystem unterstützt werden, wäre die gleiche Uhr als Bezugssuhr und damit die Zuordnung zum selben Bezugssystem denkbar. Dieser Spezialfall wird im TP-Standard nicht behandelt und bleibt daher in der hier festgelegten Zuordnung von Bezugssystemen zu den Transaktionsprozessen unberücksichtigt.

TP-Dienst	Bezugszeitpunkte
TP-Begin-Dialogue	TP-Begin-Dialogue-Req TP-Begin-Dialogue-Ind [TP-Begin-Dialogue-Rsp TP-Begin-Dialogue-Cnf]
TP-Begin-Transaction	TP-Begin-Transaction-Req TP-Begin-Transaction-Ind
TP-Commit	TP-Commit-Req TP-Prepare-Req TP-Prepare-Ind TP-Ready-Ind TP-Commit-Ind TP-Done-Req TP-Commit-Complete-Ind
TP-Rollback	TP-Rollback-Req TP-Rollback-Ind TP-Done-Req TP-Rollback-Complete-Ind

Req: Request Ind: Indication

Rsp: Response Cnf: Confirmation

Tabelle 1: Bezugszeitpunkte der TP-Dienste an der TP-Schnittstelle

5.1.1 Aufbau eines Transaktionsbereiches für unverkettete Transaktionen

Der Aufbau eines Transaktionsbereiches erfolgt für imverkettete Transaktionen durch sukzessive Eröffnung einzelner Transaktionsverbindungen über bestehenden Dialogen oder gleichzeitig mit dem Dialogaufbau. Den Ausgangspunkt für den Aufbau des Transaktionsbereiches bildet die Transaktionswurzel. Dabei kommen die Dienste TP-Begin-Transaction für die Integration bestehender Dialoge und TP-Begin-Dialogue⁶ für die Generierung neuer Dialoge zum Einsatz. Die beiden Dienste sind auf die Datentrans-

⁶Mit dem Aufruf des Dienstes TP-Begin-Dialogue kann vom Dialoginitiatorprozeß eine Bestätigung angefordert werden. In diesem Fall beinhaltet die vollständige Abwicklung des Dienstes die in eckige Klammern eingefassten Primitive.

ferphase beschränkt. Die Primitive der beiden Dienste erfassen nur zwei Bezugssysteme des Transaktionsbereiches.

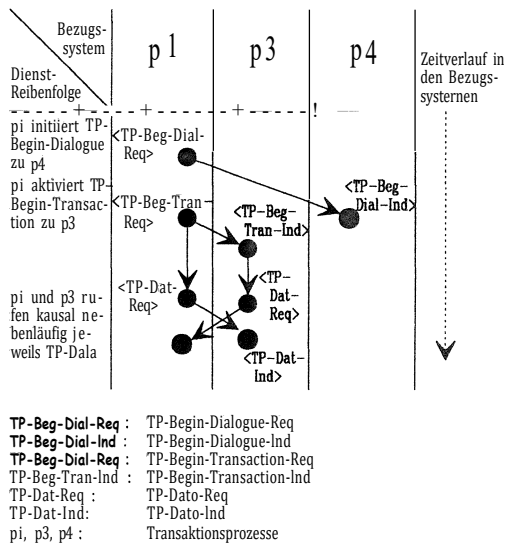


Abbildung 8: Aufbau eines Transaktionsbereiches für unverkettete Transaktionen

Aus den Regeln für den Aufbau eines Dialogbaumes ergibt sich, daß ein Anwendungsprozeß, der durch eine TP-Begin-Dialogue-Ind in einen Transaktionsbereich integriert wurde, an diesem kausalen Bezugszeitpunkt an keinem anderen Dialog beteiligt sein darf. Die unmittelbare Folge von TP-Begin-Dialogue-Ind und TP-Begin-Transaction-Req an der Schnittstelle zu einer Subtransaktion ist nicht zulässig. Der Dienst TP-Begin-Dialogue bindet einen ungebundenen Anwendungsprozeß in den Dialogbaum und abhängig von den im Dienst gesetzten Parametern in den Transaktionsbereich ein. In jedem Fall ist der neue Prozeß ein Blattknoten in der Kommunikationsstruktur und besitzt daher keine Dialoge, die er mit TP-Begin-Transaction zu einer Transaktionsverbindung erweitern könnte. Dem TP-Transaction-Req muß in diesem Fall ein TP-Begin-Dialogue-Req vorangehen. Abbildung 8 stellt ein Beispiel für eine kausale Zeitstruktur beim sukzessiven Aufbau eines Transaktionsbereiches dar.

tionsbereiches dar. Sie bezieht sich auf den Anfang von Transaktion TB aus Abbildung 7 auf Seite 50 und den Transaktionsbereich TB2 aus Abbildung 5 auf Seite 43. Dem Dialog d13 zwischen den Prozessen p1 und p3 wird mit TP-Begin-Transaction eine Transaktionsverbindung zugeordnet. Der Dialog d14 wird als Transaktionsverbindung einer unverketteten Transaktion mit TP-Begin-Dialogue neu eröffnet. Die Primitive TP-Begin-Dialogue-Req und TP-Begin-Transaction-Req stehen an der Schnittstelle in keinem kausalen Zusammenhang. Aus dem Beispiel wird deutlich, daß anwendungsspezifische Aufrufe von TP-Diensten, in diesem Fall TP-Data-Dienste, kausal abhängig von der Integration des Anwendungsprozesses in den Dialogbaum sind. Jede Transaktionsverbindung ist nebenläufig bezüglich der Übertragungsrichtungen.

5.1.2 Ausbau eines Transaktionsbereiches für eine Transaktionskette

Der Transaktionsbereich einer Transaktionskette wird von der Vorgängers-Transaktion vererbt und kann durch Aufruf des Dienstes TP-Begin-Dialogue erweitert werden. Der Dienst ist auf die Datentransferphase beschränkt⁷.

Die kausalen Zusammenhänge am Anfang einer Transaktion innerhalb eines vererbten Transaktionsbereichs ergeben sich aus den kausalen Beziehungen beim Transaktionsabschluß der vorangehenden Transaktion. Abbildung 9 zeigt die Integration des Prozesses p7 in den Transaktionsbereich TB1 aus Abbildung 5 auf Seite 43. Dieser Dialogaufbau wird am Anfang von Transaktion T1.2 (siehe Abbildung 6 auf Seite 48) eingeleitet. Die Dialoge d25 und d26 wurden dagegen von der Transaktion T1.1 vererbt. Das Ende der Transaktion T1.1 ist an der TP-Schnittstelle durch die kausalen Bezugszeitpunkte TP-Commit-Complete-Ind sichtbar.

⁷ Der Dienst TP-Begin-Transaction ist im Zusammenhang mit verketteten Transaktionen nicht einsetzbar. Mit dem Dialogaufbau wird durch Angabe entsprechender Parameter im Dienst TP-Begin-Dialogue gleichzeitig die Eröffnung der Transaktionsverbindung erreicht.

die Datentransferphase zu beenden. In beiden Fällen ist für den Transaktionsprozeß kein Aufruf anwendungsspezifischer Kommunikationsdienste für die betroffenen Verbindungen mehr möglich.

Das Ereignis TP-Prepare-Ind stellt für eine Subtransaktion nur eine kausale Beziehung zum Vorgänger her. Damit geht der Subtransaktion die Anforderung zur internen Entscheidung bezüglich des Transaktionsabschlusses zu. Ereignisse aus Diensten, die nur die Nachfolger einbeziehen, sind kausal nebenläufig zur TP-Prepare-Ind. Diese Nebenläufigkeit gilt auch für die Eröffnung einer neuen Transaktionsverbindung mit TP-Begin-Dialogue-Req bzw. TP-Begin-Transaction-Req.

Abbildung 10 beschreibt die kausale Abhängigkeit der Commit-Primitive beim Transaktionsabschluß für den Transaktionsbereich TBI aus Abbildung 5 auf Seite 43 für den Fall, daß jeder Transaktionsprozeß die Weiterleitung der Commit-Vorbereitung unmittelbar mit dem TP-Commit-Req durchführt. Die folgenden Erläuterungen lassen sich an diesem Beispiel nachvollziehen.

Der TP-Commit-Req einer Subtransaktion, die die Rolle eines Blattes im Transaktionsbereich einnimmt, zieht keine TP-Prepare-Ind nach sich. Die TP-Commit-Request-Primitive aller Subtransaktionen eines Transaktionsbereiches stehen in kausaler Vorgängerrelation zu den TP-Commit-Ind-Primitiven aller Subtransaktionen. Jede TP-Commit-Ind wird vom TP-Service-Provider generiert und bildet das kausale Vorgängerereignis für den TP-Done-Req der entsprechenden Subtransaktion. Mit dem TP-Done-Req bestätigt eine Subtransaktion, daß sie von der Entscheidung für den positiven Abschluß Kenntnis genommen hat. Der TP-Standard fordert nicht, daß die lokale Subtransaktion zu diesem Zeitpunkt die Aktualisierung der lokalen Transaktionsobjekte abgeschlossen hat. Mit einer TP-Commit-Complete-Ind zeigt der TP-Service-Provider einer Subtransaktion an, daß alle Subtransaktionen des unterlagerten Teilbaumes im Transaktionsbereich die Kenntnis der Entscheidung bestätigt haben. Es entsteht dabei eine kausale Abhängigkeit einer TP-Commit-Complete-Ind von dem lokalen TP-Done-Req und den TP-Done-Req-Primitiven aller direkten und indirekten Nachfolger.

Abbildung 11 stellt die alternative Vorbereitung aller Transaktionsprozesse zum Transaktionsabschluß mit den verbindungspezifischen Primitiven TP-Prepare-Req dar. Die Commit-Phase-1 unterscheidet sich von der mit TP-

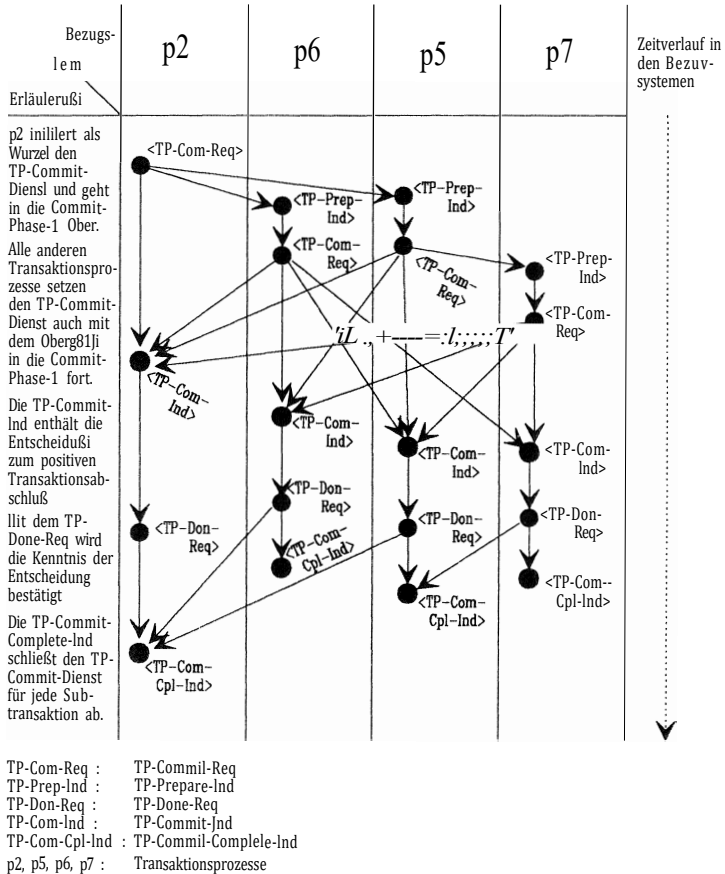


Abbildung 10 kausale Abhängigkeit beim Transaktionsabschluß

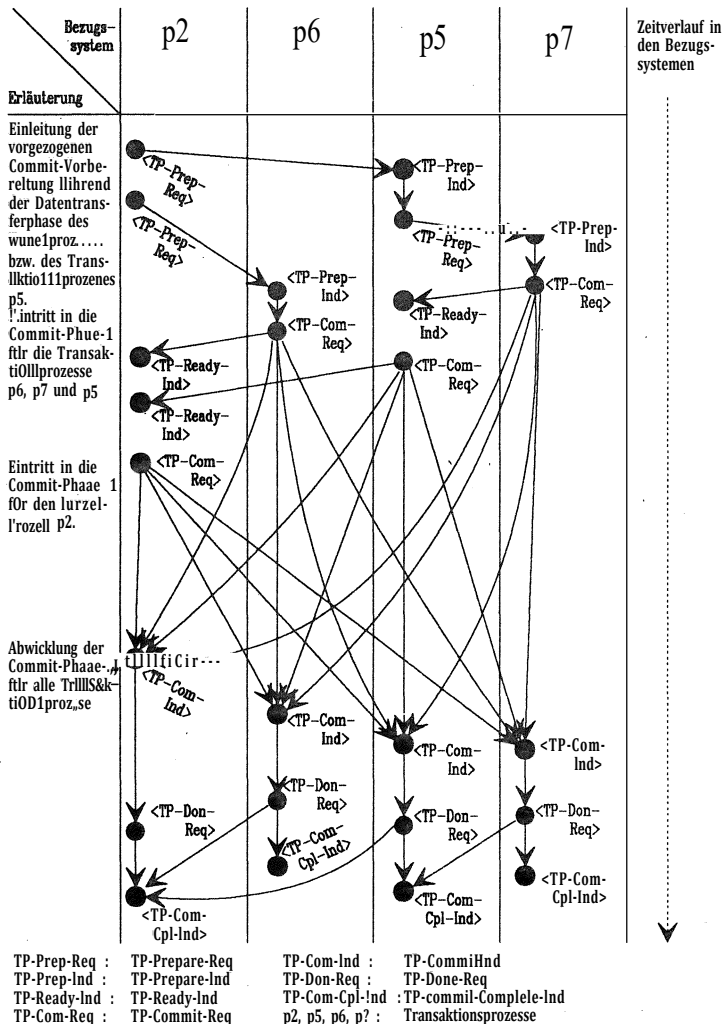


Abbildung 11: Commit-Vorbereitung während der Datentransferphasen

Commit-Req eingeleiteten Commit-Vorbereitung. Die Commit-Phase-2 wird davon nicht berührt. Der Abschluß der mit TP-Prepare-Req eingeleiteten Commit-Vorbereitung auf einer Verbindung wird dem Transaktionsprozeß mit einer verbindungs-spezifischen TP-Ready-Ind angezeigt, wenn der Transaktionsprozeß noch keinen TP-Commit-Req abgesetzt hat und sich demnach noch in der Datentransferphase befindet.

Mischformen der in beiden Abbildungen dargestellten Commit-Vorbereitung sind auch innerhalb eines Transaktionsprozesses möglich. Dies führt zu der Definition zweier Teilmengen $early(p)$ und $ready(p)$ der Nachfolgermenge $next(p)$ eines Transaktionsprozesses p :

$$early(p) = \{ q \mid \begin{array}{l} q \text{Enext}(p), \\ \text{TP - Prepare - Req}(q) - \text{TP - Prepare - Ind}(p) \end{array} \} \quad (4)$$

Diese Menge charakterisiert die Nachfolger, die durch einen vorzeitigen Aufruf zur Commit-Vorbereitung aufgefordert wurden.

$$ready(p) = \{ q \mid \begin{array}{l} q \text{Eearly}(p), \\ \text{TP - Commit - Req}(q) - \text{TP - Ready - Ind}(p) \end{array} \} \quad (5)$$

Die Menge $ready(p)$ charakterisiert die Nachfolgerprozesse, deren Bestätigung für den Abschluß der Commit-Vorbereitung noch während der Datentransferphase des Transaktionsprozesses p eintrifft bzw. abgewartet wird. Ein Transaktionsprozeß aus der Nachfolgermenge kann an der TP-Prepare-Ind nicht erkennen, ob sich sein Vorgänger noch in der Datentransferphase oder bereits in der Commit-Phase-1 befindet,

5.1.4 Negativer Abschluß einer Transaktion

Der negative Abschluß einer Transaktion kann von jeder Subtransaktion eingeleitet werden, die noch keinen TP-Commit-Req abgesetzt hat und damit noch keine Bereitschaft zum positiven Abschluß bekundet hat. Außerdem kann der TP-Service-Provider den negativen Abschluß erzwingen. Ein Transaktionsabbruch kann nicht mehr eintreten, wenn im Subsystem der Transaktionswurzel die globale positive Entscheidung gefällt wurde.

Mit der kausalen Zeitstruktur des Transaktionsabbruchs einer verketteten Transaktion, den ein Zwischenknoten eines Transaktionsbereiches eingeleitet.

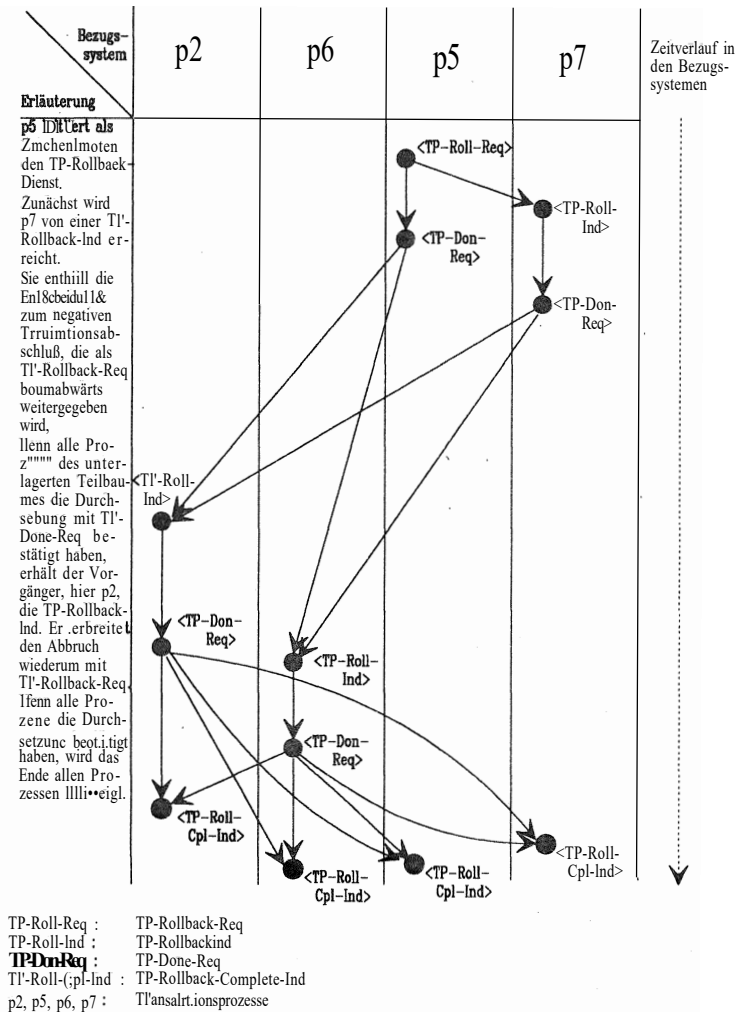


Abbildung 12: Kausale Zeitstruktur des TP-Rollback-Dienstes

hat, wird ein Beispiel für die Fortpflanzung des TP-Rollback-Dienstes dargestellt. Abbildung 12 zeigt, daß die Rollback-Einleitung in einem Knoten zunächst den diesem Knoten unterlagerten Teilbaum erfaßt⁸. Erst, wenn alle Subtransaktionen dieses Teilbaumes die negative Entscheidung bestätigt haben, erfaßt der Rollback-Ablauf den nächst höheren Knoten im Transaktionsbereich.

Der Dienst TP-Rollback kann von mehreren Subtransaktionen im Transaktionsbereich initiiert werden. In diesem Fall wachsen die vom Rollback erfaßten Teilbereiche des Transaktionsbereiches zusammen und breiten sich anschließend über die noch nicht erfaßten Subtransaktionen des Transaktionsbereiches aus. Dabei wird deutlich, daß die Abwicklung des Rollback-Dienstes keiner zentralen Koordination durch einen Transaktionsprozeß bedarf. Eine Abhängigkeit einer Subtransaktion von anderen Subtransaktionen ergibt sich beim negativen Abschluß erst nach der eigenen Bestätigung der Rollback-Entscheidung mit dem Dientsprimitiv TP-Done-Req. Diese Abhängigkeit ergibt sich aus dem Warten auf die Rollback-Complete-Indication. Sie behindert die Freigabe der von der Subtransaktion belegten Objekte nicht und ist daher unkritisch. Auf eine Verfeinerung des Rollback-Ablaufes im nachfolgenden Abschnitt wird deshalb verzichtet.

5.2 Verfeinerung der kausalen Zeitobjekte des TP-Service-Providers

Aus der Protokollabwicklung zur Bereitstellung der TP-Dienste TP-Begin-Dialogue, TP-Begin-Transaction und TP-Commit ergeben sich neben den Primitiven der TP-Schnittstelle interne Ereignisse des TP-Service-Providers und Sende- und Empfangsereignisse für PDUs, die in die Menge der Bezugszeitpunkte der verfeinerten Strukturen eingehen.

Der TP-Standard definiert eine Abbildung des TP-Dienstes auf einen intern verbindungspezifischen Dienst. Für TP-Dienste, die nur zwei Transaktionsprozesse einbeziehen, ergibt sich eine 1:1-Zuordnung zwischen den

⁸Das dargestellte Beispiel zeigt die kausale Zeitstruktur für den Dienst TP-Rollback in einer verketteten Transaktion.

Primitiven des internen Dienstes und den TP-Primitiven. Im Fall des TP-Commit-Dienstes betrifft der TP-Commit-Req eines Transaktionsprozesses p die Transaktionsverbindungen aller Nachfolger $p_x \in \text{next}(p)$. Es ergibt sich eine 1 : n-Abbildung, wobei $n = |\text{next}(p)|$ die Mächtigkeit der direkten Nachfolgermenge von p angibt. Die aus einer 1 : n-Abbildung hervorgerufenen internen Dienstprimitive sind kausal nebenläufig.

Für die Dienste zum Aufbau eines Transaktionsbereiches wurde aufgrund der 1:1-Zuordnung auf eine separate Ausweisung der internen Dienstprimitive verzichtet. Für den TP-Commit-Dienst sind alle internen Dienstprimitive aufgeführt, die aus einer 1:n-Abbildung hervorgehen oder den Endpunkt einer n:1-Abbildung bilden⁹.

TP-Dienst	Bezugszeitpunkte des internen Dienstes	Senden und Empfang von Protocol Data Units
TP-Begin-Dialogue		Begin-Dialogue-RI-PDU Begin-Dialogue-RC-PDU
TP-Begin-Transaction		Begin-Transaction-RI-PDU
TP-Commit	Int-Prepare-Req Int-Ready-Req Int-Commit-Req Int-Commit-Rsp	Prepare-RI-PDU Ready-RI-PDU Commit-RI-PDU Commit-RC-PDU

Req: Request **Ind:** Indication
Rsp: Response **Cnf:** Confirmation
Int: Internal **PDU:** Protocol Data Unit
RI: Request-Indication **RC:** Response-Confirmation

Tabelle 2: Bezugszeitpunkte der TP-Dienste im TP-Service-Provider

Die Tabelle 2 listet in Spalte 2 Primitive des internen Dienstes des TP-Service-Providers, soweit sie keiner 1:1-Abbildung unterliegen. In Spalte 3 sind die Protokolldateneinheiten aufgeführt, deren Sende- und Empfangsergebnisse an der Schnittstelle zwischen einem Kommunikationsmanager eines

⁹Der CCR-Dienst [CCR89] wird vom TP-Service-Provider intern und ebenfalls verbindungspezifisch benutzt. Die Primitive der CCR-Dienste ergeben sich aus einer 1:1-Abbildung zu dem bisher erwähnten internen verbindungspezifischen Dienst. Deshalb wird auf eine Angabe der CCR-Primitive verzichtet.

Subsystems und dem Kommunikationssystem sichtbar sind¹⁰.

5.2.1 Verfeinerung der Zeitobjekte für den Aufbau eines Transaktionsbereiches

Das TP-Service-Provider-Zeitobjekt eines TP-Begin-Dialogue-Dienstes wird durch eine Folge von vier Zeitobjekten verfeinert. Für das entsprechende Zeitobjekt des TP-Begin-Transaction-Dienstes ergibt sich ebenfalls eine Folge von vier Zeitobjekten aus der Zerlegung. Die vier Zeitobjekte repräsentieren folgende Aktionen:

1. Umsetzung des Request-Primitivs in eine PDU.
2. Verzögerung der PDU durch Einlagerung in einen verbindungspezifischen Puffer.
3. Übertragung der PDU.
4. Umsetzung der PDU in ein Indication-Primitiv.

Die Einlagerung von PDUs in einen verbindungspezifischen Puffer wird in TP durch Konkatenationsregeln definiert. Mit der **Konkatenation** wird eine Optimierung des Nachrichtenflusses erreicht. Die Konkatenation von PDUs im Sendepuffer ist nicht zwingend vorgeschrieben. Dagegen muß jede TP-Protokollmaschine die Auflösung von empfangenen Konkatenationssequenzen implementieren. Das letzte Element einer übertragenen Konkatenationssequenz ist von der Pufferung nicht betroffen. Für die beiden PDUs, die bei der Eröffnung einer Transaktionsverbindung auftreten können, ist die Pufferung zulässig.

Abbildung 13 stellt die beiden kausalen Zeitstrukturen dar.

5.2.2 Verfeinerung der Zeitobjekte für den positiven Transaktionsabschluß

Für den TP-Commit-Dienst müssen fünf durch die Ereignispaare

¹⁰Die Protokolldateneinheiten von TP und CCR werden aufgrund einer 1:1-Abbildung nicht unterschieden.

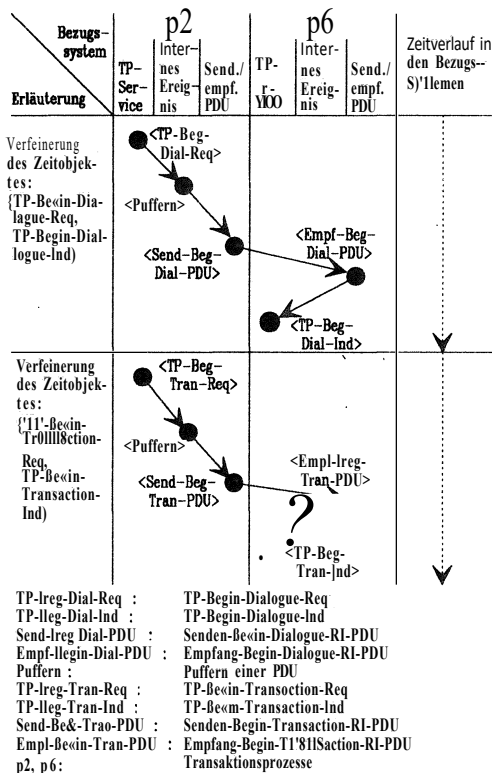
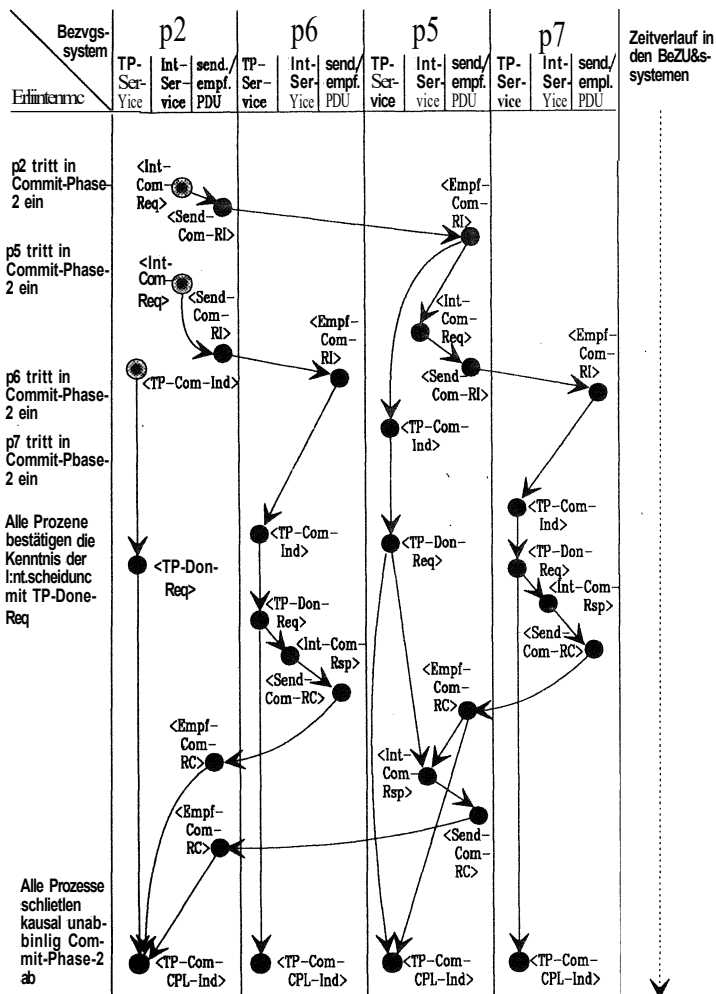


Abbildung 13: Verfeinerung der Zeitobjekte für den Aufbau eines Transaktionsbereiches



TP-Com-Req :	TP-Commit-Req	Send-Prep-RI :	Senden-Prepare-RI-PDU
TP-COM-Ind :	TP-Commit-Ind	Send-Ready-HI :	Senden-Ready-RI-PDU
TP-Com-Cpl-Ind :	n>-Commit-Complete-Ind	Send-Com-RI :	Senden-Conunit-RI-PDU
		Send-Com-RC :	Senden-Commit-RC-PDU
TP-Prep-Req :	n>-Prepore-Req	Empl-Prep-RI :	Empfang-Prepare-RI-PDU
TP-Prep-Ind :	TP-Prepare-Ind	Empl-Ready-RI :	Empfang-Ready-RI-PDU
TP-Ready-Ind :	TP-Ready-Ind	Empl-Com-RI :	Empfang-Commit-RI-PDU
TP-Don-Req :	TP-Done-Req	Empl-Com-RC :	Empfang-Commil-RC-PDU
Int-Prep-Req :	Intenal-Prepare-Req	p2, p5, p6, p7 : Transaktionsprozesse	
Int-Ready-Req :	Intrenal-Ready-Req		
Int-Com-Req :	Intenal-Commit-Req		
Int-Com-Rsp :	Intenal-Commit-Rsp		

- (TP-Commit-Req, TP-Prepare-Ind),
- (TP-Prepare-Req, TP-Prepare-Ind),
- (TP-Commit-Req, TP-Ready-Ind),
- (TP-Commit-Req, TP-Commit-Ind) und
- (TP-Done-Req, TP-Commit-Complete-Ind)

gekennzeichnete Zeitobjekte verfeinert werden. Dabei sind die internen Dienstprimitive zu berücksichtigen. Für die komplexe kausale Zeitstruktur an der TP-Schnittstelle für den Transaktionsbereich TB! aus Abbildung 5, Seite 43, wird in den beiden Abbildungen 14 und 15 die Verfeinerung dargestellt. Sie beziehen sich auf die Abbildung 10 auf Seite 59. Die durch Schattierung hervorgehobenen Übergangsereignisse der beiden Abbildungen müssen für die vollständige Darstellung des Protokollflusses in Deckung gebracht werden.

Für die vorgezogene Commit-Vorbereitung in Abbildung 11 auf Seite 60 wird die Verfeinerung in Abbildung 16 ebenfalls für den Transaktionsbereich TB! aus Abbildung 5 dargestellt. Diese Abbildung wird auch durch Teil 2 des Protokollablaufs für den Transaktionsabschluß, d.h. durch Abbildung 15, fortgesetzt.

Die in diesen Abläufen enthaltenen PDUs dürfen nicht durch konkatenationsbedingte Pufferung verzögert werden. Soweit die Konkatenation einer hier aufgeführten PDU erlaubt ist, bildet sie den Abschluß einer Konkatenationssequenz und beendet damit die konkatenationsbedingte Pufferung für bereits im Puffer vorliegende PDUs.

5.3 Kommunikations.intervalle eines Transaktionsbereiches

Aus der Verknüpfung der verfeinerten kausalen Zeitstrukturen der TP-Dienste, die die Kommunikationszeiträume der Transaktionsprozesse begrenzen, lassen sich kausale Zeitintervalle ableiten, die die 3 Bearbeitungsphasen der Transaktionsprozesse wiedergeben.

Datentransferintervall: D [Danf, Clanf_p, bzw.]C2end, Clanf_p,

Die beiden Ausprägungen eines Datentransferintervalls ergeben sich aus der Tatsache, daß der Anfang dieses Intervalls in einer verketteten Transaktion durch das Endeereignis der vorangehenden Transaktion bestimmt sein kann, aber nicht sein muß.

Commit-Phase-1-Intervall: C1 [Clanf, C2anf_p,

Commit-Phase-2-Intervall: C2 [C2anf, C2end_p];

Die Menge der zulässigen Ereignispaare zur Begrenzung dieser Intervalle hängt zum einen vom ausgewählten Transaktionskonzept Π , zum anderen von der Rolle¹² eines Subsystems im Transaktionsbereich ab. Diese Intervalle sind prozeßspezifisch, nicht verbindungspezifisch, definiert. Die Zugehörigkeit eines Intervalls zu einem Transaktionsprozeß drückt sich im Index p aus. Für die Intervalle werden für jede Rolle eines Transaktionsprozesses drei Ereignismengen $E'o, EcJ_u, E\mathcal{J}_2, Ert, E0j, E02, E\mathfrak{F}, E_1$ und E_2 angegeben. Der hochgestellte Index deutet auf die Rolle des Transaktionsprozesses hin. Der untere Index gibt die Kommunikationsphase wieder. Die Ereignismengen enthalten die Aufzählung der Ereignisse, die in jedem an einer erfolgreichen Transaktion beteiligten Transaktionsprozeß auftreten.

Aus den nachfolgend angegebenen Ereignisteilmengen entsteht durch Vereinigung die Ereignismenge E_p , eines Transaktionsprozesses, die alle für einen Transaktionsabschluß notwendigen Ereignisse enthält:

¹¹Verkettete bzw. unverkettete Transaktion

¹²Wurzel, Zwischenknoten, Blatt

$$\begin{aligned}
E_w &= E_D \cup \bar{E}_a, \cup E_{a2} \\
E_{ZW} &= E_{TW} \cup E_{ZW} \cup E_{TW} \\
E_{bl} &= E_{b1} \cup E_{b1} \cup E_{b1}
\end{aligned}$$

5.3.1 Das Datentransferintervall eines Transaktionsprozesses

Folgende Datentransferintervalle sind unter den angegebenen Einschränkungen definiert:

- $[TP - \textit{Begin} - \textit{Dialogue} - \textit{Req}, TP - \textit{Commit} - \textit{Req}]_v$,
eingeschränkt auf die Transaktionswurzel
- $[TP - \textit{Commit} - \textit{Complete} - \textit{Ind}, TP - \textit{Commit} - \textit{Req}]_p$,
eingeschränkt auf einen Transaktionsprozeß einer verketteten Transaktion
- $[Senden - \textit{Commit} - RC - PDU, TP - \textit{Commit} - \textit{Req}]_v$,
eingeschränkt auf Transaktionsprozesse einer verketteten Transaktion ohne Wurzelfunktionalität
- $[TP - \textit{Begin} - \textit{Transaction} - \textit{Req}, TP - \textit{Commit} - \textit{Req}]_p$,
eingeschränkt auf die Transaktionswurzel einer unverketteten Transaktion
- $[TP - \textit{Begin} - \textit{Dialogue} - \textit{Ind}, TP - \textit{Commit} - \textit{Req}]_v$,
eingeschränkt auf Transaktionsprozesse ohne Wurzelfunktionalität
- $[TP - \textit{Begin} - \textit{Transaction} - \textit{Ind}, TP - \textit{Commit} - \textit{Req}]_r$,
eingeschränkt auf Transaktionsprozesse einer unverketteten Transaktion ohne Wurzelfunktionalität

Für Prozesse, die in eine Transaktion einer Kette eingebunden sind, beginnt das Datentransferintervall mit einer offenen Intervallgrenze nach dem Abschluß der vorangehenden Transaktion. Für jeden Transaktionsprozeß trifft genau eines der aufgezählten Datentransferintervalle zu. Das Anfangsereignis dieses Intervalls wird zukünftig mit **Danf** bezeichnet, um die wiederholte Aufzählung aller möglichen Anfangsereignisse zu vermeiden.

Das Endeereignis des Intervalls wird bereits dem Commit-Phase-1-Intervall zugeordnet. Die rollenspezifischen Ereignismengen der Transaktionsprozesse werden nachfolgend erläutert:

Für die Ereignisse, die sich in einem Transaktionsprozeß vor dem Eintreten der TP-Prepare-[nd ereignen, wird eine nicht näher spezifizierte Menge $\{D - \text{Ereignisse} - \text{vor} - TP - \text{Prepare} - \text{Ind}\}$ eingeführt. Für die Ereignisse, die nach der TP-Prepare-[nd eintreten, wurde entsprechend eine Menge $\{D - \text{Ereignisse} - \text{nach} - TP - \text{Prepare} - \text{Ind}\}$ angegeben. Diese Menge enthält alle Ereignisse *TP-Prepare-Req* und *Int-Prepare-Req* der Verbindungen zu den Nachfolgern aus der Menge *early(p)* eines Prozesses p und die Ereignisse *Empfang-Ready-RI-PDU* und *TP-Ready-Ind* der Verbindungen zu den Nachfolgern aus der Menge *ready(p)*. Kausale Zusammenhänge der Ereignisse dieser Mengen werden nicht betrachtet.

Ereignismengen der Datentransferintervalle:

$$E_D^w := \{ \text{Danf}, \text{Puffern} \} \\ \cup \{ D - \text{Ereignisse} - \text{nach} - TP - \text{Prepare} - \text{Ind} \}$$

$$E_{TP}^w := \{ \text{Danf}, \text{Empfang} - \text{Prepare} - \text{RI} - \text{PDU}, \\ \text{TP} - \text{Prepare} - \text{Ind} \} \\ \cup \{ D - \text{Ereignisse} - \text{vor} - TP - \text{Prepare} - \text{Ind} \} \\ \cup \{ D - \text{Ereignisse} - \text{nach} - \text{Prepare} \}$$

$$E_{\cancel{TP}}^w := \{ \text{Danf}, \text{Empfang} - \text{Prepare} - \text{RI} - \text{PDU}, \\ \text{TP} - \text{Prepare} - \text{Ind} \} \\ \cup \{ D - \text{Ereignisse} - \text{vor} - TP - \text{Prepare} - \text{Ind} \} \\ \cup \{ D - \text{Ereignisse} - \text{nach} - TP - \text{Prepare} - \text{Ind} \}$$

Während des Datentransferintervalls ergibt sich die Möglichkeit, die Einhaltung lokaler Zeitbedingungen durch lokale Zeitüberwachung zu gewährleisten. Die Überschreitung eines vorgegebenen Zeitlimits für die Dauer des Datentransferintervalls läßt sich als internes Ereignis für die Auslösung des Transaktionsabbruchs einsetzen.

5.3.2 Das Commit-Phase-1-Intervall eines Transaktionsprozesses

Das Commit-Phase-1-Intervall beginnt für jeden Transaktionsprozeß mit dem Ereignis TP-Commit-Request. Es endet, sobald die lokale Entscheidungsverkündung oder die Propagierung der Entscheidung an einen Nachfolger eingeleitet wurde.

$$\bullet [TP - Commit - Req, Int - Commit - Req]_w$$

$$\bullet [TP - Commit - Req, TP - Commit - Ind]_w$$

Die Intervallgrenze zwischen Commit-Phase-1-Intervall und Commit-Phase-2-Intervall ist an der TP-Schnittstelle nicht exakt erfaßbar. Die TP-Protokollmaschine des Wurzelprozesses beendet das Commit-Phase-1-Intervall nach dem Empfang der Ready-RI-PDUs aller direkten Nachfolger, für die noch keine Ready-RI-PDU während des Datentransferintervalls eingetroffen ist. Sie trifft die Entscheidung und kann dabei auch interne Bedingungen berücksichtigen. Die TP-Protokollmaschine des Wurzelsubsystems kann deshalb die Zeitdauer bis zum Eintreffen aller Zustimmungen zum Transaktionsabschluß durch lokale Zeitüberwachung begrenzen und ggf. die Transaktion zurück

setzen.

Aus diesem Intervall wird eine der folgenden Ereignismengen als Teilmenge in die Ereignismenge einer Subtransaktion übernommen:

Ereignismengen der Commit-Phase-1-Intervalle:

$$\begin{aligned}
 E_{C1}^w &:= \{ TP - Commit - Req \} \\
 \cup & \{ \forall \pi_1: E_{next}(\pi_1) : Int - Prepare - Req(p_x), \\
 & \quad \text{Rk } \pi_1 \text{ early}(\pi_1) \quad \quad \quad Senden - Prepare - RI - PDU(p_x) \} \\
 \cup & \{ \forall p_x: E_{next}(\pi_1) : Empfang - Ready - RI - PDU(p_x) \} \\
 & \quad p, \pi_1 \text{ ready}(p, \pi_1)
 \end{aligned}$$

$$\begin{aligned}
 E_{C1}^{zw} &:= \{ TP - Commit - Req \} \\
 \cup & \{ \forall p_x: E_{next}(\pi_1) : Int - Prepare - Req(p_x), \\
 & \quad \text{Rk } \pi_1 \text{ f early}(\pi_1) \quad \quad \quad Senden - Prepare - RI - PDU(p_x) \} \\
 \cup & \{ \forall p_x: E_{next}(\pi_1) : Empfang - Ready - RI - PDU(p_x) \} \\
 & \quad p, \pi_1 \text{ ready}(p, \pi_1) \\
 \cup & \{ Int - Ready - Req(prev(p;)), \\
 & \quad \quad \quad Senden - Ready - RI - PDU \}
 \end{aligned}$$

$$\begin{aligned}
 E_{C1} &:= \{ TP - Commit - Req, \\
 & \quad Int - Ready - Req(prev(p;)), \\
 & \quad \quad \quad Senden - Ready - RI - PDU \}
 \end{aligned}$$

5.3.3 Das Commit-Phase-2-Intervall eines Transaktionsprozesses

Das Commit-Phase-2-Intervall beginnt für jeden Transaktionsprozeß entsprechend dem Ende des Commit-Phase-1-Intervalls mit dem ersten der Ereignisse TP-Commit-Ind für die lokale Commit-Verkündung oder Int-Commit-Req für eine Nachfolger-Subtransaktion. Das Intervallende tritt mit dem letzten der Ereignisse TP-Commit-Complete-Ind oder dem Absenden des Commit-RC-PDU an die Vorgänger-Subtransaktion ein.

- $[Int - Commit - Req, TP - Commit - Complete - Ind]_p$,
- $[TP - Commit - Ind, TP - Commit - Complete - Ind]_p$,

- $[Int - Commit - Req, Senden - Commit - RC - PDU]_p$,
für Subsysteme ohne Wurzelfunktionalität
- $[TP - Commit - Ind, Senden - Commit - RC - PDU]_v$,
für Subsysteme ohne Wurzelfunktionalität

In beiden Intervallen kennt der Transaktionsprozeß die gemeinsam getroffene Entscheidung und ist daran gebunden. Durch lokale Zeitüberwachung kann der positive Abschluß der Transaktion nicht mehr verhindert werden. Als Teilmengen der rollenabhängigen Ereignismengen ergeben sich für dieses Intervall:

Ereignismengen der Commit-Phase-2-Intervalle:

$$\begin{aligned}
 E_{f_2} := & \{ TP - Commit - lnd(p;)\} \\
 \cup & \{ Vp_x \ E \ next(p,): \ lnt - Commit - Req(p_x), \\
 & \quad \quad \quad Senden - Commit - RI - PDU(p_x), \\
 & \quad \quad \quad Empfang - Commit - RC - PDU(p_x)\} \\
 \cup & \{ TP - Done - Req(p;), \\
 & \quad \quad \quad TP - Commit - Complete - lnd(p;)\}
 \end{aligned}$$

$$\begin{aligned}
 E_0 := & \{ TP - Commit - lnd(p;)\} \\
 \cup & \{ Vp_x \ E \ next(pi) : \ lnt - Commit - Req(p_{..}), \\
 & \quad \quad \quad Senden - Commit - RI - PDU(p_{..}), \\
 & \quad \quad \quad Empfang - Commit - RC - PDU(p_{..})\} \\
 \cup & \{ TP - Done - Req(p;), \\
 & \quad \quad \quad TP - Commit - Complete - lnd(p;), \\
 & \quad \quad \quad lnt - Commit - Rsp(prev(p;)), \\
 & \quad \quad \quad Senden - Commit - RC - PDU(p;)\}
 \end{aligned}$$

$$\begin{aligned}
 E_2 := & \{ TP - Commit - lnd(p;)\} \\
 \cup & \{ Vp_x \ E \ next(p,): \ lnt - Commit - Req(p_x), \\
 & \quad \quad \quad Senden - Commit - RI - PDU(p_{..}), \\
 & \quad \quad \quad Empfang - Commit - RC - PDU(p_{..})\} \\
 \cup & \{ TP - Done - Req(p;), \\
 & \quad \quad \quad TP - Commit - Complete - lnd(p;), \\
 & \quad \quad \quad lnt - Commit - Rsp(prev(p;)), \\
 & \quad \quad \quad Senden - Commit - RC - PDU(p;)\}
 \end{aligned}$$

6 Zusammenhänge der Kommunikationsaktivitäten eines Transaktionssystems

Die Einführung von Aktivitäten¹³ erfolgt in der Absicht, die Bearbeitung von Kommunikationsdiensten als Verknüpfung atomarer Aktionen der Kommunikationsmanager und des Kommunikationssystems darzustellen. Ein Startereignis aus dem Ereignisraum E und eine Zeitdauer beschreiben eine Aktivität.

Im nächsten Abschnitt 6.1 wird eine Erweiterung der Ereignismenge um interne Ereignisse eines Kommunikationsmanagers vorgenommen. Die neuen Ereignisse sollen die Modellierung von konkurrierenden Zugriffen und Wartesituationen ermöglichen. Auf dieser Ereignismenge werden anschließend die Kompositionen von Aktivitäten während der drei Phasen eines Transaktionsprozesses aufgebaut. Das Kapitel endet mit einer Interpretation der Ausführungszeiten der Aktivitätskompositionen für Monoprozessor-Subsysteme.

6.1 Die Menge der Kommunikationsaktivitäten

Die Menge der Kommunikationsaktivitäten innerhalb einer verteilten Transaktion gliedert sich in Aktivitäten der Kommunikationsmanager, Übertragungsaktivitäten des Kommunikationssystems und Kommunikationsaktivitäten der Subtransaktionen. Innerhalb der Subtransaktionen sind Kommunikationsaktivitäten von anderen Aktivitäten ohne Kenntnis der Spezifikation nicht unterscheidbar. Alle TP-Indication- und TP-Confirmation-Primitive bilden Startereignisse für Kommunikationsaktivitäten der Subtransaktionen. Sie werden hier nicht behandelt. Für die Aktivitäten der Kommunikationsmanager und des Kommunikationssystems folgt eine mögliche Unterteilung in atomare Bearbeitungsabschnitte unter Angabe der dafür geltenden Voraussetzungen.

¹³Der Begriff der Aktivität wurden auf Seite 34 eingeführt.

6.1.1 Aktivitäten eines Kommunikationsmanagers

Alle bisher eingeführten Bezugszeitpunkte einer verteilten Transaktion mit Ausnahme der Sendeereignisse werden von den Kommunikationsmanagern der beteiligten Subsysteme entsprechend den im TP-Standard definierten Abbildungen bearbeitet. Diese Abbildungen betreffen nur die TP-Protokollmaschinen, ihre Beziehungen zur TP-Schnittstelle und die Anbindung an die unterlagerte Darstellungsschicht des ISO-Referenzmodells.

Neben diesen Vorgaben können sich auch Verzögerungen aus dem konkurrierenden Ablauf mehrerer TP-Protokollmaschinen in einem Kommunikationsmanager und aus dem konkurrierenden Zugriff der Subtransaktionen auf den Kommunikationsmanager ergeben.

Für das Datentransferintervall wird auf die Auflösung der Kommunikationsaktivitäten verzichtet. Das 2-Phasen-Commit-Protokoll führt zu Wartesituationen eines Kommunikationsmanagers bezüglich der Verbindungen einer TP-Protokollmaschine und der zugehörigen Subtransaktion. In der Wartesituation ist eine Verbindung bis zum Eintreffen eines definierten Ereignisses vom konkurrierenden Zugriff ausgenommen. Das Warten auf ein von der Subtransaktion erzeugtes Request-Ereignis verzögert die Generierung von Schnittstellenereignissen durch den Kommunikationsmanager.

Die neu erfaßten Situationen führen zur Ergänzung der in Kapitel 5 durch die Mengen E^w , $E^{\bullet w}$ und E^h angegebenen Bezugszeitpunkte um zwei weitere Arten von Ereignissen:

- *Queuing*(p;)
- *Warten*(p;)

Queuing: Verzögerungen, die sich aus dem konkurrierenden Zugriff auf Ressourcen ergeben, führen zur Bildung von Warteschlangen. Sie werden mit dem Begriff "Queuing" belegt. Queuing tritt im Kommunikationsmanager an der TP-Schnittstelle und an der Schnittstelle zum Kommunikationssystem auf. p; gibt den Transaktionsprozeß an, der als Verursacher des in die Warteschlange eingereihten Ereignisses auftritt.

Warten: Aus der Abwicklung des 2-Phasen-Commit-Protokolls können Wartesituationen entstehen, die eine Beteiligung an einer Konkurrenzsituation verhindern. Warteereignisse leiten Zeiträume der Untätigkeit des Kommunikationsmanagers bezüglich einer Verbindung oder der Schnittstelle zu einer Subtransaktion ein. p; gibt den Transaktionsprozeß an, der als Verursacher des erwarteten Ereignisses auftritt.

Das Anfangsereignis der Datentransferphase *Danf* führt nur im Kommunikationsmanager eines Wurzelprozesses zu einer Aktivität *während der Datentransferphase*. Für Zwischenknoten und Blattprozesse beginnt die Datentransferphase mit einem TP-Indication-Primitiv und hat eine Aktivität, der jeweiligen Subtransaktion zur Folge.

Die Aktivität zur Verarbeitung des Puffer-Ereignisses im Kommunikationsmanager der Transaktionswurzel wird in die Menge der Ereignisse {*Dnach - TP - Prepare - Ind*} aufgenommen und tritt deshalb in den Aktivitätskompositionen nicht in Erscheinung.

Als Ereignismenge eines Kommunikationsmanagers ergibt sich für die Unterstützung einer Subtransaktion unter Einbeziehung der neuen Ereignisse folgende Menge:

$$\begin{aligned}
E(p_i) = & \{ \text{Danf}, \\
& \text{Queuing}(p_i), \\
& \text{TP - Commit - Req}(p_i), \\
& \text{Warten}(p_i) \\
& \text{TP - Done - Req}(p_i), \\
& \} \\
\cup & \{D - \text{Ereignisse - nach - TP - Prepare - Ind}\} \\
& \text{Queuing}(\text{prev}(p_i)), \\
\cup & \{p_i \text{ i- w: Empfang - Prepare - RI - PDU}(\text{prev}(p_i)), \\
& \text{/nt - Ready - Req}(\text{prev}(p_i)), \\
& \text{Warten}(\text{prev}(p_i)), \\
& \text{Empfang - Commit - RI - PDU}(\text{prev}(p_i)), \\
& \text{Int - Commit - Rsp}(\text{prev}(p_i)) \\
& \} \\
\cup & \{p_i \text{ f- w: } \{D - \text{Ereignisse - vor - TP - Prepare - Ind}\}\} \\
\cup & \{ \text{Vp!E next}(p_i) : \text{Int - Prepare - Req}(p_x), \\
& \text{P! i! early}(p_i) \text{ Warten}(p_x) \\
& \} \\
\cup & \{ \text{Ypx E next}(p_i), : \text{Queuing}(p_x), \\
& \text{Px f ready}(p_i) \text{ Empfang - Ready - RI - PDU}(p_x) \\
& \} \\
\cup & \{ \text{Vpx E next}(p_i): \text{Int - Commit - Req}(p_x), \\
& \text{Queuing}(p_x), \\
& \text{Empfang - Commit - RC - PDU}(p_x) \\
& \}
\end{aligned}$$

Alle Ereignisse dieser Ereignismenge bilden Startereignisse von Aktivitäten eines Kommunikationsmanagers.

6.1.2 Aktivitäten des Kommunikationssystems

Die Sendeereignisse leiten eine Übertragungsaktivität des Kommunikationssystems ein. Mit der Abbildung der ISO-Schichten 1 bis 6 auf das Kommunikationssystem werden alle aus diesen Schichten resultierenden Einflüsse auf das Zeitverhalten in den durch Sendeereignisse eingeleiteten Übertra-

gungsaktivitäten zusammengefaßt. Die während der Datentransferphase auftretenden Übertragungen sind von dem anwendungsspezifischen Kommunikationsverhalten abhängig. Sie werden deshalb in der Ereignismenge des Kommunikationssystems nicht berücksichtigt. Die Übertragung einer PDU zur Eröffnung einer Transaktionsverbindung liegt noch außerhalb des Datentransferintervalls des empfangenden Transaktionsprozesses und wird deshalb nicht aufgeführt. Die Ereignismenge des Kommunikationssystems k reduziert sich auf die Ereignisse, die Übertragungen während des 2-Phasen-Commit-Protokolls einleiten:

$$E(k) =$$

$$\begin{aligned} & \{p, f w: \text{Senden} - \text{Ready} - \text{RI} - \text{PDU}(p;, \text{prev}(p,))\}U \\ & \{V_{p,} \text{Next}(p;) \setminus \text{early}(p;): \text{Senden} - \text{Prepare} - \text{RI} - \text{PDU}(p;, p,)\}U \\ & \{V_{p,} \text{Next}(p;): \text{Senden} - \text{Commit} - \text{RI} - \text{PDU}(p;, p,), \\ & \quad \text{Senden} - \text{Commit} - \text{RC} - \text{PDU}(p;, p,)\} \end{aligned}$$

Die Sendeereignisse sind durch die Angabe der Verbindungspartner einer Transaktionsverbindung eindeutig zugeordnet. Dabei werden folgende Eigenschaften für das Kommunikationssystem vorausgesetzt:

- Nachrichten gehen nicht verloren.
- Nachrichten werden nicht verfälscht.
- Erhalt der Reihenfolge wird garantiert.
- Alle Verbindungen sind voneinander unabhängig parallel zu betreiben.

Die ersten drei Eigenschaften werden durch die Ebenen 1-6 des ISO-Referenzmodells verwirklicht. Die letzte Eigenschaft wird gefordert, um konkurrierenden Zugriff auf das Kommunikationssystem eines verteilten Transaktionssystems auszuschließen.

6.2 Beschreibung der Kommunikationsphasen durch Aktivitätskompositionen

Zunächst werden die lokalen Aktivitäten eines Kommunikationsmanagers in verschachtelte sequentielle und parallele Verknüpfungen überführt. Die

Warteaktivitäten eines Kommunikationsmanagers werden danach mit überdeckenden Aktivitäten des Kommunikationssystems und anderer Kommunikationsmanager in Beziehung gesetzt. Dabei wird vorausgesetzt, daß unmittelbar nach der Bearbeitung eines Startereignisses der TP-Schnittstelle (TP-Request- oder TP-Response-Primitiv) bzw. der Schnittstelle zum Kommunikationssystem (Empfang einer PDU) alle während dieser Aktivität generierten internen Ereignisse (Internal-Request, Internal-Response) bearbeitet werden. Dadurch wird das Queuing auf die TP-Schnittstelle und die Schnittstelle zum Kommunikationssystem beschränkt. Zur Darstellung der Aktivitätskompositionen kommt folgende Schreibweise zum Einsatz:

- $(\text{Ereignis}(p_x), d_j) b$: beschreibt eine Aktivität mit einem von dem Transaktionsprozeß P_x abhängendem Startereignis und einer logischen Zeitdauer d_j . Mit den Indizes j wird eine Unterscheidung der logischen Zeiträume zur Bearbeitung der Ereignisse eingeführt. Der Index b einer Aktivität verweist auf die bearbeitende Komponente, d.h. den Transaktionsprozeß bzw. das Kommunikationssystem.
- $\langle \quad \rangle$: klammert eine Folge von Aktivitäten oder Aktivitätskompositionen.
Die Elemente der Folge sind durch Kommata getrennt.
- $\{ \quad \}$: klammert eine Menge von parallelen Aktivitäten oder Aktivitätskompositionen.
Die Elemente der Menge sind durch Kommata getrennt.
- $[\quad]$: klammert eine Menge von bedingt auftretenden Aktivitäten oder Aktivitätskompositionen.

Die nachfolgende Tabelle 3 enthält eine Aufzählung aller von den Kommunikationsmanagern bzw. dem Kommunikationssystem auszuführenden Aktivitäten und legt die Indizierung der logischen Ausführungszeiten fest.

Startereignis	Dauer
(Dan/	<i>di</i>)
<i>(e E D - Ereignisse - vor - TP - Prepare - Ind,</i>	<i>d2</i>)
<i>(Queuing,</i>	<i>d3</i>)
<i>(Empfang - Prepare - RI - PDU,</i>	<i>d4</i>)
<i>(e E D - Ereignisse - nach - TP - Prepare - Ind,</i>	<i>ds</i>)
<i>(TP - Commit - Req,</i>	<i>ds</i>)
<i>(Int - Prepare - Req,</i>	<i>dl</i>)
<i>(Warten(p;),</i>	<i>ds</i>)
<i>(Int - Ready - Req,</i>	<i>d9</i>)
<i>(Empfang - Ready - RI - PDU,</i>	<i>dl0</i>)
<i>(Warten,</i>	<i>d11</i>)
<i>(Empfang - Commit - RI - PDU,</i>	<i>dl2</i>)
<i>(Warten,</i>	<i>dl3</i>)
<i>(TP - Done - Req,</i>	<i>d14</i>)
<i>(Int - Commit - Req,</i>	<i>dl5</i>)
<i>(Warten,</i>	<i>dl6</i>)
<i>(Empfang - Commit - RC - PDU,</i>	<i>dl1</i>)
<i>(Int - Commit - Rsp,</i>	<i>dl5</i>)
<i>(Senden,</i>	<i>dm</i>)

Tabelle 3: Liste der Aktivitäten

6.2.1 Verknüpfung der Aktivitäten eines Kommunikationsmanagers

In diesem Abschnitt werden die in Kapitel 5.3 definierten Intervalle eines Transaktionsprozesses in Aktivitäten des Kommunikationsmanagers zerlegt, um Aufschluß über die Wartezustände zu erhalten. In den kausalen Zeitstrukturen von Kapitel 5 lassen sich sequentielle Aktivitäten aus der kausalen Abhängigkeit und parallelisierbare Aktivitäten aus der kausalen Nebenläufigkeit von Zeitobjekten ableiten.

Das Datentransferintervall eines Kommunikationsmanagers ohne Wurzelfunktionalität wird in folgende Aktivitäten zerlegt:

$[D]anf, C!anf[_{p,f}]$

\langle $\{ \forall e \in \{D - Ereignisse - vor - TP - Prepare - Ind\} : (e, d_2) \}$
 $(Queuing(prev(p), d_1)_{p'},$
 $(Empfang - TP - Prepare - RI - PDU(prev(p)), d_4)_{p'},$
 $\{ \forall e \in \{D - Ereignisse - nach - TP - Prepare - Ind\} : (e, d_5) \}$
 \rangle :

Dabei wird für alle Ereignisse der beiden Mengen $\{D - Ereignisse - vor - Prepare - Ind\}$ und $\{D - Ereignisse - nach - Prepare - Ind\}$ kausale Unabhängigkeit angenommen. Während der logischen Zeiträume d_1 und d_3 findet die Verarbeitung aller Kommunikationsereignisse eines Datentransferintervalls mit Ausnahme der Bearbeitung der TP-Prepare-RI-PDU statt. Diese Kommunikationsereignisse sind in der Intervall-Ereignismenge nicht explizit aufgeführt. Außerdem sind Wartezeiten des Kommunikationsmanagers während der lokalen Aktivitäten der Subtransaktion enthalten.

In einem Wurzelsubsystem fehlt das Eintreffen einer TP-Prepare-RI-PDU. Der Kommunikationsmanager erkennt nur den Anfang des Datentransferintervalls. Das Anfangsereignis hängt im Unterschied zu der oben angegebenen Struktur nur von der lokalen Subtransaktion ab. Die Aktivitätskomposition reduziert sich zu:

$[JJ]anf, C!anf[_{w,f}]$

\langle $(Danf(w), d_1)_w,$
 $\{ \forall e \in \{D - Ereignisse - nach - TP - Prepare - Ind\} : (e, d_5) \}$
 \rangle :

In das Datentransferintervall können zusätzliche Aktivitäten eingeführt werden, die parallel zu den aufgelisteten Aktivitäten eine lokale Zeitüberwachung realisieren. Solche Aktivitäten werden der angegebenen Struktur durch parallele Komposition überlagert.

Das Commit-Phase-1-Intervall stellt sich für einen lokalen Kommunikationsmanager als Sequenz von lokalen Aktivitäten dar:

[Clan/, C2anf]p,-

```

\ ( Queuing(p,,d3)d,,
  ( TP - Commit - Req(p;,<4)d,,
    { { Vp, E next(Pi), : ( Int - Prepare - Req(p,,d1)p,,
      { p, ;! early(pi) (Warten(p,,d5)p,,
        (Queuing(p,,d3)p,,
        (Empfang - Ready - RI - PDU(p,,d5)p,,
      }
    }
    { Vp E early(Pi), : ( (Warten(p,,d5)p,,
      { p, ;! ready(p,) (Queuing(p,,d5)p,,
        (Empfang - Ready - RI - PDU(p,,d5)p,,
      }
    }
  },
}
( Int - Ready - Req(prev(p;),d10)v,,
( Warten(prev(p,,),dn)p,,
( Queuing(prev(p;), d3)p,,
( Empfang - Commit - RI - PDU(prev(p;),d12)p,,
);

```

Für den Kommunikationsmanager des Wurzelsystems verkürzt sich die Aktivitätsliste zu:

[illegible]

87

[C2an/, C2end[r,1->

```

( { ( (Warten(p;),d13)p,,
    (Queuing(pi), d3)p,,
    (TP - Done - Req(p;), du)p,,
  ) ,
  { VpX E next(p;): ( (Int - Commit - Req(px), d1s),
                      (Warten(px), dl6)p,,
                      (Queuing(p;), d3)r,,
                      (Empfang - Commit - RC - PDU(p_x), d17)p
                    )
  }
} ,
{ (Int - Commit - Rsp(prev(p;)),d18),r,
}
>;

```

Für den Kommunikationsmanager eines Wurzelsystems verkürzt sich die angegebene Komposition um die Aktivität: (Int - Commit - Rsp,d18).

6.2.2 Die Überlagerung von Warteaktivitäten eines Kommunikationsmanagers

Die bereits behandelten internen Aktivitätskompositionen eines Kommunikationsmanagers können von externen Aktivitätskompositionen überlagert werden. Die Überlagerung betrifft die Warteaktivitäten eines Kommunikationsmanagers. Jede Warteaktivität ist von genau einem benachbarten Subsystem abhängig. Deshalb wird jede Warteaktivität durch eine verbindungsspezifische Parallelität charakterisiert. Die Nebenläufigkeit mehrerer Warteaktivitäten wurde bereits im vorangehenden Kapitel erfaßt.

Für das Datentransferintervall wird auf die Aufzählung aller möglichen Überlagerungen mit externen Aktivitätskompositionen verzichtet. Der Einfluß anwendungsspezifischen Kommunikationsverhaltens macht eine allgemeingültige Beschreibung der Überlagerung unmöglich. Berücksichtigt werden nur die Aktivitäten eines Datentransferintervalls, die Aktivitäten des Commit-Phase-1-Intervalls eines benachbarten Subsystems überlagern.

Warten - auf - Ready - RI - PDU_p, :

In die Warteaktivität eines Subsystems p auf das Eintreffen der Entscheidung ($Warten(preu(p)); d_{11}$) können aufgrund der nachfolgend beschriebenen Parallelität Aktivitäten des gesamten Transaktionsbereiches mit Ausnahme des unterlagerten Teilbaumes einfließen.

Warten - auf - Commit - RI - PDU_B :

```

{ (Warten(prev(p;)) , dl )p ,
  ( (Senden - Ready - RI - PDU(p; , prev(p;)) , d19)k ,
    (Queuing(prev(p;)) , d8)in v(p) ,
    (Empfang - Rndy - RI - PDU(prev(p;)) , d9)pm (p, l
    [ { (e, d2) ! e ∈ E R C {D - nach - TP - Prepare - Ind} }r, m (p) ,
      (TP - Commit - Req(prev(p;)) , d9)pm (p) ,
      (Queuing(p, ev(p;)) , d3)p, a(p) ,

      3 Af C next(prev(p;)) ,
      { M n ready(prev(pi)) = 0 : { (Warten(py) , d8)prev(pi) ,
        ∀ py E Af
          (Queuing(py) , d3)prev(pi) ,
          (Empfang - Ready - RI - PDU(py) , d9)pm i r, l
        )
      }
    !
  }
  [
    '∀ py E next(prev(p;)) ,
    { By 4 ready(prev(pi)) : { (Warten(py) , d8)prev(pi) ,
      P, i M
        (Queuing(py) , d3)p, a(p) ,
        (Empfang - Ready - RI - PDU(py) , d9)pm (p)
      )
    }
    (Int - Ready - Req(prev(p;)) , d10)p, a(p) ,
    Warten - auf - Commit - RI - PDUp , a(p) ,
    (Queuing(prev(prev(p;))) , d8)p, a(p), a(p) ,
    (Empfang - Commit - RI - PDU(prev(prev(p;)) , d12)p, a(p) ,
    { { ∀ py 'f p E n.: t(prev(p;)) : (Int - Commit - Req(py) , d, s)pm (p; d :
      ( (Int - Commit - Req(p; , dis)p, a(p) ,
        (Senden - Commit - RI - PDU(prev(p;)) , p; , d19)k
      )
    }
  }
}

```

Zuletzt ergibt sich eine Parallelität bei der lokalen Bekanntgabe der Entscheidung gegenüber der Weiterleitung der Entscheidung in den unterlagerten Teilbäumen eines Subsystems. Die Warteaktivität auf die Bestätigung der lokalen Subtransaktion des Transaktionsprozesses P_i wird dem internen Verhalten des Subsystems zugeordnet und ist nicht weiter zerlegbar. Die in der folgenden Formel aufgelöste Warteaktivität bezieht sich auf eine Transaktionsverbindung zu einem Nachfolgerprozeß.

Warten - auf - Commit - RC - PDU_{p_x, P_x} E next(p), :

$$\begin{aligned} & \{ \text{ (Warten}_{(p_x), d, e}_{p_i}, \\ & \quad \langle \text{ (Senden - Commit - RI - } PDU_{(p_i, P_x), d_{ig}} \rangle k, \\ & \quad \text{ (Queuing}_{(p_i), d_3}_{p_i}, \\ & \quad \text{ (Emfang - Commit - RI - } PDU_{(p_i), d_{12}} \rangle \rangle \\ & \quad \{ (\text{ (Warten}_{(p_x), d_{13}} \rangle_{p_x} > \\ & \quad \quad \text{ (Queuing}_{(p_i), d_3}_{p_i}, \\ & \quad \quad \text{ (TP - Done - Req}_{(p_x), d_{14}} \rangle_{p_i}, \\ & \quad) \text{ ' } \\ & \quad \{ \text{ v}_{p_y} \text{ E next}_{(p_x)} : (\text{ Warten - auf - Commit - RC - } PDU_v, , \\ & \quad \quad \text{ (Queuing}_{(p_i), d_3}_{p_i}, \\ & \quad \quad \text{ (Emfang - Commit - RC - } PDU_{(p_y), d_{17}} \rangle r, \\ & \quad) \\ & \quad \} \\ & \quad \} \text{ ' } \\ & \quad \{ (\text{ (Int - Commit - Rs}_{p_i}, d_{is} \rangle_{v_x}, \\ & \quad \quad \text{ (Senden - Commit - RC - } PDU_{(p_x, p_i), d_{19}} \rangle k \\ & \quad) \\ & \quad \} \\ & \quad \} \\ & \} \end{aligned}$$

Alle drei Formeln besitzen einen rekursiven Anteil. Die Terminierung der Rekursionen ist durch die endliche Anzahl von Transaktionsprozessen in einem Transaktionsbereich gewährleistet.

6.3 Interpretation von Aktivitäten für Monoprozessor-Subsysteme

Die Abbildung der logischen Zeiträume der komponierten Aktivitäten auf eine lokale Uhr eines Subsystems ermöglicht dem Kommunikationsmanager eine Auswertung des lokalen Zeitbedarfs für Bearbeitungsabschnitte, die sich aus einer Menge vollständig bearbeiteter Aktivitäten zusammensetzen. Er kann für solche Bearbeitungsabschnitte interne Zeitanforderungen, die bezüglich der lokalen Uhr festgelegt wurden, überwachen und protokollieren. Die Reaktionsfreiräume für die Zeitüberwachung wurden bereits in Kapitel 5.3 an der Intervallstruktur aufgezeigt. Für Bearbeitungsabschnitte, die vollständig in einem Intervall enthalten sind, gelten die für das ganze Intervall definierten Reaktionsmöglichkeiten.

In einem Mehrprozessor-Subsystem müssen zur Abschätzung des Zeitverhaltens die Zuteilungsstrategien von Prozessoren zu den Aktivitäten berücksichtigt werden. Ausgangspunkt für die nachfolgend aufgezeigte Abbildung der logischen Zeiträume auf die lokale Uhr ist ein Monoprozessor-Subsystem. Diese Beschränkung führt zur sequentiellen Bearbeitung aller internen Aktivitäten. Das Zeitverhalten hängt dann innerhalb eines Transaktionsprozesses nur von der Reihenfolge der Aktivitätsbearbeitung ab.

6.3.1 Lokale Abschätzung der Dauer interner Aktivitätskompositionen

Die parallelen Kompositionen interner Aktivitäten eines Kommunikationsmanagers werden in einem Monoprozessor-Subsystem quasiparallel bearbeitet. Die dabei gewählte Reihenfolge kann den lokalen Startzeitpunkt externer Aktivitätskompositionen beeinflussen. Warteaktivitäten werden von dieser Sequentialisierungsregel ausgenommen. Warten ist als eine Aktivität der Untätigkeit zu interpretieren. Während einer Warteaktivität kann eine andere Aktivität bearbeitet werden. Auf die sequentiellen Verknüpfungen interner Aktivitäten hat die Einschränkung auf ein Monoprozessorsystem keinen Einfluß.

Für die Abbildung der logischen Zeiträume auf lokale Zeiträume wird für

jeden Transaktionsprozeß p_i eine partielle Funktion über der Menge der Aktivitäten der an der Transaktion beteiligten Kommunikationsmanager definiert:

$$F_{p_i}((a, d_n)_{p_i}) = \begin{cases} d_i & \text{falls } p_i = p_j \\ \text{nicht def.} & \text{sonst} \end{cases}$$

Die Menge der Aktivitäten $(a, d_n)_{p_i}$ ist für jeden Transaktionsprozeß ist aus Tabelle 3 auf Seite 84 zu entnehmen. Der Index p_i einer Aktivität bezeichnet den ausführenden Transaktionsprozeß.

Die nachfolgenden Berechnungen gehen von einem deterministischen Verhalten eines Kommunikationsmanagers aus, welches unabhängig von den bearbeiteten Verbindungen ist. Folgendes Verhalten wird für einen deterministischen Kommunikationsmanager gefordert:

- Die Aktivitäten eines Startereignisses der TP-Schnittstelle und aller darauffolgenden internen Ereignisse werden zusammenhängend bearbeitet.
- Die Aktivitäten eines Empfangsereignisses und aller darauffolgenden internen Ereignisse werden zusammenhängend bearbeitet.
- Der lokale Zeitbedarf d_i zur Verarbeitung jeder empfangenen PDU ist konstant, unabhängig von der Verbindung.
- Die lokale Bearbeitungszeit d_i zur Verarbeitung eines internen Dienstprimitivs ist konstant, unabhängig von der Verbindung.
- Die lokale Dauer zur Verarbeitung eines TP-Request- oder -Response-Primitivs ist konstant, unabhängig von der Verbindung.

Im Kommunikationsmanager ergibt sich für die Aktivitätskomposition des Datentransferintervalls eines Transaktionsprozesses ohne Wurzelfunktionalität als Maximum des lokalen Zeitbedarfs die Summe:

$$\forall p \neq w : \max(DDauer) = \sum d'_2 + d'_3 + d'_4 + \sum d'_5;$$

Die Mächtigkeit der Menge **D - Ereignisse - vor - TP - Prepare - Ind** sowie die Anzahl der Ereignisse der Menge **D - Ereignisse - nach - TP -**

Prepare-Ind sind nicht bekannt. Deshalb wurde hier und im folgenden auf die Angabe der Grenzen der Summen über d_i und d verzichtet.

Die lokale Bearbeitungsdauer für das Anfangsereignis des Datentransferintervalls der Wurzel d und der Prepare-R1-PDU d_i ist unter den oben angegebenen Voraussetzungen durch Konstanten des Kommunikationsmanagers bestimmt. Die Zeiträume I_i ; d_i und I ; d hängen von dem anwendungsspezifischen Kommunikationsverhalten und dem anwendungsinternen Zeitbedarf ab.

Für die Aktivitätskomposition des Datentransferintervalls eines Wurzelprozesses ergibt sich als Maximum des lokalen Zeitbedarfs die Summe:

$$\text{Wurzel: } \max(DDauer) = d_i + L d ;$$

Die untere Schranke für die lokale Dauer des Datentransferintervalls ergibt sich aus der Annahme, daß weder anwendungsspezifische Kommunikationsaktivitäten noch Queuing auftreten:

$$\sum d'_2 = 0, d'_3 = 0, \sum d'_5 = 0$$

Für einen Transaktionsprozeß ohne Wurzelfunktionalität ergibt sich:

$$\forall p \neq w: \min(DDauer) = d_i$$

Für den Wurzelprozeß ergibt sich als Minimum:

$$\text{Wurzel: } \min(DDauer) = d_i;$$

Der notwendige Rahmen zur Begrenzung des Datentransferintervalls kann nur von der Subtransaktion bereitgestellt werden. Die Vorgabe einer oberen Schranke durch den Kommunikationsmanager erscheint nicht sinnvoll.

Die obere Schranke für das Commit-Phase-1-Intervall ergibt sich aus den folgenden Annahmen:

- Der TP-Commit-Req wird durch Queuing verzögert.
- Die Commit-RI-PDU wird durch Queuing verzögert.

- o Alle Ready-RI-PDUs treffen gleichzeitig ein.
- Während der Datentransferphase wurde die vorzeitige Commit-Vorbereitung für keinen Nachfolger eingeleitet.
- Alle Ready-RI-PDUs werden durch Queuing verzögert.

Die Indizes π der Nachfolgerprozesse $p_x \in \text{next}(p)$ orientieren sich für diese Berechnung an der Reihenfolge der Int-Prepare-Bearbeitung. Unter der Voraussetzung konstanter verbindungsunabhängiger Zeiten für d , d , d , d_0 und d_2 ergibt sich daraus folgende Berechnungsformel:

$$\begin{aligned} \text{mal}(\text{C1Dauer}) = & d; + d + \text{I}(\text{next}(p), 1 - (d + d) + L \cdot \sum_{i=1}^n (p_i) / d; (P_x) + \\ & \text{maa} \cdot \{x \in \{1, \dots, |\text{next}(p)|\} : d; (P_x) - L \cdot \sum_{i=1}^n d q + \\ & , d_0 + d;_1 + d; + d;_2; \end{aligned}$$

Das Minimum für die Dauer eines Commit-Phase-1-IntPrvalls ergibt sich aus den Annahmen, daß

- auf keiner Verbindung zum Vorgänger oder zu einem Nachfolger eine Verzögerung durch Queuing eintritt;
- keine lokalen Wartezeiten entstehen; d.h. während jeder Warteaktivität auf eine Ready-RI-PDU wird immer eine der Aktivitäten (*Int* - Prepare-Req(p_x), d_7) $_p$, oder (Empfang-Ready-RI-PDU(p_x), d_8) $_p$, für einen anderen Nachfolger ausgeführt.
- die Warteaktivität vor dem Empfang der Commit-RI-PDU den Wert 0 besitzt: $d_{11} = 0$
- " bereits während der Datentransferphase die Commit-Vorbereitung für alle Nachfolger abgeschlossen wurde.
 $\text{next}(p_i) = \text{early}(p_i) = \text{ready}(p_i)$

Es ergibt sich als minimale Bearbeitungszeit des Commit-Phase-1-Intervalls:

$$\forall p \neq w : \min(\text{C1Dauer}) = d'_6 + d'_{10} + d'_{12};$$

In der Wurzel entfallen außerdem die lokalen Zeiträume d_0 und d_2 .

Das Maximum der lokalen Zeitdauer für das Commit-Phase-2-Intervall ergibt sich aus:

$$\begin{aligned} \max(C2Dauer) = & \lfloor \text{next}(p_i) \rfloor \cdot (d_{i5} + d_{i1}) + L_{n_i} \cdot t(p_i) \cdot d_{i5} + \\ & \max \{ \quad f i x E \{1, \dots, \text{Ine.Tt}(p_i) \cdot I\} : d \setminus i P_x \} - (x - 1) \cdot d_{i5} \} \\ & \cup \{ d_{i3} + d_{i7} + d_{i4} \} \} \\ & + d_{i8}' \end{aligned}$$

Entsprechend dem Commit-Phase-1-Intervall ergibt sich das Minimum für das Commit-Phase-2-Intervall aus der Annahme, daß alle Wartezeiten von den lokalen Aktivitäten (TP - Done - Req, d_{i4})_p, (Int - Commit - Req(\bar{p}_i), d_{i5})_p, und (Empfang - Commit - RC - PDU(p_x), d_{i7})_p, vollständig überlagert werden.

$$\min(C2Dauer) = d_{i4} + d_{i5} + \lfloor \text{next}(p_i) \rfloor \cdot (d_{i5} + d_{i7})$$

Für das Wurzelsubsystem besitzt d_{i8} den Wert 0.

6.3.2 Einfluß der Übertragungszeiten auf die Wartezeiten

Alle Warteaktivitäten eines Kommunikationsmanagers mit Ausnahme der Warteaktivität auf den TP-Done-Req von der lokalen Subtransaktion sind verbindungspezifisch. An den Aktivitätskompositionen in Kapitel 6.2.2 zeigt sich, daß die Überlagerung dieser Warteaktivitäten mit je einer Sendeaktivität des Kommunikationssystems beginnt und endet. Jeder Übertragungsaktivität wurde die Zeitdauer d_{i9} zugeordnet.

Mit der Aktivität (Warten(p_x), d_{i8})_p ist die Bearbeitungszeit eines Nachfolgers $P_x \in \text{next}(p_i)$ vom Empfang der Aufforderung zur Commit-Vorbereitung bis zum Beginn der Wartezeit auf die Entscheidung eingrenzbar. Die Abschätzung wird von den beiden Aktivitäten Senden - Prepare - RI - PDU(P_x , P_x), d_{i9})_k und (Senden - Ready - RI - PDU(p_x , P), d_{i9})_k beeinflusst. Das Intervall [Empfang - Prepare - RI - PDU, Senden - Ready - **RI - PDU**]_p klammert alle Aktivitäten des Kommunikationsmanagers für den Prozeß P_x zwischen den beiden Übertragungen ein. Die Dauer dieses Vorbereitungsintervalls von P_x bezüglich der lokalen Uhr von Prozeß p ergibt sich aus:

$$D' \left(\begin{array}{l} \text{Empfang - Prepare - RI - PDU,} \\ \text{Senden - Ready - RI - PDU} \end{array} , P \right) := d_s' - d_k' - d_{19}',$$

Entsprechend gilt für die während der Aktivität $(Warten(p_x), d_{16})_v$ verstreichende Bearbeitungszeit eines Nachfolgerprozesses $P_x \in next(p)$:

$$D' \left(\begin{array}{l} \text{Empfang - Commit - RI - PDU,} \\ \text{Senden - Commit - RC - PDU} \end{array} , P \right) := d_{16}' - d_{19}' - d_{19}',$$

Mit d_9 werden hier die beiden Übertragungen $(Senden - Commit - RI - PDU(P, P_x), d_{19})_k$ und $(Senden - Commit - RC - PDU(p_x, P), d_{19})_k$ bezüglich der Uhr von Transaktionsprozeß p bewertet.

Aus der Warteaktivität $(Warten(prev(p)), du)_v$ läßt sich eine Bearbeitungszeit des Vorgängerprozesses vom Eintreffen der *Ready - RI - PDU* beim Vorgänger bis zum Weiterleiten der *Commit - RI - PDU* an Prozeß p ermitteln:

$$D' \left(\begin{array}{l} \text{Empfang - Ready - RI - PDU,} \\ \text{Senden - Commit - RI - PDU} \end{array} , P \right) := du - d_s' - d_{19}'$$

Mit d_9 werden hier die beiden Übertragungen $(Senden - Commit - RI - PDU(P, P_x), d_m)_k$ und $(Senden - Commit - RC - PDU(p_x, P), d_m)_k$ bezüglich der Uhr von Transaktionsprozeß p erfaßt.

Für das Übertragungsverhalten wurden weder verschiedene Verbindungen noch verschiedene Übertragungsrichtungen auf einer Verbindung berücksichtigt. Neben diesen beiden Faktoren können sich weitere Einflüsse z.B. aus der Netzlast ergeben. Die daraus resultierende Unsicherheit der Übertragungszeit kann in einem Transaktionsprozeß nur dann abgeschätzt werden, wenn für die Übertragung jeder PDU die zugeordnete Dauer d_9 bezüglich der lokalen Uhr ein Maximalwert D_{max}' und ein Minimalwert D_{min}' bekannt ist. Wenn vom Kommunikationssystem keine Angaben über die minimale Übertragungsdauer vorliegen, gehen diese mit dem Wert 0 in die Abschätzung ein. Aus den drei Gleichungen ergeben sich unter Einbeziehung von maximalen und minimalen Übertragungszeiten 6 Abschätzungen:

$$Dmin'([\begin{array}{l} \text{Empfang-Prepare-RI-PDU,} \\ \text{Senden-Ready-RI-PDU} \end{array}]_{p_x}, p) := d'_8 - dmax'_{19} - dmax'_{19};$$

$$Dmax'([\begin{array}{l} \text{Empfang-Prepare-RI-PDU,} \\ \text{Senden-Ready-RI-PDU} \end{array}]_{p_x}, p) := d'_8 - dmin'_{19} - dmin'_{19};$$

$$Dmin'([\begin{array}{l} \text{Empfang-Commit-RI-PDU,} \\ \text{Senden-Commit-RC-PDU} \end{array}]_{p_x}, p) := d'_{16} - dmax'_{19} - dmax'_{19};$$

$$Dmax'([\begin{array}{l} \text{Empfang-Commit-RI-PDU,} \\ \text{Senden-Commit-RC-PDU} \end{array}]_{p_x}, p) := d'_{16} - dmin'_{19} - dmin'_{19};$$

$$Dmin'([\begin{array}{l} \text{Empfang-Ready-RI-PDU,} \\ \text{Senden-Commit-RI-PDU} \end{array}]_{p_x}, p) := d'_{11} - dmax'_{19} - dmax'_{19};$$

$$Dmax'([\begin{array}{l} \text{Empfang-Ready-RI-PDU,} \\ \text{Senden-Commit-RI-PDU} \end{array}]_{p_x}, p) := d'_{11} - dmin'_{19} - dmin'_{19};$$

Für die Abgrenzung der Bearbeitungszeit eines Partnerprozesses während der Warteaktivität eines Prozesses p ist nur die Summe der Übertragungsdauer der beiden PDUs interessant. Für die Einordnung des Partnerintervalls in die kontinuierliche Zeitachse von p ist die Kenntnis von $dmax_p$ und $dmin_p$ für jede PDU notwendig. Die Abschätzung kann erst nach Ablauf der jeweiligen Wartezeit erfolgen.

7 Grenzen der Echtzeitfähigkeit von ISO-TP

Für eine verteilte Transaktion stellen sich die Grenzen der Echtzeitfähigkeit von ISO-TP an der TP-Schnittstelle dar. Die in Kapitel 6 analysierten Aktivitäten der Kommunikationsmanager und des Kommunikationssystems bleiben der Anwendung verborgen. In diesem Kapitel werden Zeitanforderungen der Subtransaktionen einer verteilten Transaktion auf ihre Realisierbarkeit unter Einsatz von ISO-TP als Kommunikationsgrundlage überprüft. Die sich ergebenden Grenzen lassen sich zum Teil auf einzelne Aktivitäten der Kommunikationsmanager zurückführen.

Zeitanforderungen einer verteilten Anwendung bestehen für beliebige Ereignisse, die während des Ablaufs der verteilten Prozesse eintreten können. Grundlage der Definition einer Zeitanforderung ist die Festlegung einer Bezugsuhr. Bei verteilter Verarbeitung ist die Festlegung einer globalen Uhr als Basis für Zeitanforderungen im allgemeinen nicht möglich. Die Menge der möglichen Zeitanforderungen wird durch die folgende Definition eingeschränkt:

Eine Zeitanforderung wird als Relation über den Eintrittszeitpunkten der Ereignisse eines Transaktionsprozesses bezüglich einer lokalen Uhr definiert. Anforderungen an einen Eintrittszeitpunkt können durch absolute Zeitwerte oder durch Zeitabstände zu anderen Ereignissen formuliert werden. Für jedes Ereignis ist eine relative Zeitangabe unter Bezug auf den Eintrittszeitpunkt des Anfangsereignisses der Datentransferphase möglich. Relationen über Paaren von Zeitpunkten erlauben die Angabe einer lokalen Zeitdauer als Zeitrahmenbedingung. Zeitvorgaben für Paare von Ereignissen, die verschiedenen Anwendungsprozessen angehören, werden nicht behandelt.

Die Grenzen der Echtzeitfähigkeit des Standards drücken sich in der mangelnden Bereitstellung und Auswertung von zeitbezogenen Daten aus. Neben der

- Überwachung von kausalen Zeiträumen unter Bezug auf eine definierte Uhr ist die
- Begrenzbarkeit der zeitüberwachten Aktivitäten durch definierte Reaktionsmöglichkeiten

wesentlich für die in der Echtzeitverarbeitung geforderte Vorhersehbarkeit des Zeitverhaltens. Zunächst werden Zeitbedingungen behandelt, die innerhalb eines Subsystems vollständig bearbeitet werden können. Im Anschluß daran werden subsys_temübergreifende Zeitanforderungen betrachtet, deren Behandlung durch Protokollelemente unterstützt werden muß.

7.1 Berücksichtigung von Zeitanforderungen an der TP-Schnittstelle

Die TP-Schnittstelle eines Subsystems wird durch die Aktionen einer Subtransaktion und Aktivitäten des Kommunikationsmanagers zur Realisierung einer TP-Protokollmaschine charakterisiert. Die Kontrolle von Zeitbedingungen innerhalb eines Transaktionsprozesses und die Möglichkeiten zur Reaktion auf deren Verletzung werden zunächst durch drei Beispiele erläutert.

Beispiel 1: Eine Subtransaktion eines Transaktionsbereiches setzt während ihrer Datentransferphase eine Anfrage an eine Partnersubtransaktion ab und erwartet die Antwort innerhalb eines bezüglich der lokalen Uhr festgelegten Zeitraumes.

Das Warten auf die Antwort findet innerhalb der Datentransferphase statt und kann durch Zeitüberwachung der auftragserteilenden Subtransaktion begrenzt werden. Nach Ablauf der Frist kann die Subtransaktion z.B. den Abbruch der Transaktion einleiten. Die Überschreitung des festgesetzten Zeitraumes kann auf folgende lokale Ursachen zurückzuführen sein:

- Konkatenationsbedingte Pufferung der Anfrage im lokalen Kommunikationsmanager
- Konkurrenzabhängiges Queuing der Anfrage oder der Antwort im lokalen Kommunikationsmanager

- Unterschätzung des Zeitbedarfs für die lokale Bearbeitung der Anfrage oder der Antwort im lokalen Kommunikationsmanager

Beispiel 2 Eine Subtransaktion setzt an mehrere Nachfolger-Subtransaktionen Aufträge ab, die Veränderungen von Transaktionsobjekten ohne Ergebnis¹⁴ bewirken. Als kausal begrenzter Zeitraum für die Auftragsbearbeitung wird die Commit-Phase-2 abgesprochen¹⁵. Damit wird eine synchrone Abwicklung aller Aufträge angestrebt.

Der Anfangszeitpunkt des Commit-Phase-2-Intervalls ist nur im Kommunikationsmanager sichtbar. Die Zeitüberwachung der Commit-Phase-2 kann deshalb in der auftragserteilenden Subtransaktion nicht unmittelbar durchgeführt werden. Als kleinster Rahmen, der die Commit-Phase-2-Intervalle aller Nachfolger enthält, bietet sich nur der kausal begrenzte Zeitraum zwischen TP-Commit-Req und TP-Commit-Complete-Ind an. Damit wird die Phase 1 in die Zeitüberwachung einbezogen und führt zu einer Verzerrung der Aussage über die Dauer der synchronen Auftragsabwicklung. Die Überschreitung der angestrebten Frist kann nur lokale Konsequenzen auslösen. Eine Reaktion mit konsistenter prozeßübergreifender Wirkung ist nicht definiert.

Die beiden Beispiele zeigen, daß das lokale Zeitverhalten eines Kommunikationsmanagers für eine Subtransaktion nicht vorhersehbar ist. Diese Aussage gliedert sich in 9 Punkte:

1. Die Pufferung einer PDU im Sendepuffer einer Verbindung ist für eine Subtransaktion nicht kontrollierbar.
2. Die Pufferung einer PDU ist durch die Subtransaktion nur durch nachfolgende Dienstaufrufe begrenzt, die nur als letztes Element einer Konkatena-tionssequenz auftreten können bzw. deren Konkatena-tion nicht zulässig ist.

4 Update-Transactions

15 Der Ansatz zur Bearbeitung von Update-Aufträgen als Field Call [GR93] vermeidet aufwendige Undo-Operationen für den Fall, daß die Transaktion zurückgesetzt werden muß. Die Auftragsausführung beginnt erst, wenn die Entscheidung zum positiven Transaktionsabschluß bereits bekannt ist.

3. Das Queuing eines Request- oder Response-Primitives im lokalen Kommunikationsmanager ist in der Subtransaktion nicht kontrollierbar.
4. Das Queuing eines Request- oder Response-Primitives im lokalen Kommunikationsmanager ist durch die Subtransaktion nicht begrenzbear.
5. Das Queuing einer vom lokalen Kommunikationsmanager empfangenen PDU ist in der Subtransaktion nicht kontrollierbar.
6. Das Queuing einer vom lokalen Kommunikationsmanager empfangenen PDU ist durch die Subtransaktion nicht begrenzbear.
7. Die Dauer von Aktivitäten eines Kommunikationsmanagers zur Bearbeitung von TP-Dienstprimitiven, internen Dienstprimitiven und PDUs ist in der Subtransaktion nicht kontrollierbar.
8. Die Warteaktivitäten des Kommunikationsmanagers auf die Commit-RC-PDUs der Nachfolgerprozesse sind in S_b Subtransaktion nicht kontrollierbar.
9. Die Warteaktivitäten des Kommunikationsmanagers während der beiden Commit-Phasen sind in der Subtransaktion nicht begrenzbear.

Aus der Sicht des Kommunikationsmanagers stellt sich die TP-Schnittstelle wie folgt dar:

1. Anforderungen einer Subtransaktion bezüglich der maximalen Pufferung einer PDU im Kommunikationsmanager sind nicht verfügbar.
2. Anforderungen einer Subtransaktion bezüglich des maximalen Queuing eines Request- oder Response-Primitives sind im Kommunikationsmanager nicht verfügbar.
3. Anforderungen einer Subtransaktion bezüglich des maximalen Queuing einer PDU im Kommunikationsmanager sind nicht verfügbar.
4. Anforderungen der Subtransaktionen bezüglich der Dauer von Warteaktivitäten sind dem Kommunikationsmanager nicht bekannt.

Aus diesen beiden Blickrichtungen wird deutlich, daß weder die TP-Protokollmaschine noch die Subtransaktion zeitbezogene Daten bereitstellt, die eine Gegenüberstellung der Zeitanforderungen mit der lokal verursachten Zeitverzögerung erlauben.

In der Definition der TP-Schnittstelle sind nur solche Dienstprimitive und Parameter vorgesehen, die die Abwicklung des TP-Protokolls unterstützen. Damit wird eine minimale Funktionalität festgelegt. Die Implementierung einer Obermenge der vorgegebenen minimalen Funktionalität auf einem Subsystem bildet keinen Widerspruch zum TP-Standard.

7.2 Berücksichtigung von Zeitanforderungen im TP-Protokoll

Die Vorhersehbarkeit des Zeitverhaltens von Prozessen eines Transaktionsbereiches in einem Transaktionsprozeß setzt die Übermittlung von zeitbezogenen Daten und deren Bewertung relativ zum lokalen Zeitbegriff voraus. Der Informationsfluß zwischen den Transaktionsprozessen wird durch Übertragungsprotokolle geregelt. Das ISO-Referenzmodell beschreibt eine Verschachtelung von Übertragungsprotokollen, die sich in ISO-TP durch die offene Architektur einer TP-Protokollmaschine fortsetzen läßt. Die Auswahl von anwendungsspezifischen Kommunikationsdiensten für die Datentransferphase auf einer Transaktionsverbindung bewirkt die Einbettung der zugehörigen anwendungsspezifischen Protokolle in die TP-Protokollmaschinen der beiden Verbindungspartnerprozesse. Unabhängig von dieser verschachtelten Protokollhierarchie ist die zum ISO-Protokollturm nebenläufige Abwicklung beliebiger Protokolle denkbar. Als Beispiel können sich die Transaktionsprozesse an einem Synchronisationsprotokoll für lokale Uhren beteiligen. Sie sind in diesem Zusammenhang nicht an die in ISO-TP vorgegebene Baumstruktur als Basis für die Protokollabwicklung gebunden.

Aus dieser Sicht wird nachfolgend die durch den Einsatz des TP-Protokolls in einem Transaktionsprozeß zu gewinnende Information über das globale Zeitverhalten betrachtet. Die hier betrachteten zeitlich relevanten Abläufe sind unabhängig von dem im vorangehenden Kapitel aufgeführten Mängel der Kommunikationsschnittstelle. Ein Transaktionsprozeß vereinigt die zeit-

bezogenen Daten der Subtransaktion und der TP-Protokollmaschine in sich. Unter Bezug auf den Begriff des Transaktionsprozesses wird die Abstraktion von den im vorangehenden Abschnitt behandelten Mängeln des Informationsflusses an der TP-Schnittstelle erreicht. Damit können alle Informationen, die in einem Transaktionsprozeß, d.h. in der Subtransaktion oder in der TP-Protokollmaschine, vorliegen, im TP-Protokoll-Fluß verwertet werden.

Mit dem Einsatz von ISO-TP in einer verteilten Transaktion wird für jeden Transaktionsprozeß ein Kommunikationsrahmen festgelegt. Prozeßübergreifende Zeitanforderungen können für den gesamten Rahmen oder für Ausschnitte definiert sein. Die räumliche Reichweite eines durch zwei Ereignisse vorgegebenen Rahmens ist durch die darin ablaufenden Protokolle bestimmt. Aus der räumlichen Reichweite ergeben sich Abhängigkeiten des überwachten Zeitrahmens gegenüber dem Zeitverhalten des Kommunikationssystems und allen in die Protokollabwicklung einbezogenen Transaktionsprozessen. Ohne Berücksichtigung des Informationsflusses der aus anwendungsspezifischen Diensten resultierenden Protokollabläufe stellen sich das räumliche und das zeitbezogene Wissen eines Transaktionsprozesses wie folgt dar:

- Räumliches Wissen:

- Jeder Transaktionsprozeß p_i kennt seinen Vorgänger-Transaktionsprozeß $prev(p_i)$.
- Jeder Transaktionsprozeß p_i kennt alle direkten Nachfolger $next(p_i)$ im Transaktionsbereich.
- Die Position eines Transaktionsprozesses in einem Transaktionsbereich relativ zur Transaktionswurzel ist während der ganzen Transaktion konstant.
- Die Position eines Transaktionsprozesses relativ zu den Transaktionsprozessen des unterlagerten Teilbaumes ist nach dem Eintritt in die Commit-Phase-1 und dem Empfang aller Ready-RI-PDUs konstant.
- Die Position eines Transaktionsprozesses im Transaktionsbereich ist nach dem Empfang der Commit-RI-PDU konstant.

- Zeitliches Wissen:

- Jeder Transaktionsprozeß weiß mit dem Empfangszeitpunkt der letzten Ready-RI-PDU, daß alle Transaktionsprozesse des unterlagerten Teilbaumes in die Commit-Phase-1 eingetreten sind.
- Jeder Transaktionsprozeß weiß zum Zeitpunkt des Empfangs der Commit-RI-PDU, daß der direkte Vorgänger und alle ihm übergeordneten Prozesse bereits in die Commit-Phase-2 eingetreten sind und daß alle Prozesse des Transaktionsbereiches mindestens die Commit-Phase-1 erreicht haben.
- Jeder Transaktionsprozeß weiß zum Zeitpunkt des Empfangs der Commit-RC-PDU, daß der hierarchisch unterlagerte Teilbaum die Kenntnis der Entscheidung zum positiven Transaktionsabschluß bestätigt und die Kommunikation mit anderen Transaktionsprozessen für diese Transaktion beendet hat.
- Die Übermittlung zeitbezogener Daten wird vom TP-Protokoll nicht unterstützt. Kenntnisse über Zeitanforderungen eines Transaktionsprozesses sind daher in einem anderen Transaktionsprozeß nicht vorhanden.
- Der Zeitbegriff verschiedener Transaktionsprozesse ist im allgemeinen nicht synchronisiert. Die Berücksichtigung von Zeitanforderungen eines Transaktionsprozesses in einem anderen Transaktionsprozeß ist so nicht möglich.

Während der Datentransferphase können anwendungsspezifische Protokolle zum Einsatz kommen. Soweit diese Protokolle einen Transfer zeitbezogener Daten erlauben, läßt sich die übermittelte Information zur prozeßübergreifenden Zeitdatenverarbeitung einsetzen. Das räumliche Wissen eines Transaktionsprozesses ist nicht ausreichend für die Abgrenzung des räumlichen Einflßbereiches auf den definierten Zeitrahmen. Für die Erweiterung des räumlichen Wissens können anwendungsspezifische Protokolle ebenfalls hilfreich sein.

Mit dem Eintritt in die Phase 1 des Commit-Dienstes werden die Schnittstellen der integrierten Kommunikationsdienste unzugänglich. Während der beiden Commit-Phasen ist nur noch der von TP gebotene Informationsfluß definiert.

Das nachfolgende Beispiel zeigt eine Situation auf, in der die Auswertung der Zeitanforderungen eines Transaktionsprozesses in einem übergeordneten Transaktionsprozeß vorteilhaft wäre.

Beispiel 3 Ein Transaktionsprozeß hat den Auftrag, eine nicht umkehrbare Operation auf einem Transaktionsobjekt auszuführen. Die Zugriffszeit auf das Transaktionsobjekt unterliegt harten zeitlichen Bedingungen. Der Transaktionsprozeß wartet zunächst das Eintreffen der TP-Prepare-Ind ab und fordert danach seinen unterlagerten Teilbaum zur Commit-Vorbereitung auf. Erst nach dem Eintreffen der TP-Ready-Indication-Primitive aller Nachfolgerprozesse belegt er das zeitkritische Transaktionsobjekt. Unmittelbar danach setzt er den TP-Commit-Req ab und tritt damit in das Commit-Phase-1-Intervall ein. Für den Transaktionsprozeß besteht keine Möglichkeit zur Begrenzung des eigenen Commit-Phase-1-Intervalls. Wenn eine heuristische Entscheidung¹⁶ nicht möglich ist, kann nur das Eintreffen der globalen Entscheidung abgewartet werden.

Die beschriebene Blockierung für einen nicht beeinflussbaren Zeitraum könnte von einem übergeordneten Transaktionsprozeß, der sich noch in der Datentransferphase befindet, begrenzt werden. Dies setzt voraus, daß der übergeordnete Prozeß die vorliegende Zeitbedingung kennt und ihre Überschreitung feststellen kann. Die Einleitung des Transaktionsabbruchs wäre für den übergeordneten Transaktionsprozeß möglich. Die Abbruch-Meldung kann unmittelbar nach unten weitergeleitet werden.

¹⁶ Die im TP-Standard vorgesehene Möglichkeit der (lokalen) heuristischen Entscheidung läßt sich nur dann einsetzen, wenn die Operation entweder umkehrbar ist oder ohne Fehler auf einen späteren Zeitpunkt verschoben werden kann. Im ersten Fall entspricht die heuristische Entscheidung dem positiven Abschluß. Nach der vorgezogenen Ausführung der Operation wird das Transaktionsobjekt freigegeben. Das Eintreffen einer negativen globalen Entscheidung erfordert dann die erneute Belegung des Transaktionsobjektes zur Ausführung der Umkehroperation. Im zweiten Fall wird das Transaktionsobjekt unverändert freigegeben und beim Eintreffen einer positiven globalen Entscheidung erneut angefordert, um die Operation durchzuführen.

7.3 Grenzen des Commit-Verfahrens

Prinzipielle Grenzen der Behandlung von Zeitanforderungen in verteilten Transaktionen ergeben sich durch Ausfälle von Transaktionsprozessen und Transaktionsverbindungen. Die Garantie der ACID-Rules über solche Ausfallsituationen hinweg kann zu zeitlich nicht eingrenzbaren Blockierungen von Transaktionsprozessen führen. Die folgenden Beispiele zeigen unauflösbare Wartesituationen auf.

Beispiel 4 Ein Transaktionsprozeß setzt nach dem Eintreffen der TP-Ready-Ind aller Nachfolgerprozesse einen TP-Commit-Req ab. Dies führt zum Absenden einer Ready-RI-PDU an den Vorgängerprozeß. Der Transaktionsprozeß begibt sich damit in die Wartesituation auf die globale Entscheidung. Durch einen Ausfall der Transaktionsverbindung zum Vorgänger tritt eine Partitionierung des Transaktionsbereiches ein, die den Transaktionsprozeß vom Wurzelprozeß trennt. Der Empfang der globalen Entscheidung ist erst nach der Reparatur der Verbindung möglich. Die Zeitdauer dieses Vorgangs ist in keinem Transaktionsprozeß begrenzt. Die Transaktionsprozesse des vom Wurzelprozeß abgekoppelten Teilbaumes sind für einen unabsehbaren Zeitraum blockiert.

Beispiel 5 Ein Transaktionsprozeß empfängt die globale Entscheidung von seinem Vorgänger und leitet sie an alle Nachfolgerprozesse weiter. Die Ausführung der Operationen auf den lokal belegten Transaktionsobjekten sind davon abhängig, daß alle Transaktionsprozesse des unterlagerten Teilbaumes die Entscheidung durchgesetzt haben. Wenn im unterlagerten Teilbaum ein Transaktionsprozeß ausfällt, verzögert sich die lokale Durchsetzung der Entscheidung und der Abschluß des 2-Phasen-Commit-Protokolls für alle übergeordneten Transaktionsprozesse.

Beispiel 5 geht davon aus, daß die Transaktionsprozesse des unterlagerten Teilbaumes zuerst die Entscheidung durchsetzen und danach den TP-Done-Req absetzen. Damit wird das 2-Phasen-Commit-Protokoll in der in Kapitel 2.1.3 angegebenen Form eingesetzt. Der TP-Standard läßt offen, ob zum Zeitpunkt des TP-Done-Req die Entscheidung bereits durchgesetzt ist, oder nur die Kenntnis der Entscheidung bestätigt wird. Die erste Variante

entspricht der in Kapitel 2.1.3 vorgestellten strengen Form des 2-Phasen-Commit-Protokolls. Die zweite Variante entspricht dem Einsatz von ISO-TP zur Realisierung der beiden ersten Phasen des 3-Phasen-Commit-Protokolls. Die dritte Phase wird dann in jeder Subtransaktion lokal, d.h. ohne Kommunikation realisiert.

Die Forderung der Entscheidungsdurchsetzung vor dem TP-Donc-Req bewirkt eine Einschränkung des zulässigen lokalen Verhaltens für die Transaktionsprozesse. Korrektes lokales Verhalten eines Transaktionsprozesses ist in einem Partnerprozeß nicht überprüfbar. Der Commit-Dienst kann für den Transaktionsabschluß nur unterstützend wirken. Die Gewährleistung der ACID-Rules beruht auf vertrauenswürdigen Subtransaktionen, die die Vorschriften für das lokale Verhalten einhalten.

Die in ISO-TP vorgegebene Kommunikationsstruktur beinhaltet keine redundanten Verbindungen. Jeder Ausfall einer Verbindung oder eines Transaktionsprozesses führt zwangsläufig zu einer Partitionierung des Transaktionsbereiches. In diesem Fall können keine Aussagen über das Zeitverhalten getroffen werden.

8 Ein Ansatz zur Behandlung zeitkritischer TP-Anwendungen

Die Bearbeitung zeitkritischer verteilter Transaktionen unter Einsatz des ISO-TP-Dienstes erfordert ergänzende Maßnahmen zur Abschätzung des Zeitverhaltens. In diesem Kapitel wird ein Ansatz vorgestellt, der in jedem beteiligten Prozeß alle drei Kommunikationsphasen berücksichtigt.

8.1 Zielsetzung

Die Unterstützung von Echtzeitanforderungen und Transaktionsverarbeitung für eine verteilte Anwendung führt zu Konflikten bei der Behandlung von Wartesituationen. Ein solcher Konflikt entsteht bei der Verletzung einer Zeitbedingung eines Transaktionsprozesses während einer Blockierungssituation. Die Gewichtung der Anforderungen gibt die Behandlung solcher Konflikte vor. Zur Auswahl stehen zwei Zielsetzungen:

- Die auf eine Verletzung von Zeitanforderungen folgende Ausnahmebehandlung wird in jedem Fall ohne Berücksichtigung der ACID-Rules vollzogen.
- Die Gewährleistung der ACID-Rules wird gegenüber den Echtzeitanforderungen vorrangig behandelt. Eine Ausnahmebehandlung wird nur dann verwirklicht, wenn die Transaktionseigenschaften davon unberührt bleiben.

In dem hier vorgestellten Ansatz werden die Echtzeitanforderungen dem Erhalt der ACID-Eigenschaften untergeordnet. In Konfliktsituationen wird damit eine unbegrenzbare Blockierung hingenommen. Der zur Transaktionsunterstützung eingesetzte ISO[®]TP-Standard legt die Grenzen der Reaktionsmöglichkeiten zur Behandlung von Zeitanforderungen fest. Für einen Transaktionsprozeß bestehen die beiden aus der global festgelegten Entscheidung abgeleiteten Terminierungsmöglichkeiten:

- Die globale Entscheidung erzwingt für jeden Transaktionsprozeß den positiven Abschluß.

- Der Transaktionsabbruch wird von jedem Transaktionsprozeß mitgetragen.

Die transaktionsweite Berücksichtigung von Zeitanforderungen soll hier nur für das Ziel eines positiven Transaktionsabschlusses erreicht werden. Als Reaktion mit transaktionsbezogener Wirkung kommt daher nur der Abbruch der Transaktion in Betracht. Für Transaktionsprozesse, die bereits vor der Überschreitung einer Zeitbedingung aus anderen Gründen den Abbruch der Transaktion einleiten oder von einem Partnerprozeß in das Rollback-Verfahren¹⁷ einbezogen wurden, werden lokale Zeitanforderungen nicht betrachtet.

Die Einleitung eines Transaktionsabbruches zum spätest möglichen Zeitpunkt führt zur Verschwendung von Rechenleistung. Wenn abgebrochene Transaktionen erneut gestartet werden, ergibt sich daraus eine Veränderung der Lastsituation der einzelnen Subsysteme, die sich negativ auf den Zeitverlauf einzelner Transaktionsprozesse auswirken kann. Die Bewertung der Lastsituation und ihres Einflusses auf den Zeitverlauf einzelner Subtransaktionen wird auf die Subsysteme begrenzt. Die Begrenzung des durch häufiges Zurücksetzen entstehenden Leistungsverlustes wird der Einhaltung von Zeitanforderungen untergeordnet.

Die angegebenen Zielvorstellungen sollen unter Beschränkung auf die Kommunikationsstruktur eines Transaktionsbereiches realisierbar sein. Prozeß-Cluster, deren Prozesse in einem Subsystem konkurrierend bearbeitet werden, sind in ISO-TP nicht berücksichtigt. Der Informationsfluß innerhalb eines Clusters [MSF83] durchbricht die Kommunikationsstruktur genau dann, wenn zwischen den kommunizierenden Prozessen kein TP-Dialog besteht. Auf die Auswertung von Zeitinformationen unter Ausnutzung von Clusterstrukturen wird deshalb nicht eingegangen. Jeder Transaktionsprozeß bezieht sein gesamtes Wissen über den Transaktionsbereich und die darin eingebundenen Transaktionsprozesse ausschließlich durch Nachrichtenaustausch mit seinen Verbindungspartnern auf den vorgegebenen Transaktionsverbindungen.

¹⁷zum Abbruch der Transaktion

Unter diesen Vorgaben liegt die Absicht in der Ausschöpfung der mit ISO-TP verträglichen Reaktionsmöglichkeiten für die Behandlung von Zeitbedingungen. Insbesondere sollen die Möglichkeiten zur zeitabhängigen Begrenzung von Warteaktivitäten der Kommunikationsmanager während des 2-Phasen-Commit-Protokoll-Ablaufs erweitert werden. Da eine vollständige Vermeidung aller Blockierungssituationen nicht möglich ist, sollte der Commit-Koordinator von der Verletzung einer Zeitanforderung während des 2-Phasen-Commit-Protokolls in Kenntnis gesetzt werden. Eine solche Mitteilung könnte z.B. für die Einleitung einer kompensierenden verteilten Transaktion von Bedeutung sein.

In dem Ansatz werden folgende Fragestellungen berücksichtigt:

- " Wie lange darf eine verteilte Transaktion maximal dauern bzw. bis zu welchem Zeitpunkt soll die Transaktion spätestens abgeschlossen sein?
- Wie lange darf eine Subtransaktion maximal dauern bzw. bis zu welchem Zeitpunkt soll die Subtransaktion spätestens abgeschlossen sein?
- s Wie lange bzw. bis zu welchem Zeitpunkt darf eine Subtransaktion im Commit-Phase-1-Intervall verharren, ohne Zeitbedingungen zu verletzen?
- e Innerhalb welches Zeitraumes bzw. bis zu welchem Zeitpunkt kann ein Transaktionsprozeß mit dem ungestörten Abschluß der Kommunikation¹⁸ im unterlagerten Teilbaum rechnen?
- e Sind während der Transaktion die Zeitanforderungen aller Transaktionsprozesse eines Teilbaumes eingehalten worden?

Die Aufzählung berücksichtigt sowohl relative als auch absolute Zeitbedingungen. Die Zeitbedingungen werden von jedem Transaktionsprozeß

¹⁸Es wird nur die Kommunikationszeit überwacht. Lokale Aktivitäten, die nach dem TP-Done-Req durchgeführt werden, sind nicht auszuschließen. Die Einbeziehung des Abschlußverhaltens einer Subtransaktion relativ zum Abschluß der Kommunikationszeit, rnmes in einem Partnerprozeß beruht auf anwendungsspezifischen Absprachen. Absprachen über das Verhalten eines Transaktionsprozesses während des Transaktionsabschlusses sind in einem Partnerprozeß nicht kontrollierbar. Die Absprachen beruhen auf der Vertrauenswürdigkeit des kontrollierten Prozesses.

bezüglich der lokalen Uhr des unterstützenden Subsystems formuliert. Für die Datentransferphase sind weitere Anforderungen denkbar, die vom anwendungsspezifischen Kommunikationsverhalten abhängen. Die Behandlung der Zeitanforderungen soll soweit möglich durch lokale Zeitüberwachung erfolgen. Damit wird die Ausdehnung des Nachrichtentransfers auf wenige zusätzliche Botschaften beschränkt.

8.2 Voraussetzungen

Die zeitliche Bewertung des Nachrichtenaustausches zwischen zwei im Transaktionsbereich direkt verbundenen Prozessen unterliegt folgenden Annahmen über das Kommunikationssystem:

- Das Kommunikationssystem gewährleistet eine unverfälschte Übertragung von Nachrichten.
- Nachrichten gehen nicht verloren.
- Die Reihenfolge der auf einer Transaktionsverbindung in gleicher Richtung übertragenen Nachrichten bleibt erhalten.

Diese Forderungen werden in einem Kommunikationssystem, welches Verbindungen der Ebene 6 des ISO-Referenzmodells anbietet, erfüllt. Für die zeitliche Abschätzung des Übertragungsverhaltens wird in [Dol86] angenommen, daß jedes Subsystem für jede Verbindung zu seinen Nachbarn eine obere Schranke für die Übertragung und die Verarbeitung der Nachrichten kennt. Dabei wird auf die Übertragungsrichtung nicht eingegangen. Die beiden Übertragungsrichtungen müssen nicht unbedingt die gleichen Grenzwerte aufweisen. Eine differenzierte Betrachtung der Übertragungsrichtungen und der Lastsituation des Kommunikationssystems wird hier durch folgende Annahmen vermieden:

- Das Kommunikationssystem garantiert bezogen auf jede Verbindung eine minimale und eine maximale Übertragungszeit für die Übertragung einer Nachricht und die Rückübermittlung einer Antwort.
- An der Schnittstelle zum Kommunikationssystem tritt keine Verzögerung der Sendeereignisse durch vorangehendes Queuing ein.

Die maximale Übertragungszeit für eine Transaktionsverbindung $(p, q) \in V$ zweier Transaktionsprozesse $p, q \in P$ wird mit $\hat{U}_{max}(p, q)$, die minimale Übertragungszeit mit $\hat{U}_{min}(p, q)$ notiert.

Für die Abwicklung einzelner Operationen in einer Subtransaktion wird den beteiligten Resource-Managern die Angabe einer oberen Grenze für die Ausführungszeiten abverlangt. Diese Angabe gilt auch für den Kommunikationsmanager. Für jedes Kommunikationsereignis muß bezüglich jeder Verbindung eine Verarbeitungsdauer bekannt sein. Zur Vereinfachung wird für jeden Kommunikationsmanager das in 6.3.1 definierte deterministische Verhalten angenommen, welches verbindungsunabhängige Bearbeitungszeiten für die Kommunikationsereignisse vorgibt.

Auf die Dauer der Queuing-Aktivitäten, d.h. den Zeitraum bis zur Bearbeitung eines anstehenden Ereignisses, wirkt sich die Auslastung des Subsystems aus. Queuing tritt an den Schnittstellen des Kommunikationsmanagers und auch bei anderen Resource-Managern des Subsystems auf. Die Begrenzung von Queuing-Situationen wird in Subsystemen mit garantierten maximalen Antwortzeiten aufgegriffen. Für die Subsysteme, die an einer verteilten Transaktion mitwirken, wird die Begrenzbarkeit von Queuing-Situationen gefordert.

Für die Behandlung von Zeitbedingungen während des 2-Phasen-Commit-Verfahrens ist eine Erweiterung der oben aufgeführten Annahmen nötig, die das Verhalten der lokalen Uhren charakterisiert:

Die lokale Uhr jedes Subsystems läßt sich für einen Prozeß p als lineare Funktion $t_p = a \cdot t + b$ über der realen Zeit t annehmen, wobei a und b konstant sind.

Diese Annahme läßt eine konstante Abweichung zu, erlaubt aber keine Schwankungen der lokalen Uhr. Die Modellierung einer Rechneruhr mit einer solchen idealisierten Uhr ist unrealistisch. Rechneruhren weichen z.B. aufgrund von Temperaturschwankungen oder als Folge eines Synchronisationsverfahrens von dieser Vorstellung ab. Die Annahme wird deshalb zu folgender Voraussetzung abgeschwächt:

- Die Abweichung der durch einen Prozeß p des Transaktionsbereiches charakterisierten lokalen Uhr t eines Subsystems von einer linearen Funktion t , über der realen Zeit t läßt sich durch lineare Grenzfunktionen

$$k_p \cdot t + l_p \leq t'_p \leq m_p \cdot t + n_p \quad (6)$$

beschreiben, wobei k_p, l_p, m_p und n_p konstant sind.

Durch diese Annahme ist für jede lokale Uhr eine obere und eine untere Schranke für die Schwankungen vorgegeben. Die Konstanten verschiedener Uhren werden durch Indizes unterschieden, die die Zuordnung zu den Transaktionsprozessen ausdrücken.

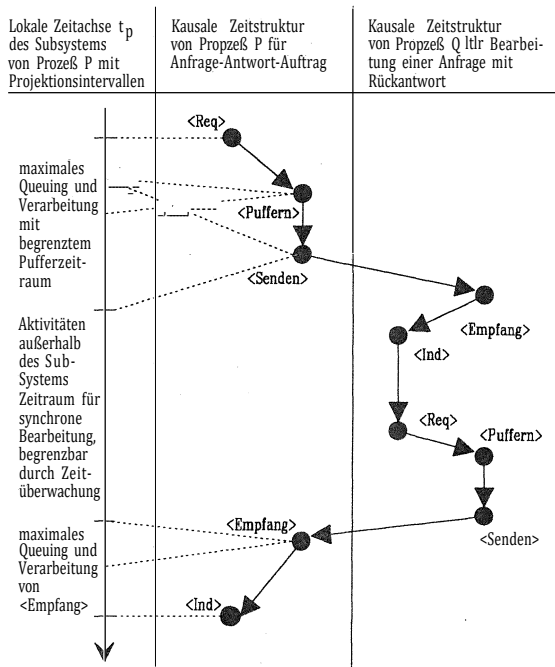
8.3 Lokale Begrenzung des Kommunikationsverhaltens während des Datentransferintervalles

Eine Subtransaktion kann Zeitanforderungen, die sich auf Kommunikationsereignisse innerhalb des Datentransferintervalles beziehen, lokal überwachen und begrenzen. Die lokale Bearbeitungsdauer im Kommunikationsmanager läßt sich aber nicht beeinflussen. Aktivitätsfolgen, die die Grenze des Datentransferintervalles überschreiten, sind lokal auch nicht mehr begrenzbare. Dennoch können Verletzungen von Zeitbedingungen für solche Zeiträume auf lokale Ursachen zurückzuführen sein. Die lokalen Ursachen liegen in der Aufenthaltsdauer von Nachrichten im Kommunikationsmanager. Unbestimmbare Faktoren ergeben sich aus dem

- Queuing-Verhalten und dem
- Puffern von PDUs vor dem Senden.

Für das Queuing wurde oben bereits eine durch die lokale Antwortzeit vorgegebene obere Grenze vorausgesetzt.

Das Puffern von Nachrichten im Sendeprozeß wirkt sich als unbekannte Verzögerung aus. Für die Datentransferphase wird eine Erweiterung der TP-Schnittstelle nötig, die es erlaubt, das Puffern von Nachrichten anwendungsabhängig zu begrenzen. Diese Maßnahme wirkt sich auf die synchrone



<Req>, <Ind> : Wahrnehmbare Ereignisse an der TP-Schnittstelle
 <Puffern>: Internes Ereignis des Kommunikationsmanagers
 <Senden>, <Empfang>: Ereignisse an der Schnittstelle zum Kommunikationssystem
 — : Intervalle, die bei Kenntnis des maximalen Zeitverhaltens im
 Kommunikationsmanager für die Subtransaktion des Prozesses
 P abschätzbar sind.

Abbildung 17: Toleranzen synchroner Bearbeitungszeiträume bei Begrenzung des Pufferhaltens

Bearbeitung von Operationen innerhalb einer Transaktion aus. Ein Transaktionsprozeß, der einen Auftrag an einen Partnerprozeß vergibt und bis zum Eintreffen der Antwort einen zum Auftrag synchronen internen Bearbeitungsschritt durchführen will, kann die Dauer der entfernten Auftragsabwicklung nur dann abschätzen, wenn die lokalen Kommunikationsaktivitäten zeitlich begrenzt sind. Die maximale Aufenthaltsdauer der Nachricht im Puffer addiert sich zur maximalen Übertragungszeit. Wenn die Pufferung unbeschränkt ist, kann nur der frühest mögliche Empfangszeitpunkt des Auftrages abgeschätzt werden. Abbildung 17 zeigt dagegen die Toleranzen der synchronen Bearbeitungszeiträume bei begrenzter Pufferung. Der dargestellte Ablauf gibt eine Anfrage eines Prozesses p an einen Partnerprozeß q und die Übermittlung einer Antwort wieder. Dabei werden die internen Ereignisse der Kommunikationsmanager, insbesondere die Pufferung erfaßt. Die Projektionen der kausalen Ereignisse auf die lokale Zeitachse des Transaktionsprozesses p gibt den Kenntnisstand der Subtransaktion des Prozesses p wieder.

Bereits bei der Eröffnung einer Transaktionsverbindung kann durch Begrenzung der Pufferung die Parallelisierung innerhalb der verteilten Transaktion gefördert werden. Dies kann sich beschleunigend auf die Dauer der Datentransferphasen im unterlagerten Teilbaum auswirken. Für den Wurzelprozeß lassen sich durch die Begrenzung der Pufferung lokale Ursachen für die Überschreitung der geforderten Transaktionsdauer ausschließen.

Zeiträume, die durch zwei Ereignisse der Datentransferphase eines Transaktionsprozesses begrenzt sind, werden vollständig lokal überwacht. Zeiträume, die im Datentransferintervall beginnen und in einem der beiden Commit-Intervalle enden, werden während der Datentransferphase lokal überwacht. Wenn eine Verletzung der Zeitbedingung bereits während des Datentransferintervalles erkennbar ist, wird dies im lokalen Transaktionsprozeß erkannt und behandelt. Beispiele für solche Zeitanforderungen können

- die Dauer der Beteiligung eines Transaktionsprozesses an einer verteilten Transaktion bis zu ihrem lokalen Abschluß,
- die Dauer der Beteiligung eines Transaktionsprozesses bis zum Eintreffen der Entscheidung,

- die Dauer vom Absenden oder Empfang einer PDU mit Benutzerdaten bis zum lokalen Abschluß der Transaktion oder
- die Dauer vom Absenden oder Empfang einer PDU mit Benutzerdaten bis zum Eintreffen der Entscheidung

betreffen. Vorstellbar sind obere und untere Grenzwertbedingungen für solche Zeiträume. An der Aufzählung wird deutlich, daß auch Zeitanforderungen formulierbar sind, deren Bezugsereignisse verschiedenen Verbindungen zuzuordnen sind.

8.4 Die Basis für den Transfer zeitbezogener Daten

Für die Übermittlung von zeitbezogenen Daten auf einer Transaktionsverbindung wird eine Basis gewählt, die in beiden Verbindungspartner-Prozessen eingrenzbar ist. Zeitbezogene Daten können Zeitanforderungen charakterisieren oder zur Angabe von Zeiträumen benutzt werden. Die Basen für die Interpretation zeitbezogener Daten werden in einem Transaktionsprozesses während der Commit-Vorbereitung der Nachfolgerprozesse und der eigenen Commit-Vorbereitung wie folgt bestimmt:

Jeder Transaktionsprozeß p führt eine Vektoruhr f_{ip} und ordnet jedem Verbindungspartner $q_x \in \text{next}(p)$ eine Komponente $u(q_x)$ dieser Uhr zu. Außerdem ist eine Komponente für die Verbindung zum Vorgänger, $\text{prev}(p)$, enthalten. Alle Sende- und Empfangsereignisse erhalten einen lokalen Zeitstempel t' und verändern damit die Komponente der Vektoruhr, die dem entsprechenden Verbindungspartner zugeordnet ist.

Uhrvektor:

$$\{\bar{u}_p\} = \begin{pmatrix} t(e(\text{prev}(p))) \\ t(e(q_1)) \\ t(e(q_2)) \\ \vdots \\ t'_p(e(q_{|\text{next}(p)|})) \end{pmatrix}$$

Jeder Komponente des Uhrvektors ist aufgrund der Schwankungen des lokalen Uhr ein Unsicherheitsintervall zugeordnet. Für

einen Transaktionsprozeß p , $E \text{ next}(p) \cup \text{prev}(p)$ ergibt sich das Unsicherheitsintervall eines von p , abhängenden Ereignisses $e(p_x)$ aus:

$$t(e(p_x))_{1+} [\underline{t}_p^{\text{in}}(e(p_x)), \underline{t}_p^{\text{max}}(e(p_x))] \\ t_{\text{min}}(e(p_x)) = \frac{t(e(p_x)) - n_p}{m_p} \quad (7)$$

$$t_p^{\text{max}}(e(p_x)) = \frac{t'_p(e(p_x)) - l_p}{k_p} \quad (8)$$

Die Konstanten der für einen Prozeß p lesbaren lokalen Uhr k_p, l_p, m_p, n_p wurden in Gleichung 6 auf Seite 114 eingeführt.

In einer zweispaltigen Matrix b mit Zuordnung der Zeilen zu den Verbindungen werden für jeden Nachfolgerprozeß maximale und minimale Projektionen der Zeiträume zwischen Empfang der Prepare-RI-PDU beim Nachfolgerprozeß und dem Senden der Ready-RI-PDU vom Nachfolgerprozeß $ve(\text{zeit})met$. In der Berechnung dieser Komponenten werden die Schwankungsbreite der lokalen Uhr und die Unsicherheit der Übertragungszeiten berücksichtigt. Für die Verbindung eines Prozesses p zum Vorgängerprozeß werden die beiden Werte aus der Zeitdifferenz zwischen dem Ereignis *Empfang - Prepare - RI - PDU*($\text{prev}(p)$) und dem Ereignis *Senden-Ready-RI-PDU*($\text{prev}(p), p$) bestimmt. Dieser Zeitraum enthält keine Übertragung zum Vorgängerprozeß. Deshalb wird nur die Schwankungsbreite der lokalen Uhr in die Berechnung einbezogen.

In Abbildung 18 sind die beschriebenen Projektionen und die Basiszeiträume für einen Prozeß p und einen seiner Nachfolgerprozesse $q \in \text{next}(p)$ dargestellt. Für p ergeben sich die Maximal- und die Minimalkomponente der Basismatrix b_p für den Prozeß q . Für q wird die Erzeugung der Komponenten der Basismatrix b_q für den Vorgängerprozeß p gezeigt.

Die Matrix b_p gibt für jede Verbindung eines Prozesses p zwei Basiswerte für die Interpretation von zeitbezogenen Daten des zugehörigen Partnerprozesses bezüglich der linearen Zeitfunktion t_p vor.

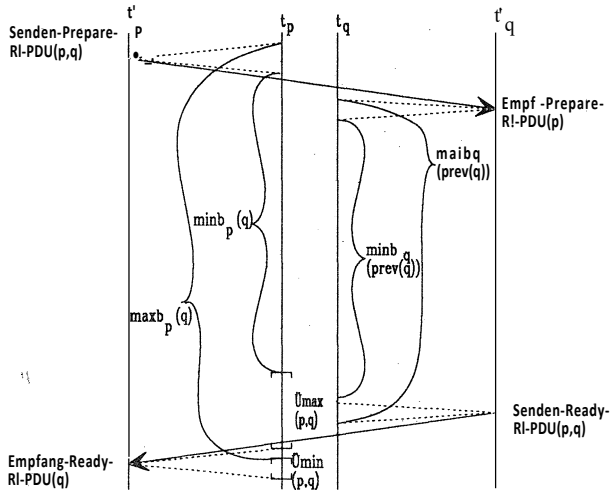


Abbildung 18: Projektionen zur Ermittlung der Werte der Basismatrix

Basismatrix:

$$\{\bar{b}_p\} = \begin{pmatrix} \min b_p(\text{prev}(p)) & \max b_p(\text{prev}(p)) \\ \min b_p(q1) & \max b_p(q1) \\ \min b_p(q2) & \max b_p(q2) \\ \vdots & \vdots \\ \min b_p(q_{|\text{next}(p)|}) & \max b_p(q_{|\text{next}(p)|}) \end{pmatrix}$$

Die Komponenten dieser Matrix ergeben sich für $i \in \{1, \dots, |\text{next}(p)|\}$ aus der folgenden Berechnung.

$$\begin{aligned} \max b_p(q_i) &:= t_i \cdot x(\text{Empfang} - \text{Ready} - \text{RI} - \text{PDU}(q_i)) \\ &- t_i \cdot x(\text{Senden} - \text{Prepare} - \text{RI} - \text{PDU}(p, q_i)) \\ &- f_{\min}(p, q_i) \end{aligned}$$

$$t_i \cdot x(\text{Empfang} - \text{Ready} - \text{RI} - \text{PDU}(q_i)) - t_i \cdot x(\text{Senden} - \text{Prepare} - \text{RI} - \text{PDU}(p, q_i)) > f_{\max}(p, q_i):$$

$$\begin{aligned} \min b_p(q_i) &:= t_i \cdot u(\text{Empfang} - \text{Ready} - \text{RI} - \text{PDU}(q_i)) \\ &- t_i \cdot x(\text{Senden} - \text{Prepare} - \text{RI} - \text{PDU}(p, q_i)) \\ &- \{1, \max(p, q_i)\} \end{aligned}$$

$t'(\text{in(Empfang - Ready- RI - PDU}(q,)) - t'(\text{"x(Senden - Prepare- RI - PDU}(p, q)) \quad \text{Umax}(p, q) :$

$$\text{minbv}(q;) \neq 0;$$

Für die Verbindung zum Vorgängerprozeß werden die Zeitpunkte beim Empfang der Prepare-RI-PDU und zum Absenden der Ready-RI-PDU herangezogen. Der Empfang findet in diesem Fall vor dem Senden statt. Es ergibt sich:

$$\begin{aligned} \text{maxbv}(\text{prev}(p)) &:= t'; \text{"x(Senden - Ready- RI - PDU}(\text{prev}(p), p)) \\ &- t'; \text{in(Empfang - Prepare - RI - PDU}(\text{prev}(p))) \\ \text{minbv}(\text{prev}(p)) &:= t'; \text{in(Senden - Ready- RI - PDU}(\text{prev}(p), p)) \\ &- t'; \text{"x(Empfang- Prepare - RI - PDU}(\text{prev}(p), p)) \end{aligned}$$

Die Auswahl eines Basiswertes zur Übermittlung bzw. Interpretation zeitbezogener Daten richtet sich nach der Zielsetzung der Daten. Zeitbedingungen, die eine obere Grenze für den Eintrittszeitpunkt eines Ereignisses vorgeben, sind unter Verwendung des Maximalwertes zu kodieren und unter Anwendung des verbindungsbezogenen Minimalwertes vom Partnerprozeß zu interpretieren. Für ein Ereignis, welches bezüglich der lokalen Uhr des Subsystems bis spätestens zum Zeitpunkt x eingetreten sein muß, wird eine Kodierung gewählt, die beim Kommunikationspartner den frühesten Zeitpunkt zu konstruieren erlaubt, zu dem die Bedingung unter ungünstigen Voraussetzungen verletzt werden kann.

Umgekehrt wird die Angabe einer unteren Grenze für den Eintrittszeitpunkt eines Ereignisses unter Verwendung des Minimalwertes kodiert und unter Anwendung des verbindungsbezogenen Maximalwertes vom Partnerprozeß interpretiert. Für ein Ereignis, welches frühestens zum Zeitpunkt x eintreten darf, wird die Bedingung so kodiert, daß sie den unter ungünstigen Voraussetzungen spätesten Zeitpunkt beschreibt, zu dem sie gerade noch verletzt werden kann.

Die Angabe eines Zeitbedarfs wird unter Bezug auf den Minimalwert kodiert und unter Einsatz des Maximalwertes im Verbindungspartner interpretiert.

8.5 Zeitüberwachung des Commit-Phase-1-Intervalles

Zeitbedingungen, die sich während der Commit-Phase-1 auf das Verhalten der Nachfolgerprozesse beziehen, können durch vorgezogenes Prepare in das Datentransferintervall verlagert und damit lokal überwacht werden. Als einziges Empfangsereignis bleibt dann für die Commit-Phase-1 der Empfang der Entscheidung vom Vorgängerprozeß abzuwarten. Die Überwachung von Zeitbedingungen eines Transaktionsprozesses, die sich auf dieses Ereignis beziehen, wird auf übergeordnete Transaktionsprozesse verlagert, die sich noch in der Datentransferphase befinden.

Begrenzung der Wartezeit auf die Entscheidung: Mit dem Absenden der TP-Ready-RI-PDU an den Vorgängerprozeß kann eine Zeitanforderung bezüglich der Ankunft der Entscheidung nach oben weitergegeben werden. Die Zeitanforderung wird als Zeitraum vom Absenden der Ready-RI-PDU nach oben bis zum spätesten akzeptierten Empfangszeitpunkt der Entscheidung formuliert. Für diesen Zeitraum wird durch eine Projektion auf die lineare Zeitfunktion eine kleinste obere Schranke ermittelt. Von der ermittelten Dauer wird zunächst die maximale Übertragungszeit abgezogen, um festzustellen, ob die Zeitbedingung durch die Übertragung verletzt wird. Der verbleibende Zeitraum wird durch die Maximal-Basis-Komponente für den Vorgänger $\max_{b_p}(\text{prev}(p))$ dividiert und in dieser Form an den Vorgänger übertragen.

Wenn die Berechnung dieser Zeitschranke vor dem TP-Commit-Req erfolgt, kann die Transaktion noch zurückgesetzt werden. Dies setzt voraus, daß nach dem Eintreffen der Ready-RI-PDUs aller Nachfolger alle Aktivitäten des lokalen Kommunikationsmanagers und die der lokalen Subtransaktion abgrenzbar sind und eine Bestimmung des Sendzeitpunktes für die Ready-RI-PDU im voraus erlauben.

Der Vorgängerprozeß bestimmt beim Eintreffen der Ready-RI-PDU die beiden Basiswerte für diese Verbindung aus den Grenzfunktionen der eigenen Uhr und den maximalen und minimalen Übertragungszeiten. Er vervollständigt damit die lokale Basismatrix.

Er bewertet dann die Zeitanforderungen von seinen Nachfolgern mit Hilfe der verbindungs-spezifischen Minimum-Komponenten der Basismatrix. Die Überwachung einer solchen Zeitanforderung muß relativ zur lokalen Uhr erfolgen und den Zeitraum bis zum Absenden einer Ready-RI-PDU an den Vorgängerprozeß erfassen. Es genügt, die härtesten vorliegenden Zeitanforderungen lokal zu überwachen.

Beispiel: Eine Zeitanforderung, die den maximalen Wartezeitraum eines Transaktionsprozesses q begrenzt, $[t(\text{Senden-Ready-RI-PDU}(p, q)), t(\text{Empfang - Commit - RI - PDU}(p))]$, wurde bezüglich der linearen Zeitfunktion t_0 nach unten abgeschätzt. Nach Abzug der maximalen Übertragungszeit übergibt der Prozeß q den Schätzwert in der Ready-RI-PDU in Form eines Faktors h an den Vorgängerprozeß p . Abbildung 19 zeigt die zur Übertragung der maximalen Wartezeit notwendigen Projektionen, die im folgenden beschrieben werden. Der Faktor h ergibt sich aus der Division des von Prozeß q abgeschätzten Zeitraumes durch die Komponente $maxb_0(prev(q))$ der Basismatrix von q . Aus dem durch Zeitstempel festgehaltenen Empfangszeitpunkt der Ready-RI-PDU im Prozeß p $t(\text{Empfang - Ready - RI - PDU}(q))$, und der Gleichung 7 wird der frühest mögliche Empfangszeitpunkt $t'^{in}(\text{Empfang - Ready - RI - PDU}(q))$ bezüglich der linearen Zeitfunktion t_p für diese PDU bestimmt.

Der späteste _akzeptierbare Sendezeitpunkt für die Commit-RI-PDU an den Nachfolger q wird aus der Zeit $t'^{in}(\text{Empfang - Ready - RI - PDU}(q))$ durch Addition der mit dem Faktor h multiplizierten Minimalbasis $minv(q)$ abgeschätzt.

$t'^{in}(\text{Senden - Commit - RI - PDU}(p, q)) :=$

$$t'^{in}(\text{Empfang - Ready - RI - PDU}(q)) + h \cdot minv(q);$$

Zur Überwachung der Zeitanforderung bezüglich der lokalen Uhr muß dieser Wert auf die lokale Zeitachse projiziert werden. Aus der Gleichung 7 auf Seite 118 wird durch Minimumabschätzung der späteste zulässige Sendezeitpunkt für die Entscheidung, $t(\text{Senden - Commit - RI - PDU}(p, q))$, bestimmt:

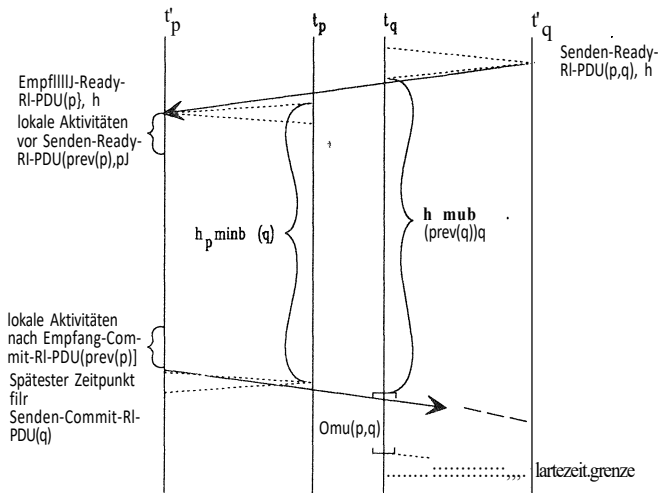


Abbildung 19: Projektionen zur Übertragung der maximalen Wartezeit an den Vorgängerprozeß

$$t(\text{Senden} - \text{Commit} - \text{RI} - \text{PDU}(p,q)) \equiv$$

$$t_{\text{min}}(\text{Senden} - \text{Commit} - \text{RI} - \text{PDU}(p,q)) \cdot m_p + n_p;$$

Der so ermittelte Wert wird vom Transaktionsprozeß p für den Nachfolgerprozeß q solange überwacht, bis eine härtere Zeitanforderung vorliegt oder die Ready-RI-PDU nach oben abgesendet wird. In dem Vergleich der Zeitanforderungen wird ggf. auch eine lokale Zeitanforderung berücksichtigt.

Mit dem Absenden der Ready-RI-PDU an den Vorgängerprozeß muß die härteste Wartezeitanforderung ermittelt werden. Zum einen muß die lokal benötigte Bearbeitungszeit zwischen dem Eintreffen einer Zeitanforderung und dem Weiterleiten der härtesten Zeitanforderung an den Vorgängerprozeß berücksichtigt werden. Zum anderen geht die lokal benötigte Zeit zwischen dem Empfang der Entscheidung, *Empfang-Commit-RI-PDU(prev(p))*, und der Entscheidungsweiterleitung, *Senden-Commit-RI-PDU(p, q)*, in die Abschätzung ein.

Diese Zeiträume sind in Kapitel 6 als lokale Aktivitätskompositionen¹⁹ erkennbar und daher lokal begrenzt. Aktivitäten, die vor dem Absenden der Ready-RI-PDU anfallen, werden durch Festlegung eines neuen Bezugszeitpunktes, dem Sendeereignis der Ready-RI-PDU an den Vorgängerprozeß, erfaßt. Aktivitäten, die erst danach ausgeführt werden, verschieben die kritische Zeitschranke auf einen früheren Zeitpunkt. Die Zeitanforderung wird dadurch auf einen engeren Zeitraum begrenzt.

Wenn die härteste Zeitanforderung nach Abzug der maximalen Übertragungszeit für die Verbindung nach oben nicht verletzt ist, folgt die Kodierung entsprechend der Maximum-Komponente der Basismatrix für die Verbindung zum Vorgängerprozeß. Die Bedingung wird mit der Ready-RI-PDU an den Vorgängerprozeß übergeben.

Für den Fall, daß ein Prozeß die Weiterleitung einer Zeitbedingung nicht mehr verantworten kann und den Transaktionsabbruch einleiten kann, werden zunächst baumabwärts alle Prozesse erreicht und damit auch der Transaktionsprozeß, der die Zeitanforderung vorgegeben hat.

- Wenn die Bearbeitungszeiten der einzelnen Transaktionsprozesse, die an der Weiterleitung des Transaktionsabbruchs beteiligt sind, kleiner oder gleich den Bearbeitungszeiten für die positive Entscheidung sind
- und kein von der Weiterleitung betroffener Transaktionsprozeß zwischenzeitlich ausgefallen ist
- und alle Verbindungen im Teilbaum noch intakt sind,

kommt die Abbruch-Entscheidung in jedem Transaktionsprozeß des Teilbaumes rechtzeitig an.

Andere Maximalzeitanforderungen im Commit-Phase-1-Intervall:

Unter der Annahme, daß jeder Transaktionsprozeß seine Nachfolgerprozesse mit vorgezogener Commit-Vorbereitung in das Commit-Phase-1-Intervall überführt und alle Ready-RI-PDUs noch in seinem Da-

¹⁹Es sind keine Warteaktivitäten auf Ereignisse der TP-Schnittstelle oder der Schnittstelle zum Kommunikationssystem enthalten.

tentransferintervall empfangen hat, bildet der Empfang der Entscheidung das einzige nicht lokal erzeugbare Ereignis des Commit-Phase-1-Intervalls. Alle Zeitanforderungen für dieses Intervall hängen mit diesem Ereignis, dem Eintrittszeitpunkt der Entscheidung, zusammen. Neben der im oben angeführten Beispiel beschriebenen Zeitanforderung sind weitere Maximum-Bedingungen denkbar, die einen Zeitraum umfassen, der erst im Commit-Phase-2-Intervall endet. Diese Zeitanforderungen müssen bereits im Commit-Phase-1-Intervall entsprechend dem beschriebenen Vorgehen behandelt werden. Für Nachfolgerprozesse, die nur eine solche Anforderung an den Vorgängerprozeß übergeben, geht diese in den Vergleich aller vorliegenden Maximalanforderungen des Commit-Phase-1-Intervalls ein.

Behandlung von Minimalanforderungen im Commit-Phase-1-

Intervall: Minimum-Bedingungen für das Commit-Phase-1-Intervall lassen sich ebenfalls an den Vorgängerprozeß weiterreichen. Alle Abschätzungen müssen konträr zum oben beschriebenen Verfahren durchgeführt werden. Eine solche Bedingung, deren Überwachung durch Weiterleiten mit der Ready-RI-PDU ermöglicht werden soll, wäre aber durch bewußtes Verzögern im Vorgängerprozeß erfüllbar und ist daher nicht als kritische Zeitanforderung zu beurteilen. Der unmittelbare Vorgängerprozeß muß nur prüfen, ob die Anforderung mit seiner härtesten Maximum-Anforderung bezüglich des Commit-Phase-2-Intervalls verträglich ist. Minimalanforderungen, die sich auf Ereignisse im Commit-Phase-2-Intervall beziehen, lassen sich ebenso erfüllen. Der Übergang des Nachfolgerprozesses vom Commit-Phase-1-Intervall in das Commit-Phase-2-Intervall wird verzögert. Andere Einflußmöglichkeiten eines Transaktionsprozesses auf die minimale Dauer des Commit-Phase-2-Intervalls in einem Nachfolgerprozeß bestehen nicht.

8.6 Abschätzung der Dauer der Commit-Phase-2

Für die Dauer der Commit-Phase-2 kann nur eine vorausschauende Abschätzung eingesetzt werden. Die Vorhersehbarkeit des lokalen Zeitverhaltens der einzelnen an der Transaktion beteiligten Transaktionsprozesse muß gewähr-

leistet sein, um eine solche Abschätzung zu erreichen. Durch Gegenüberstellung der erwarteten Dauer des Commit-Phase-2-Intervalls eines Transaktionsprozesses mit der härtesten ihm vorliegenden Zeitanforderung für dieses Intervall soll eine frühzeitige Erkennung von Konflikten erreicht werden. Der Zeitpunkt der Abschätzung muß so gewählt werden, daß im Konfliktfall ein Abbruch der Transaktion noch möglich ist. Für den Wurzelprozeß ergibt sich als spätester Zeitpunkt das Ende des Commit-Phase-1-Intervalls, für alle anderen Transaktionsprozesse das Ende des Datentransferintervalles.

Ermittlung des Zeitbedarfs für die Commit-Phase-2: Für einen Blattprozeß q ist die voraussichtliche Maximaldauer des Commit-Phase-2-Intervalls bis zum TP-Done-Req nur durch lokale Aktivitäten bestimmt. Für die Projektion dieses Zeitraumes auf die lineare Zeitfunktion t_q wird die Maximalabschätzung zur Kodierung herangezogen. Es ergibt sich das Intervall $[t_q^n(\text{Empfang} - \text{Commit} - \text{RI} - \text{PDU}(p), q), t_q^x(\text{Senden} - \text{Commit} - \text{RC} - \text{PDU}(p, q))]$. Die Berechnung dieses Intervalls aus dem bezüglich der lokalen Uhr bekannten Zeitbedarfs ist im allgemeinen, d.h. für $m, \# 1$ oder $k, \# 1$, abhängig von dem Empfangszeitpunkt der globalen Entscheidung. Deshalb beruht der Schätzwert bezüglich der linearen Zeitfunktion t , auf der Vorgabe eines spätesten Zeitpunktes für das Eintreffen der Entscheidung, $t_q^n(\text{Empfang} - \text{Commit} - \text{RI} - \text{PDU}(p))$ ²⁰. Die beiden Schätzwerte berechnen sich wie folgt:

$$t_q^n(\text{Empfang} - \text{Commit} - \text{RI} - \text{PDU}(p), q) := \frac{t_q^n(\text{Empfang} - \text{Commit} - \text{RI} - \text{PDU}(p)) - n_q}{m_q}$$

$$t_q^x(\text{Senden} - \text{Commit} - \text{RC} - \text{PDU}(p, q)) := \frac{t_q^x(\text{Senden} - \text{Commit} - \text{RC} - \text{PDU}(p, q)) - 1}{k},$$

Dabei ergibt sich $t_q^x(\text{Senden} - \text{Commit} - \text{RC} - \text{PDU}(p))$ aus $t_q^n(\text{Empfang} - \text{Commit} - \text{RI} - \text{PDU}(p))$ durch Addition des bezüglich

²⁰Die Übermittlung einer solchen Vorgabe an den Vorgängerprozeß wurde im vorliegenden Kapitel 8.5 behandelt.

der lokalen Uhr bekannten Zeitbedarfs. Der Schätzwert des Zeitbedarfs, der durch das Intervall $[t'; n(\text{Empfang} - \text{Commit} - \text{RI} - \text{PDU}(p), q), t'; n(\text{Senden} - \text{Commit} - \text{RC} - \text{PDU}(p, q))]$ gegeben ist, wird durch die Minimalbasis $\min_b_q(\text{prev}(q))$ dividiert. Der sich ergebende Faktor g wird mit der Ready-RI-PDU an den Vorgängerprozeß p übergeben.

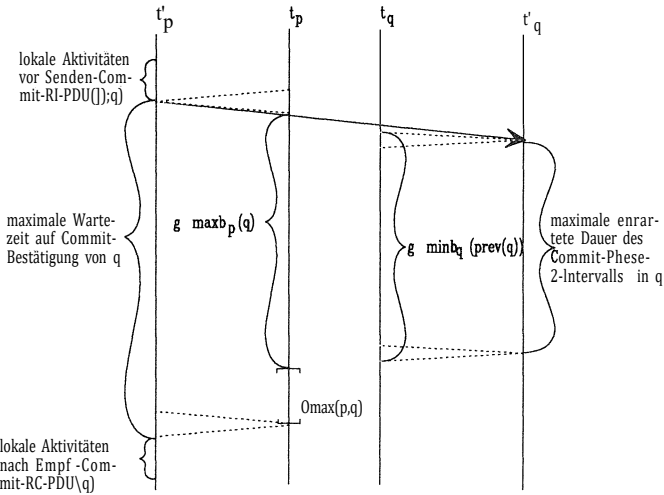


Abbildung 20: Abschätzung des maximalen Zeitbedarfs eines Nachfolgerprozesses

Jeder Transaktionsprozeß erhält beim Empfang einer Ready-RI-PDU von einem Nachfolger den maximalen Zeitbedarf für dessen Commit-Phase-2-Intervall. Er berechnet aus der Multiplikation des Faktors g mit der Maximalbasis-Komponente der Verbindung die maximale Abschätzung für das Commit-Phase-2-Intervall seines Verbindungspartners und addiert die maximale Übertragungszeit für diese Verbindung. Unter Berücksichtigung der Schwankungen der lokalen Uhr läßt sich der Zeitraum auf die lokale Zeitachse des Prozesses p projizieren und so eine maximale Wartezeit bezüglich der lokalen Uhr ermitteln. Abbildung 20 zeigt die Projektionen zur Abschätzung des maximalen Zeitbedarfs des Commit-Phase-2-Intervalls eines Prozesses q im

Vorgängerprozeß p

Nach dem Empfang aller Ready-RI-PDUs läßt sich unter Einbeziehung aller notwendigen lokalen Aktivitäten ein maximaler Zeitbedarf für das eigene Commit-Phase-2-Intervall bestimmen. Der Zeitbedarf für dieses Intervall hängt von der Reihenfolge der Entscheidungsweiterleitung an die Nachfolgerprozesse ab. Die eingesetzte Strategie beeinflusst die Menge der Aktivitäten vor einer Entscheidungsweiterleitung an einen Nachfolgerprozeß. Durch bevorzugte Behandlung von Nachfolgerprozessen mit einem hohen Zeitbedarf für die Commit-Phase-2 kann die Dauer des lokalen Commit-Phase-2-Intervalls verkürzt werden. Die Kodierung dieser Zeitdauer erfolgt wie für den Blattprozeß beschrieben und bezieht die maximale Übertragungszeit zum Vorgängerprozeß mit ein.

Der erwartete minimale Zeitbedarf eines Prozesses q kann durch eine entsprechende Abschätzung unter Einsatz der maximalen Basiskomponente von q für die Verbindung zu p zum Kodieren und der minimalen Basiskomponente von p für die Verbindung zu q zum Interpretieren übertragen werden. Dabei ist anstelle der maximalen Übertragungszeit die minimale Übertragungszeit zu berücksichtigen.

Begrenzung der Wartezeit auf die Commit-Bestätigung: Zunächst werden die in einem Transaktionsprozeß p vorliegenden Maximalzeitanforderungen betrachtet, die sich auf den Zeitraum zwischen dem Empfang der Entscheidung, *Empfang-Commit-RI-PDU(prev(p))*, und dem Absenden der Commit-Bestätigung, *Senden - Commit - RC - PDU(prev(p))*, beziehen. Der Prozeß p analysiert zu diesem Zweck alle von seinen Nachfolgern mit den Ready-RI-PDUs vermittelten maximalen Zeitanforderungen bezüglich dieses Zeitraumes und ermittelt im Vergleich mit der lokalen maximalen Zeitanforderung die härteste Bedingung. Dieser Vorgang erfolgt entsprechend den im vorangehenden Kapitel 8.5 beschriebenen Projektionen und dem nachfolgenden Vergleich. Die härteste Bedingung wird nun dem maximalen Zeitbedarf für diesen Zeitraum gegenübergestellt, um einen Konflikt frühzeitig zu erkennen und zu behandeln.

Andere Maximalbedingungen während der Commit-Phase-2: Maximalanforderungen des Commit-Phase-2-Intervalls können auch für Zeiträume vorliegen, die hNP:its in der Commit-Phase-1 oder in der Datentransferphase beginnend. Auf solche Anforderungen wurde bereits im vorangehenden Kapitel 8.1i kurz eingegangen. Aus einer solchen Zeitanforderung läßt sich unter Berücksichtigung des erwarteten maximalen Zeitbedarfs für das Commit-Phase-2-Intervall der späteste Zeitpunkt für das Eintreffen der Entscheidung ermitteln. Dieser Zeitpunkt wird dann als Anforderung des Commit-Phase-1-Intervalls behandelt. Unter solche Zeitanforderungen fallen Bedingungen bezüglich der maximalen Dauer der Transaktion im unterlagerten Teilbaum oder die Dauer vom Eintritt in das Commit-Phase-1-Intervall bis zum Ende der Transaktion im unterlagerten Teilbaum und in der lokalen Subtransaktion.

Synchronisationsanforderungen für die Commit-Phase-2: Für das Commit-Phase-2-Intervall können Synchronisationsanforderungen bezüglich der Bearbeitungszeiträume der Nachfolgerprozesse und der lokalen Subtransaktion vorliegen, wenn die Bearbeitungszeiträume der Nachfolgerprozesse eines Transaktionsprozesses untereinander oder mit der lokalen Subtransaktion eine maximale Überdeckung erreichen sollen, muß der erwartete maximale Zeitbedarf der einzelnen Prozesse für die Reihenfolge der Entscheidungsweiterleitung ausgewertet werden. Neben dem erwarteten maximalen Zeitbedarf ist hier auch der erwartete minimale Zeitbedarf interessant. Aus der Gegenüberstellung des kleinsten erwarteten minimalen Zeitbedarfs unter allen Nachfolgern und ggf. der lokalen Subtransaktion mit dem größten maximalen erwarteten Zeitbedarf läßt sich die minimale Überdeckung der Bearbeitungszeiträume berechnen. Aus der Differenz zwischen diesen beiden Werten ergibt sich die Toleranz bei der Synchronisation der betrachteten Bearbeitungszeiträume. Sie kann einer entsprechenden Anforderung gegenübergestellt werden.

Minimalanforderungen für die Commit-Phase-2: Neben Maximalanforderungen sind auch Minimalanforderungen für das Commit-Phase-2-Intervall eines Transaktionsprozesses vorstellbar. Beispielsweise garantiert ein Transaktionsprozeß die übermittelte maximale erwartete

Dauer des Commit-Phase-2-Intervalls nur dann, wenn die Entscheidung nicht vor einem definierten Zeitpunkt eintrifft. Solche Anforderungen können im lokalen Kommunikationsmanager durch entsprechende Verzögerung nach dem Empfang erreicht werden. Eine solche Verzögerung führt eine neue lokale Aktivität ein und wirkt sich damit auch auf den maximalen und den minimalen Zeitbedarf eines Transaktionsprozesses aus. Die in Kapitel 8.5 angegebene Behandlung von Minimalanforderungen durch den Vorgängerprozeß ist ebenfalls möglich.

Die Kombination der Angabe eines Zeitbedarfs für das Commit-Phase-2-Intervall mit Zeitbedingungen für das Commit-Phase-1- oder -2-Intervall erlaubt einem Transaktionsprozeß die Einschränkung der Gültigkeit seiner Aussage über den Zeitbedarf. Er garantiert den angegebenen Zeitbedarf nur, wenn die gleichzeitig übermittelten Zeitbedingungen eingehalten werden. Die Schwankungen der lokalen Uhr müssen nur während des Gültigkeitsbereiches den Grenzfunktionen genügen.

Verletzungen von Zeitanforderungen im Commit-Phase-2-Intervall treten beim Einsatz des beschriebenen Vorgehens nur dann ein, wenn eine Störung einer Verbindung oder eines Subsystems vorliegt, die eine Abweichung von den vorgegebenen Voraussetzungen bzw. dem angegebenen Verhalten nach sich zieht. Dieses Restrisiko läßt sich nicht vermeiden. Die Verletzung einer Zeitbedingung in dieser Phase kann nur noch bekanntgegeben werden, hat aber keinen Einfluß auf den Abschluß der Transaktion.

8.7 Konsequenzen aus der Unterstützung von Zeitbedingungen

Die durchgeführte Staffelung der Maßnahmen zur Berücksichtigung von Zeitbedingungen einer verteilten Transaktion orientiert sich an den in Kapitel 4.2.1 eingeführten Kommunikationsphasen. Sie charakterisieren den Reaktionsfreiraum einer Subtransaktion in Bezug auf den Ablauf der gesamten Transaktion. Dieser Freiraum läßt sich anhand der in einer Kommunikationsphase zugänglichen Kommunikationsdienste beschreiben.

In der Datentransferphase ist durch den TP-Dienst und die eingebetteten anwendungsspezifischen Kommunikationsdienste eine Vielzahl von anwendungsgesteuerten Reaktionen auf die Verletzung von Zeitbedingungen möglich. In dieser Phase waren jedoch beliebige Verzögerungen der zu versenden Nachrichten durch den Kommunikationsmanager erlaubt. Mit einer durch die Subtransaktion gesteuerten Aufenthaltsbeschränkung von Nachrichten im Kommunikationsmanager läßt sich für jede Verbindung ein Unsicherheitsintervall für den lokalen Zeitbedarf bis zur Übertragung einer Nachricht definieren. Damit wird dem lokalen Kommunikationsmanager eine Zeitbeschränkung bezüglich zu versendender Nachrichten auferlegt. Die Subtransaktion kann nun die Zeit für die Übertragung eines Auftrages an ihren Vorgänger oder einen Nachfolger im Transaktionsbereich, die Auftragsabwicklung und die Rückübermittlung der Antwort durch Zeitüberwachung kontrollieren. Der lokale Zeitbedarf zur Abwicklung des Kommunikationsdienstes ist für sie abgrenzbar.

Während der Abwicklung des Commit-Protokolls sind die Reaktionsmöglichkeiten einer Subtransaktion auf ein Minimum beschränkt. Vor dem Übertritt in die Commit-Phase-1 besteht nun für eine Subtransaktion die Möglichkeit, durch Angabe ihrer lokalen Zeitbedingungen die Verantwortung für deren Überwachung an den Kommunikationsmanager zu übertragen. Dieser nimmt außer den lokalen Zeitanforderungen auch die Zeitanforderungen wahr, die ihm von den Kommunikationsmanagern der direkten Nachfolger im Transaktionsbereich zugestellt werden. Er übergibt mit seiner eigenen Bereitschaftsmitteilung zum Transaktionsabschluß auch die Verantwortung für die Überwachung der ihm anvertrauten Zeitanforderungen an seinen Vorgänger weiter. Auf diese Weise liegt die Verantwortung für die Kontrolle der Zeitbedingungen immer in der Hand eines Kommunikationsmanagers, der noch die Möglichkeit zum Transaktionsabbruch besitzt. Bei Verletzung einer Zeitbedingung kann so rechtzeitig der Transaktionsabbruch eingeleitet werden, der zum einen den ACID-Rules genügt, zum anderen den korrekten zeitlichen Ablauf ermöglicht.

Neben dem Transaktionsabbruch sind nur dann weitere Reaktionen des Kommunikationsmanagers möglich, wenn die lokale Subtransaktion noch nicht in die Commit-Phase-1 eingetreten ist. In diesem Fall wäre die Signalisierung

der Verletzung einer Zeitbedingung gegenüber der lokalen, nicht sinnvollen. Diese kann zwar nicht mehr mit der Zeitbedingung kollidieren, die die Bedingung gesetzt hat, aber sie hat die Möglichkeit, Aktionen aus der Menge der direkten Nachfolgerprozesse, die vorzeitig zur Commit-Vorbereitung aufgefordert hat, in die anwesende, spezifische Reaktion einzubeziehen.

Der Vorgängerprozeß akzeptiert nach Vorgabe des TP-Standards nur die Nachricht der Bereitschaft zum Commit oder die Rollback-Aufforderung. Eine Erweiterung des TP-Protokolls, welche die ungehinderte Erweiterung einer verletzten Zeitbedingung bis zum Wurzelknoten des Transaktionsbereiches erlaubt, würde eine erweiterte, aber nicht notwendigerweise rechtzeitige, Konfliktbehandlung ermöglichen. Eine solche Maßnahme hat keine Auswirkung auf die Gewährleistung der ACID-Eigenschaften.

Die zeitlich korrekte Abwicklung der Commit-Phase wird durch Synchronisierung des von den einzelnen Subtransaktionen angegebenen Zeitbedarfs während der Commit-Phase-1 unterstützt. Hierbei spielt das in der Echtzeitverarbeitung geforderte vorhersehbare Zeitverhalten eine entscheidende Rolle. Die Abschätzung des maximalen Zeitbedarfs für die Bearbeitung der Commit-Phase-2 einer Subtransaktion setzt definiertes Zeitverhalten des Subsystems voraus. Der so ermittelte Zeitbedarf läßt sich, mit den gesammelten Zeitanforderungen der Subtransaktionen für die Commit-Phase-2 vergleichen. Bereits in der Commit-Phase-1 wird der Vergleich zwischen dem ermittelten Zeitbedarf und den auf die Commit-Phase-2 bezogenen Zeitanforderungen durchgeführt. Die Verträglichkeit von Zeitbedarf und Zeitanforderungen deutet auf den rechtzeitigen Transaktionsabschluß hin.

Die Erkennung eines auf die Commit-Phase-2 bezogenen Zeitkonfliktes erfolgt durch einen Kommunikationsmanager, der alle für die Commit-Phase-1 bereits diskutierten Reaktionsmöglichkeiten hat.

Der vorgestellte Ansatz erfüllt somit die Forderung, die zeitlich korrekte Abwicklung von verteilten Transaktionen zu unterstützen. Die vom TP-Standard vorgegebene Baumstruktur für einen Transaktionsbereich enthält keine redundanten Verbindungen. Maßnahmen, die das zeitlich korrekte Verhalten beim Ausfall von Verbindungen oder Subsystemen ermöglichen, sind in dem Ansatz nicht berücksichtigt. Die Erweiterung der Kommunikations-

struktur zu einem gerichteten azyklischen Graphen (siehe Kapitel 2.1.1) erlaubt den Aufbau redundanter Verbindungen und trägt so zur Ausfallsicherheit bei.

8.8 Grenzen des Ansatzes

Der vorgestellte Ansatz zur Kontrolle der Zeitanforderungen während des Commit-Verfahrens beruht auf einer Worst-Case-Analyse des zeitlichen Verlaufs. Die Abweichungen, die sich aus den Schwankungen der lokalen Uhren und dem unsicheren Übertragungsverhalten ergeben, addieren sich in jedem Prozeß. Bei der Übergabe der zeitbezogenen Informationen zwischen zwei Prozessen ergibt sich eine Multiplikation des Fehlers. Dadurch wird in vielen Fällen ein Abbruch der Transaktion herbeigeführt, der unter der Kenntnis der tatsächlichen Zeiten vermeidbar wäre. Eine umgekehrte Abschätzung führt zur positiven Entscheidung, solange noch eine geringe Chance für den rechtzeitigen positiven Abschluß der Transaktion besteht.

Für die Fehlerabschätzung bei der Bewertung der zeitbezogenen Daten sind aufgrund der Reduktion der zeitbezogenen Daten in jedem Prozeß nur die Schwankungen der Uhren und der Übertragungszeiten zu berücksichtigen, die auf einem Pfad durch den Transaktionsbereich von dem Wurzeltransaktionsprozeß bis zu einem definierten Blatt auftreten. Je nach der betrachteten Information sind unterschiedliche Pfade in die Fehlerabschätzung einzubeziehen.

Die Durchführung dieses Verfahrens setzt voraus, daß alle an der Transaktion beteiligten Prozesse korrekte Angaben über den maximalen Zeitbedarf für ihr Commit-Phase-2-Intervall machen.

Die Kontrolle der Zeitbedingungen eines Teilbaumes im übergeordneten Prozeß geht davon aus, daß dieser bereits vor dem Eintreffen des TP-Commit-Req die Verletzung der Zeitbedingungen feststellen kann. Wenn diese Annahme nicht gilt, muß der überwachende Prozeß trotz der Kenntnis einer verletzten Zeitbedingung seine Ready-RI-PDU nach oben weiterleiten. Erst der Vorgänger kann auf die verletzte Zeitbedingung reagieren. Dies kann trotz intakter Verbindungen und Prozesse ein verspätetes Eintreffen der negativen Entscheidung bei dem Transaktionsprozeß mit der härtesten Zeitbedingung

zur Folge haben. Diese Situation wird nur durch das im TP-Standard definierte Verhalten für einen Kommunikationsmanager provoziert und ist zur Erhaltung der ACID-Eigenschaften nicht unbedingt notwendig.

Eine ähnliche Situation kann auch dann entstehen, wenn einer der Transaktionsprozesse für eine Verbindung zu einem Nachfolger kein rnrgezogenes Prepare eingeleitet hat. Er gibt damit frühzeitig seine Entscheidungsfähigkeit auf und ist daher nicht mehr bereit, die Verantwortung für f, bcrwachung der Zeitbedingungen seiner Nachfolger zu übernehmen.

Wenn ein Transaktionsprozeß nach dem Empfang der Ready-RI-PDU eines Nachfolgers und vor dem Absenden der eigenen Ready-RI-PDU an seinen Vorgängerprozeß ausfällt, kann er die Überwachung für den Nachfolger nicht mehr garantieren. Der Vorgängerprozeß kennt die Zeitanforderungen noch nicht und kann deshalb nicht darauf reagieren. Eine solche Reaktion würde den betroffenen Nachfolgerprozeß auch nicht erreichen. Ein zeitbedingter Abbruch der Transaktion orientiert sich nicht an der Zeitbedingung; des betroffenen Nachfolgerprozesses.

Die Blockierung von Transaktionsprozessen beim Eintreten einer Partitionierung des Transaktionsbereiches läßt sich nicht verhindern. Ein Transaktionsprozeß, der während einer Wartesituation die Verkrtzung einer Zeitbedingung bemerkt, kann nur darauf schließen, daß entweder eine Partitionierung des Transaktionsbereiches rnrliegt oder mindestens eine der Zei(angaben nicht korrekt war.

9 Realisierung eines ISO-TP-Kommunikationsmanagers für zeitkritische Transaktionen

Die Ergebnisse aus der in den Kapiteln 5 und 6 durchgeführten Analyse des ISO-TP-Standards und seiner Einbettung als Kommunikationsmanager in die S11hsysteme ein verteilt<s Transa.ktionssystem bildeten die Grundlage für die f.ontwicklung des in Kapitel 8 beschriebern,n Ansa.tzcs. Mit der Realisierung „s Jnsatzs konnt< sPirw Trngf.ähigkeit demonstriP.rt werden.

Die Il<r<itstellung des ISO-TP-Dienstes Nfolgte auf der Grundlage des TP-frotokolls und nnk<r Berücksichtigung des CCR-Protokolls. Während der Dat.entransferphase ist der Austausch von Information unter Einsatz all der Übertragungsprotokolle möglich, deren zugehörige Dienste beim Aufbau der Dialoge ausgehandelt wurden. Die Übermittlung von Daten des Implementierungskonzeptes während der beiden Commit-Phasen ist in den Protokollbeschreibungen für TP und CCR nicht vorgesehen. Eine Erweiterung, die die Übertragung von zeit.bezogenen Daten während der Commit-Phase-1 regelt, kann deshalb nicht ohne Einfluß auf die Verträglichkeit mit dem TP- und dem CCR-Standard eingeführt werden.

Die Realisierung des ISO-TP-Kommunikationsmanagers erfolgte unter dem Anspruch, die Übermittlung zeitbezogener Daten als eigenständiges Protokoll zu verwirklichen. Damit wird die Wahl zwischen einer standardkonformen CCR- und TP-Protokollabwicklung und der durch das Übertragungsprotokoll für Zeitinformation erweiterten Variante möglich. Die dazu notwendigen Voraussetzungen wurden durch lokal begrenzte Behandlung von Zeitvorgaben geschaffen.

Zunächst wird in Kapitel 9.1 die interne Struktur des implementierten Kommunikationsmanagers und seine Einbettung in ein Subsystem geschildert. Auf die lokale Behandlung von Zeitvorgaben und die Realisierung des Protokolls zur Übermittlung und Auswertung zeitbezogener Daten wird in den darauffolgenden beiden Abschnitten 9.2 und 9.3 eingegangen.

9.1 Entwicklung und Einbettung des Kommunikationsmanagers

Für die Beschreibung der Grobstruktur des implementierten Kommunikationsmanagers ist die Klärung des Begriffes "Systemprozess", der bereits in Kapitel 1 eingeführt wurde, Begriffen "Transaktionsprozess", "Anwendungsprozess" und den darauf aufbauenden "Nichtvorgängerprozess" und "Vorgängerprozess" notwendig.

Systemprozess: Unter dem Begriff "Systemprozess" wird ein Prozess des Basisbetriebssystems eines Subsystems, z.B. Unix oder VM, verstanden.

Der Begriff des Transaktionsprozesses bzw. des Anwendungsprozesses bezeichnet dagegen eine Komponente des im TP-Standard definierten Beschreibungsmodells. Die Zuordnung zwischen einem solchen Modellprozess und den realen Systemprozessen ist abhängig vom Implementationskonzept.

Der ISO-TP-Kommunikationsmanager wurde als Teilhabersystem konzipiert. Der Begriff "Teilhabersystem" ist in der DIN-Norm 3.1400 [inf88] festgelegt und läßt sich durch Gegenüberstellung mit dem Begriff des Teilnehmersystems charakterisieren. Ein Teilhabersystem zeichnet sich durch die Bereitstellung eines Systemprozesses für mehrere Benutzer aus. Im Unterschied dazu wird in einem Teilnehmersystem jedem Benutzer ein Prozess zugeteilt. Das für den Kommunikationsmanager gewählte Implementierungskonzept hat keinen Einfluß auf andere Resource-Manager oder den Transaction-Manager. Der Kommunikationsmanager wird durch einen Systemprozess realisiert, der für jede Subtransaktion eine TP-Protokollmaschine²¹ zur Verfügung stellt. Das eingesetzte Teilhaber-Konzept ermöglicht die Verwirklichung von Bearbeitungsstrategien bei der Unterstützung des Kommunikationsverhaltens konkurrierender Subtransaktionen und ihre Berücksichtigung bei der Berechnung des lokalen Zeitbedarfs im Kommunikationsmanager.

Für die Realisierung des Kommunikationsmanagers wurden grundlegende Funktionen zur Verwaltung von Anwendungsprozessen und ihren Dialogen

²¹Der Begriff *TP-Protokollmaschine (TPPM)* wurde in Kapitel 4 mit Abbildung 3 auf Seite 38 erläutert.

bereitgestellt. Dieser Funktionsumfang wurde um die Unterstützung von unverketteten Transaktionen ergänzt. Diese Erweiterung beinhaltet die Erstellung und Verwaltung von Transaktionsverbindungen und die Verwirklichung des Commit-Verfahrens zum positiven Transaktionsabschluß und des Rollback-Mechanismus zum Zurücksetzen einer Transaktion. Zur Entwicklung wurde auf die Spezifikationsmethode PASS [Fle84] und die Programmiersprache Pearl [Wer91] zurückgegriffen. In den beiden folgenden Abschnitten wird die in der Spezifikation festgelegte Grobstruktur des realisierten Kommunikationsmanagers und die Implementierung kurz vorgestellt.

9.1.1 Grobstruktur des realisierten Kommunikationsmanagers

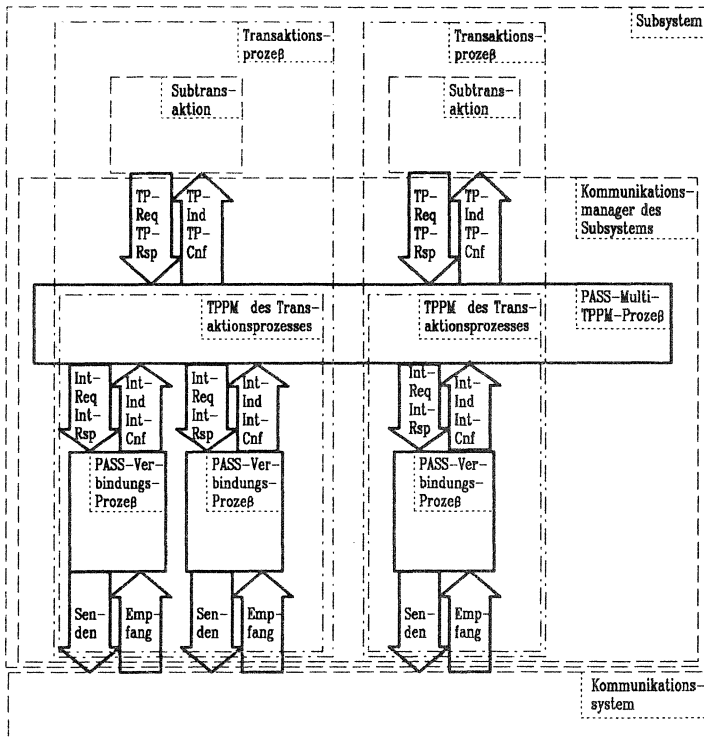
Die Anforderungen an einen Kommunikationsmanager bestehen in erster Linie darin, die im TP-Standard beschriebene funktionale Abgeschlossenheit für jede TPPM zu gewährleisten. Aussagen über gegenseitigen zeitlichen Einfluß der Abläufe verschiedener TPPMs sind darin nicht enthalten. Der Standard ist damit offen gegenüber beliebigen Bearbeitungsstrategien konkurrierender Subtransaktionen. In Kapitel 6 wurde die Modellierung von gegenseitigen zeitlichen Einflüssen der TPPMs verschiedener Transaktionsprozesse mit Zeitverzögerungen, die aus Aktivitäten anderer Systemprozesse resultieren, durch Einführung von Queuing-Aktivitäten zusammengefaßt. Die Implementierung kann jedoch nur die Bearbeitungsstrategie für Konkurrenzsituationen innerhalb des Kommunikationsmanagers vorgeben. Die Behandlung von Konkurrenzsituationen zwischen verschiedenen Systemprozessen, z.B. anderen Resource-Managern, und dem Kommunikationsmanager bleibt davon unberührt, geht aber ebenfalls in die Queuing-Aktivitäten ein.

Neben der funktionalen Unabhängigkeit verschiedener TPPMs wird im TP-Standard die Unabhängigkeit verschiedener Verbindungen eines Transaktionsprozesses gefordert. Die Unabhängigkeit der Verbindungen eines Transaktionsprozesses ist auf das Datentransferintervall beschränkt. Für die beiden Commit-Intervalle ergibt sich die Abhängigkeit der Verbindungen aus den durch den Commit-Dienst hervorgerufenen kausalen Zusammenhängen, die in den Abbildungen 14, 15 und 16 auf den Seiten 67, 68 und 69 dargestellt wurden.

PASS erlaubt die Spezifikation paralleler bzw. nebenläufiger Aktivitäten durch Einführung einer Prozeßstruktur. Jeder PASS-Prozeß beschreibt einen sequentiellen Ablauf. Parallele oder nebenläufige Aktivitäten werden unterschiedlichen PASS-Prozessen zugeordnet, die über Botschaften oder gemeinsamen Speicher kommunizieren können. PASS-Prozesse, die auf ein Monoprozessorsystem abgebildet werden, laufen, soweit die Kommunikation dies zuläßt, quasi-parallel ab. PASS-Prozesse, die auf verschiedenen Prozessoren oder Systemen realisiert sind, können parallel ausgeführt werden. Ein Überblick über die PASS-Prozeßstrukturierung und die Kommunikation der PASS-Prozesse wird durch eine PASS-Kommunikationsstruktur graphisch wiedergegeben.

In der Spezifikation des Systemprozesses zur Realisierung des Kommunikationsmanagers wurde die unabhängige Verwaltung jedes Dialoges durch Zuordnung eines PASS-Verbindungsprozesses dargestellt. Die Verwaltung der verschiedenen TPPMs wurde dagegen in einem PASS-Prozeß, dem Multi-TPPM-PASS-Prozeß zusammengefaßt. Durch die PASS-Verbindungsprozesse ist auch die unabhängige TPM-Verwaltung gewährleistet. Abbildung 21 zeigt vereinfacht die PASS-Kommunikationsstruktur des realisierten Kommunikationsmanagers und deutet die Einbettung in ein Subsystem durch Überlagerung des Modells konkurrierender Subtransaktionen mit gestrichelten Linien an. Die im Standard definierten Begriffe, TP-Protokollmaschine und Transaktionsprozeß, werden durch Strichpunktlinien angezeigt.

Die PASS-Verbindungsprozesse erhalten verbindungsspezifische Aufträge vom PASS-Multi-TPPM-Prozeß oder vom Kommunikationssystem. Jeder PASS-Verbindungsprozeß verarbeitet seine Aufträge nach der *first in first out* Strategie (FIFO). Die PASS-Verbindungsprozesse kommunizieren nicht untereinander und können daher vollständig nebenläufig arbeiten. Der PASS-Multi-TPPM-Prozeß erhält Aufträge von den Subtransaktionen oder von den PASS-Verbindungsprozessen. Er bearbeitet die Aufträge ebenfalls nach FIFO-Strategie. Die innerhalb eines PASS-Prozesses festgelegte Bearbeitungsstrategie von Aufträgen, wie z.B. hier die FIFO-Strategie, läßt keine dynamische Veränderung zu.



Komponente des Kommunikationsmodells konkurrierender Subtransaktionen mit Bezeichnung in der rechten oberen Ecke



Komponente des im ISO-11'-standard verwendeten Beschreibungsmodells mit Bezeichnung in der rechten oberen Ecke



Graphische Darstellung eines PASS-Prozesses in einer Pus-Kommunikationsstruktur mit Bezeichnung in der rechten oberen Ecke



Botschaftsübermittlung zwischen zwei PASS-Prozessen

Abbildung 21: Interne Strukturierung des Kommunikationsmanagers durch PASS-Prozesse

9.1.2 Implementation des Kommunikationsmanagers

Die Implementierung des Kommunikationsmanagers wurde in der Programmiersprache Pearl vorgenommen. Pearl ist eine auf Echtzeitverarbeitung zugeschnittene Programmiersprache, die auf einem prioritätsgesteuerten Multitaskingkonzept aufbaut und Konstrukte zur Zeitverarbeitung und zur Ausnahmebehandlung anbietet.

Die im vorangehenden Abschnitt durch kommunizierende PASS-Prozesse vorgestellte Grobstruktur des Kommunikationsmanagers läßt sich in Pearl durch Zuordnung je einer Task zu einem PASS-Prozeß verwirklichen. Die Unabhängigkeit der PASS-Verbindungsprozesse wird in Pearl durch das Multitaskingkonzept realisiert. Die Kommunikation zwischen den PASS-Prozessen wurde in der Implementierung durch Einführung eines Botschaftskonzeptes simuliert. Die Pearltasks tauschen Nachrichten über gemeinsame Speicherbereiche aus. Der abwechselnde Lese- und Schreibzugriff auf einen Speicherbereich wird durch Semaphore geregelt, die den kommunizierenden Tasks zugeordnet sind. Ereignisse, die an der Schnittstelle zum Kommunikationssystem bzw. an der TP-Schnittstelle eintreten und vom Kommunikationsmanager behandelt werden müssen, werden von einer Verbindungstask bzw. der Multi-TPPM-Task als Nachricht behandelt. Die Bearbeitung gleichzeitig vorliegender Nachrichten hängt von den Prioritäten der zuständigen Tasks ab. Pearl erlaubt eine dynamische Veränderung der Taskprioritäten.

Neben der Gewährleistung der Unabhängigkeit der Verbindungen liegt ein weiterer Grund für die Trennung der TPPM-Verwaltung und der Verbindungsverwaltung in der durch Pearl vorgegebenen statischen Speicherverwaltung. Die Realisierung einer TPPM mit allen zugehörigen Verbindungen durch eine Task gibt eine statische maximale Anzahl von Verbindungen für den zugehörigen Transaktionsprozeß vor. Durch Trennung von TPPM-Verwaltung und Verbindungsverwaltung wird eine dynamische Integration von Verbindungstasks in eine TPPM möglich. Die Anzahl der verfügbaren Verbindungen für einen Transaktionsprozeß richtet sich nur nach der maximalen Anzahl der im Kommunikationsmanager verwaltbaren Verbindungen, abzüglich der bereits für andere Transaktionsprozesse vergebenen Verbindungen.

Die Unterstützung zeitkritischer Transaktionsprozesse gegenüber Transaktionsprozessen ohne Zeitanforderungen läßt sich durch Einführung von zwei Prioritätsklassen für die Verbindungstasks erreichen. Eine Verbindung, die einem zeitkritischen Transaktionsprozeß zugeordnet ist, wird dann durch Zuteilung einer Verbindungstask aus der höheren Prioritätsklasse verwaltet.

Die Bevorzugung von zeitkritischen Transaktionsprozessen bei der TPPM-Verwaltung wäre ebenso wie für die Verbindungsverwaltung realisierbar, wenn die Multi-TPPM-Task durch mehrere Single-TPPM-Tasks ersetzt würde. In der vorliegenden Implementierung besitzt diese Task gegenüber den Verbindungstasks höhere Priorität. Dadurch wird eine gleichberechtigte Behandlung von Aufträgen verschiedener Verbindungen in der Multi-TPPM-Task erreicht.

Die Realisierung dieser beiden Vorschläge kann eine Beschleunigung zeitkritischer Transaktionsprozesse bewirken. Für die Vorhersagbarkeit des Zeitverhaltens ist diese Beschleunigung ohne Belang. Dafür muß vielmehr garantiert werden, daß die Prozessorauslastung für den Kommunikationsmanager als Systemprozeß in einer Strategie der Echtzeitplanung erfolgt. Außerdem muß gewährleistet sein, daß die Auslastung des Subsystems die Bearbeitung aller Aufträge in Echtzeit erlaubt. Durch die maximale Anzahl von Verbindungen und die maximale Anzahl unterstützbarer Transaktionsprozesse im Kommunikationsmanager ist für die Belastung durch Kommunikation eine Grenze vorgegeben.

9.2 Lokale Behandlung von Zeitvorgaben

Zur lokalen Behandlung von Zeitanforderungen einer Subtransaktion, die sich auf kommunikationsabhängige Zeiträume beziehen, ergeben sich aus der Perspektive des Kommunikationsmanagers zwei Möglichkeiten. Zum einen ist eine Zeitüberwachung des Kommunikationsmanagers denkbar. Zum anderen kann die Subtransaktion eine interne Zeitüberwachung durchführen. Für beide Fälle ergeben sich Anforderungen an den Kommunikationsmanager. Zur Erfüllung des ersten Ansatzes war die im nächsten Abschnitt beschriebene Erweiterung der TP-Schnittstelle notwendig. Die beiden realisierten Ansätze zur Unterstützung der lokalen Behandlung von Zeitanforderungen

„irwr Subtransaktion werdPn im Anschluß behandelt. Beide Ansätze sind unabhängig und können kombiniert werden. Zur Erläuterung der in den Abschnitten 9.2.1, 9.2.2 und 9.2.3 beschriebenen Ergänzungen wurde Abbildung 22 vorangestellt. Sie zeigt, eine erweiterte PASS-Kommunikationsstruktur, die die lokale Behandlung von Zeitvorgaben berücksichtigt.

Mit einer Interpretation der aus der lokal begrenzten Zeitbehandlung ableitbaren Ergebnisse schließt das Kapitel.

9.2.1 Erweiterung der TP-Schnittstelle

Die Formulierung von Zeitanforderungen durch die Subtransaktion bildet die Voraussetzung für die Übernahme der Zeitüberwachung im Kommunikationsmanager. Die TP-Schnittstelle wurde um ein Schnittstellenereignis *Timedata-Req* mit drei Parametern erweitert, welches der Subtransaktion die Formulierung folgender Zeitbedingungen erlaubt:

- Setzen einer oberen Grenze für die Dauer der Beteiligung an der Kommunikation innerhalb der Transaktion.
- Vorgabe eines Grenzzeitpunktes für das Ende der Beteiligung an der Kommunikation innerhalb der Transaktion.
- Angabe einer gewünschten Maximaldauer für die Abwicklung des Commit-Verfahrens im lokalen Transaktionsprozeß.

Mit den beiden ersten Parametern des Ereignisses *Timedata-Req* läßt sich eine Maximalzeitanforderung für einen Zeitraum angeben. Der Anfangszeitpunkt wird durch den Eintrittszeitpunkt des Ereignisses *Timedata-Req* vorgegeben. Der dritte Parameter bezieht sich auf die gewünschte Maximaldauer der lokalen Beteiligung am Commit-Verfahren und legt als Anfangszeitpunkt das Ereignis TP-Commit-Req fest. In allen drei Fällen entspricht der Endzeitpunkt des zu kontrollierenden Zeitraumes dem Ende des Commit-Phase-2-Intervalls für den betroffenen Transaktionsprozeß. Durch die Übermittlung einer Zeitanforderung von einer Subtransaktion an den Kommunikationsmanager wird implizit die Verantwortung für die Überwachung der Zeitbedingung übertragen.

Zur Abschätzung des lokalen Zeitbedarfs für das Commit-Phase-2-Intervall im Transaktionsprozeß wurde das Ereignis TP-Commit-Req um einen Parameter erweitert. Die Subtransaktion teilt dem Kommunikationsmanager darin ihren internen Zeitbedarf für die Commit-Phase-2-Bearbeitung mit. Diese Angabe ist zum einen davon abhängig, ob die Durchsetzung der Entscheidung in der Subtransaktion vor oder nach dem Abschluß der Kommunikationsaktivitäten²² erfolgt. Zum Anderen hängt der Zeitbedarf mit den in diesem Intervall auszuführenden Operationen zusammen. Der minimale Zeitbedarf ergibt sich aus der Protokollierung der globalen Entscheidung vor dem TP-Done-Req. Der maximale Zeitbedarf entsteht in einer Subtransaktion, wenn der zu bearbeitende Auftrag als Field Call²³ durchgeführt wird und der TP-Done-Req erst nach der Durchsetzung der globalen Entscheidung, d.h. am Ende der Bearbeitung, abgesetzt wird. Durch Angabe dieses anwendungsspezifischen Zeitbedarfs der Subtransaktion für das Commit-Phase-2-Intervall wird für den Kommunikationsmanager das lokale Zeitverhalten während dieses Zeitraumes abschätzbar.

9.2.2 Zeitüberwachung im Kommunikationsmanager

Mit der Übergabe des neu eingeführten TP-Schnittstelleneignisses erteilt die Subtransaktion dem Kommunikationsmanager den Auftrag, auf die Verletzung der angegebenen Zeitanforderungen mit einem provider-initiated-Rollback zu reagieren²⁴

Die Übergabe dieses Auftrages kann zu einem beliebigen Zeitpunkt während des Datentransferintervalles erfolgen. Er wirkt über die Grenze des Datentransferintervalls hinaus bis zu dem Zeitpunkt, an dem der Kommunikationsmanager sich entschließt, die lokale Bereitschaft zum Transaktionsabschluß in einer Ready-RI-PDU an den Vorgänger weiterzugeben²⁵.

²²d.h. vor oder nach dem TP-Done-Req

²³Die Ausführung des Auftrages beginnt erst dann, wenn die globale Entscheidung bekannt ist (GR93).

²⁴Für einen provider-initiierten Verbindungsabbruch aufgrund einer verletzten Zeitbedingung sind dem Kommunikationsmanager keine verbindungsbezogenen Ursachen bekannt. Deshalb erscheint als Reaktionsmöglichkeit nur das Zurücksetzen der Transaktion angebracht.

²⁵Ein Einfluß auf den Transaktionsverlauf zu diesem Zeitpunkt ist im TP-Standard nur

Aus der von der Subtransaktion übergebenen Zeitanforderung bestimmt die Multi-TPPM-Task einen Zeitpunkt bezüglich der lokalen Uhr, zu dem die Beteiligung des Transaktionsprozesses an der Kommunikation beendet sein soll. Die Überwachung der auf diesen Zeitpunkt fixierten Zeitanforderung wird von einer Timer-Task übernommen. Sie erhält von der Multi-TPPM-Task für jede Subtransaktion mit Zeitanforderung einen Auftrag. Die Aufträge werden in einer Liste nach Dringlichkeit sortiert. Der Zeitpunkt, der die kleinste Zeitanforderung repräsentiert, wird als Zeitbedingung für die Aktivierung einer Wecker-Task formuliert. Die Wecker-Task meldet der Timer-Task den Ablauf der Zeitbedingung. Die Timer-Task löscht daraufhin den Auftrag aus der Liste, sendet der Multi-TPPM-Task eine Nachricht und "stellt den Wecker" gemäß dem nächsten Eintrag in der Liste. Die Multi-TPPM-Task leitet aufgrund der verletzten Zeitbedingung einen provider-initiated-Rollback der Transaktion ein, der der lokalen Subtransaktion mit dem TP-Schnittstellenereignis TP-Rollback-Ind unmittelbar angezeigt wird. In Abbildung 22 ist die erweiterte PASS-Kommunikationsstruktur dargestellt, die die Timer-Task und die Wecker-Task enthält. Mit dieser Konstruktion wurde ein "guarded statement" simuliert, welches das gleichzeitige Warten der Multi-TPPM-Task auf TP-Schnittstellenereignisse, interne Ereignisse von den Verbindungstasks und dem Ablauf einer Zeitbedingung realisiert.

Ein Auftrag der Timer-Task bleibt solange bestehen, bis entweder die Zeitbedingung verletzt wird, oder die Stornierung des Auftrages von der Multi-TPPM-Task eintrifft. Wenn alle kausalen Bedingungen zum Erzeugen einer an den Vorgänger gerichteten Ready-RI-PDU erfüllt sind²⁶, führt die Multi-TPPM-Task eine Abschätzung bezüglich der Dauer des lokalen Commit-Phase-2-Intervalls durch. An diesem Punkt wird die Erweiterung des TP-Schnittstellenereignisses TP-Commit-Req, die Angabe des Zeitbedarfs der

für den Fall vorgesehen, daß die TPPM den Zustand ihrer internen Daten nicht halten kann. Die Verletzung einer Zeitanforderung der Subtransaktion wird dadurch nicht erfaßt. Nachdem die Ursache eines zu diesem Zeitpunkt eingeleiteten Rollback in anderen Subsystemen nicht erkennbar ist und er die ACID-Eigenschaften nicht gefährdet, wurde die Überwachung der Zeitanforderungen einer Subtransaktion und die Einleitung eines Rollback bis zu diesem Zeitpunkt vorgesehen.

²⁶d.h. der TP-Commit-Req der lokalen Subtransaktion wurde bearbeitet und die Ready-RI-PDUs von allen Nachfolgerprozessen im Transaktionsbereich wurden empfangen; siehe Abbildung 14, 15 und 16 auf den Seiten 67, 68 und 69.

Subtransaktion für die Cornrnit-Phase-2, benötigt. Zu diesem Zeitraum wird die vom Kommunikationsmanager benötigte Zeit für interne Aktivitäten während des Cornrnit-Phase-2-Intervalls hinzugefügt. Die gleichberechtigte Behandlung aller Verbindungen rechtfertigt die Annahme, daß die Aktivitäten im Kommunikationsmanager verbindungsunabhängig sind. Ihre Dauer geht als meßbare Konstante Ω die Berechnung ein. Der ermittelte Zeitraum wird der bis zum Ablauf der Zeitbedingung verbleibenden Restzeit gegenübergestellt. Wenn dieser Vergleich zu einem negativen Ergebnis führt, wird die Transaktion zurückgesetzt. Andernfalls wird eine Ready-RI-PDU erzeugt und an den Vorgänger weitergeleitet. In jedem Fall storniert die Multi-TPPM-Task den Tirner-Auftrag. Ein Auftrag der Tirner-Task bleibt solange bestehen, bis entweder die Zeitbedingung verletzt wird, oder eine Stornierung des Auftrages von der Multi-TPPM-Task eintrifft.

9.2.3 Begrenzung von kommunikationsabhängigen Zeiträumen durch die Subtransaktion

Unabhängig von der durch den Kommunikationsmanager bereitgestellten Zeitüberwachung bestehen für die Subtransaktion lokale Möglichkeiten der Zeitüberwachung. Abhängig von der konkreten Umgebung lassen sich in der Subtransaktion weitreichende zeitbezogene Informationen auswerten. Die Übertragung von zeitbezogenen Daten zwischen Subtransaktionen, die auf synchronisierte Uhren zurückgreifen können, läßt sich z.B. als *user data* in die Dienste TP-Begin-Dialogue bzw. TP-Begin-Transaction oder in beliebige TP-Data-Dienste integrieren.

Während des Datentransferintervalls kann eine Subtransaktion gezielt einzelne Verbindungen bzw. einzelne TP-Schnittstellenereignisse überwachen. Als Reaktion ist in diesem Intervall z.B. der Aufruf folgender TP-Dienste möglich:

TP-Data-Req: Reaktion auf der Basis eines anwendungsspezifischen Dienstes, der beim Aufbau des Dialoges ausgehandelt wurde.

TP-Error-Req: Reaktion durch Übermittlung einer Fehlermeldung.

TP-Request.-Control-Req: Anforderung der Sendeberechtigung. Dies ist nur für einen Dialog möglich, auf dem die Übertragungsrichtungen durch Übergabe einer Sendeberechtigung umgeschaltet wird.

TP-Rollback-Req: Zurücksetzen der Transaktion durch die Subtransaktion.

TP-Abort-Req: Abbrechen einer Transaktionsverbindung. Dieser Dienst leitet außerdem einen Rollback ein.

Für die TPPM besteht in diesem Intervall die Möglichkeit, den Datenfluß auf einer Transaktionsverbindung durch Konkatenation von aufeinanderfolgenden PDUs zu optimieren. Diese Optimierung wird in der vorliegenden Implementierung durchgeführt. Die PDUs werden in einen verbindungs-spezifischen Puffer eingelagert und solange verzögert, bis eine PDU erzeugt wird, die nicht konkatenierbar ist oder der Pufferplatz für die Aufnahme einer weiteren PDU nicht mehr ausreicht. Das erste Abbruchkriterium für das Ende der Verzögerung ist im TP-Standard definiert und kann von der Subtransaktion durch Aufruf geeigneter TP-Dienste herbeigeführt werden. Das zweite Kriterium, die Puffergröße, ist implementierungsabhängig. Für die Subtransaktion verschwimmen durch die konkatenationsbedingte Verzögerung von PDUs die Informationen über das Zeitverhalten ihrer Kommunikationspartner.

Die Einführung einer Pufferzeitbegrenzung als weiteres implementierungsabhängiges Kriterium für den Abbruch der Pufferung erlaubt der Subtransaktion die Abschätzung der Aufenthaltszeit einer PDU im Kommunikationsmanager. In der vorliegenden Implementation wird die maximale Aufenthaltsdauer einer PDU im Puffer durch eine Konstante festgelegt. Die individuelle Vorgabe einer maximalen Verzögerung im Puffer durch die Subtransaktion erfordert eine Erweiterung der TP-Schnittstelle. Durch Ergänzung aller Dienstaufrufe der TP-Schnittstelle um einen weiteren Parameter könnte die Subtransaktion für jede PDU eine individuelle maximale Verzögerung im Puffer vorgeben.

Die Pufferzeitbegrenzung wurde in Anlehnung an die im vorangehenden Abschnitt beschriebene Zeitüberwachung realisiert. Die Timer-Task erhält in diesem Fall von jeder Verbindungstask, die eine PDU in den Puffer einlagert, einen Auftrag zur Zeitbegrenzung. Die Auftragsverwaltung erfolgt

durch Eintrag in die Auftragsliste. Wenn ein solcher Auftrag das oberste Element der Liste bildet, wird die Weckertask für den angegebenen Zeitpunkt eingeplant. Sobald die Wecker-Task aktiv ist, sendet sie eine Nachricht an die Timer-Task. Diese löscht den Auftrag aus der Liste und meldet der auftragserteilenden Verbindungstask das Ende der Auftragsabwicklung. Die Verbindungstask leert daraufhin den Puffer und übergibt die bis dahin konkatenierten PDUs an das Kommunikationssystem. Wenn der Puffer aus einem der beiden anderen Gründe bereits vorzeitig geleert wurde, wird der Auftrag für die Timer-Task storniert.

Die Auswirkungen der Pufferzeitbegrenzung beschränken sich auf das Datentransferintervall. In den beiden Commit-Intervallen ist die konkatenationsbedingte Pufferung nicht zulässig. Im Commit-Phase-1-Intervall besteht für die Subtransaktion die Möglichkeit der Begrenzung der Wartezeit auf die globale Entscheidung durch Treffen einer heuristischen Entscheidung. Diese Entscheidung wird an der TP-Schnittstelle nur dann sichtbar, wenn die Subtransaktion im Commit-Phase-2-Intervall mit dem TP-Done-Req diese Information an den Kommunikationsmanager übergibt. Der Kommunikationsmanager unterstützt heuristische Entscheidungen nur durch Vergleich der lokalen Situation und der Informationen aller Nachfolger und zeigt gegebenenfalls dem Vorgänger-Transaktionsprozeß eine Mix-Situation an. Eine Lösung von Konfliktsituationen wird vom Kommunikationsmanager nicht unterstützt²⁷.

9.2.4 Bewertung der gewonnenen Information

Die durch lokale Maßnahmen gewonnene Information über das Zeitverhalten während des Transaktionsabschlusses läßt keine allgemeingültige Aussage über den lokalen Kommunikationsendzeitpunkt zu. Unter den Annahmen, daß

- der Zeitbedarf der lokalen Subtransaktion für die Bearbeitung der Commit-Phase-2 größer ist als die Dauer der Commit-Phase-2-

²⁷Eine heuristische Entscheidung ist nicht in jedem Fall durchführbar, siehe auch Fußnote 16 auf Seite 106. Sie wurde zur Vervollständigung der Möglichkeiten der Subtransaktion realisiert.

Bearbeitung des unterlagerten Teilbaumes einschließlich der Übertragungszeiten und

- die Zeitanforderungen einer Subtransaktion für den Abschluß der Beteiligung an der Kommunikation härter sind als die Zeitanforderungen der Nachfolgerprozesse,

ist die Entscheidung des Kommunikationsmanagers auf positiven Abschluß gerechtfertigt. Der fehlende Einblick in benachbarte Transaktionsprozesse verhindert eine zuverlässige Entscheidung, die die Einhaltung der Zeitanforderung bezüglich der Dauer des Commit-Phase-2-Intervalles bei fehlerfreiem Ablauf garantiert. Durch das im nächsten Kapitel vorgestellte Protokoll soll die fehlende Information über das Zeitverhalten benachbarter Transaktionsprozesse bereitgestellt werden.

Die aufgeführten Möglichkeiten zur lokalen Behandlung von Zeitanforderungen sind abhängig vom Transaktionskonzept unterschiedlich zu bewerten.

In einer unverketteten Transaktion kann jeder Transaktionsprozeß die Kommunikationsgrenzen der verteilten Transaktion für den unterlagerten Teilbaum durch die eigenen Kommunikationsgrenzen ermitteln. Das aus der Vereinigung von Datentransferintervall, Commit-Phase-1-Intervall und Commit-Phase-2-Intervall entstehende Kommunikationsintervall eines jeden Nachfolgerprozesses ist im lokalen Kommunikationsintervall enthalten. Für den Wurzelprozeß eines Transaktionsbereiches lassen sich daher die Kommunikationsgrenzen der verteilten Transaktion überblicken. Die erste Transaktion einer Transaktionskette ist in diesem Zusammenhang einer unverketteten Transaktion gleichzusetzen.

Die Kommunikationsphase eines an einer verketteten Transaktion beteiligten Transaktionsprozesses beginnt mit Ausnahme der ersten Transaktion einer Folge implizit mit dem Abschluß der vorangehenden Transaktion. Ein Transaktionsprozeß einer verketteten Transaktion kann deshalb die Kommunikationsphase seines unterlagerten Teilbaumes nur unter Einbeziehung von Ereignissen der vorangehenden Transaktion abschätzen. Die Kommunikationsintervalle aufeinanderfolgender Transaktionen können sich für verschiedene Transaktionsprozesse überlappen. Unter Einbeziehung des Empfangsereig-

nisses der globalen Entscheidung der vorangehenden Transaktion lassen sich die Kommunikationsgrenzen für den unterlagerten Teilbaum abschätzen.

9.3 Realisierung eines Protokolls zur prozeßübergreifenden Behandlung von zeitbezogenen Daten

Im folgenden wird ein TPTiming-Protokoll definiert, welches verteilte Transaktionen unterstützt, deren Subtransaktionen voneinander unabhängigen Zeitanforderungen unterliegen. Der in Kapitel 9.2 beschriebene lokale Dienst zur Behandlung von Zeitanforderungen einer Subtransaktion wird durch Unterlagerung eines Protokolls zur Realisierung externer Zeitüberwachung während der Commit-Abwicklung ergänzt. Folgende Zeitanforderungen werden für die Subtransaktion eines Transaktionsbereiches unterstützt:

- Die globale Entscheidung über den Abschluß der verteilten Transaktion wird spätestens zum Zeitpunkt t_1 erwartet.
- Der unterlagerte Teilbaum darf höchstens bis zum Zeitpunkt t_2 in die Kommunikation eingebunden sein.

Die beiden Anforderungen sind voneinander unabhängig und können kombiniert werden. Sie werden von einer Subtransaktion durch Erzeugen des in Abschnitt 9.2.1 vorgestellten Schnittstellenereignisses *Timedata-Req* an den Kommunikationsmanager übergeben.

9.3.1 Abwicklung des TPTiming-Protokolls

Das TPTiming-Protokoll regelt die Kodierung, Übertragung und Auswertung von zeitbezogenen Informationen, die zur Überwachung der oben angegebenen Zeitanforderungen notwendig sind. Zur Übertragung von Zeitinformation zwischen den Transaktionsprozessen wird eine TPTiming-RI-PDU eingeführt. Sie enthält vier Parameter:

'ff'Timing-RI-PDU

7,,itgrenz< für Eintrdfen der Entscheidung,
 homml1nikaf1onsendzeitpunkt im unterlagrcrten Teilbaum,
 (Dauer des Comrnit - Phase - 2 - Intervalls,
 Grad d<r TPTiming - Untr,rstützung.)

Di" heiden ersten Parameter kodieren die oben genannten Zeitanforderun-
 >Ph. Der dritt" Paramdrr gibt die gPschätzte Dauer des Commit-Phase-2-
 lrdrrvalls eines Transaktionsprozesses wieder. Der vierte Parameter zeigt an,
 ob TPTiming von allen Subtransaktionen und den zugeordneten Kommuni-
 kationsmanagern im gesamten unterlagerten Teilbaum unterstützt wird. Die
 Übermittlung dieser PDU ist an die Übertragung der Ready-RI-PDU von
 einem Transaktionsprozeß an seine Vorgänger gebunden. Durch Konkate-
 nation wird die **TPTiming-RI-PDU** der Ready-RI-PDU vorangestellt, sodaß
 jeder Transaktionsprozeß mit dem Empfang einer Ready-RI-PDU von ei-
 nem Nachfolgerprozeß auch dessen Zeitanforderungen und der voraussichtli-
 che P Z<itbedarf für das Commit-Phase-2-Intervall bekannt sind.

Durch die Auswertung einer empfangenen **TPTiming-RI-PDU** ergeben sich
 im l<ommunikationsma.na.ger Erkenntnisse über die Zeita.nforderungen aller
 Nachfolgerprozess. cks 11111,erstilzten Tra.nsa.ktiornnprozsses. Er übernimmt
 di. Vcrant.wortung fir di< C!lwrwachng diesPr Zc,itanforderung<"n solange, bis
 all- l!Pd11g11ng,n zm Erz11g1111g <11, Hedy-fU-PDlr für ckn lokalen Trans-
 aktionspro7f•ß <rfillt sinrl. W"" di,s- Sit11atio11 ohn< eirw zwischenzeitliche
 Vrrlffz11ng rirwr ,orlirgend<n 7,rit,rnfndernng rint.ritt., grneriPrt ckr Kom-
 mrmnikationsrnanagff ans **ckn** im Schnitt.stc•llenPreignis **Timcdata-Req** ent-
 haltenen Zeitinformationen der lokalen Subtransaktion und den ausgeWP.r-
 teten **TPTiming-RI-PDU**s eine neue **TPTiming-Rf-PDU**. Sie wird der an
 den Vorgängerprozeß gerichteten Ready-RI-PDU vorangestellt.

Durch den beschriebenen Ablauf werden die Zeitanforderungen des Trans-
 aktionsbereiches ausgehend von den Blattprozessen bis zum Wurzelprozeß
 gesammelt. In jedem Transaktionsprozeß findet eine Reduktion der Zeitinfor-
 mation statt. Ebenso wird der voraussichtliche Zeitbedarf für die Commit-
 Phase-2 von den Blattprozessen bis zum Wurzelprozeß hochgerechnet.

Der Übergabezeitpunkt der Verantwortung für die Überwachung der Zeitan-

forderungen eines Transaktionsprozesses an seinen Vorgängerprozeß wurde so gewählt, daß er sich nahtlos an die lokalen Maßnahmen zur Zeitbegrenzung anfügt.

9.3.2 Konfliktlösung für das TPTiming-Protokoll

Für die Anbindung des TPTiming-Protokolls all das TP-Protokoll ist zu prüfen, ob dadurch neue Blockierungssituationen entstehen können. Drei Fälle sind zu unterscheiden:

- Ein Transaktionsprozeß lehnt den Dialogaufbau mit der Begründung ab, daß TPTiming von seinem Kommunikationsmanager nicht unterstützt wird.
- Eine Subtransaktion setzt an ihren Kommunikationsmanagereinen TP-Commit-Req ab, ohne ihn vorher durch einen *Timedata-Req* über den lokalen Zeitbedarf im Commit-Phase-2-Intervall und ggf. vorliegenden Zeitanforderungen zu unterrichten.
- Ein Transaktionsprozeß sendet an seinen Vorgängerprozeß eine TP-Ready-RI-P_DU und stellt keine *TPTiming-RI-PDU* voran.

In der vorliegenden Implementierung wird auf die Dialogaufbauablehnung mit einem wiederholten Aufbauversuch ohne Angabe des TPTiming-Protokolls reagiert. Die Abwicklung der verteilten Transaktion wird durch die fehlende Unterstützung für TPTiming nicht verhindert. Dies hat zur Folge, daß der Transaktionsbereich einen Teilbaum enthält, der vom TPTiming nicht erfaßt wird. Deshalb wurde in der *TPTiming-RI-PDU* ein Parameter vorgesehen, der anzeigt, ob für alle Transaktionsprozesse eines Teilbaumes TPTiming unterstützt wird. Dieser Parameter relativiert die mit der *TPTiming-RJ-PDU* eintreffenden Informationen.

Wenn ein Transaktionsprozeß erkennt, daß mindestens ein Transaktionsprozeß seines unterlagerten Teilbaumes keine TPTiming-Unterstützung erhält, ist eine Gegenüberstellung von ermitteltem Zeitbedarf für die Commit-Phase-2 und den Zeitanforderungen bezüglich dem Abschluß der Kommunikation im unterlagerten Teilbaum ohne Bedeutung. Zudem muß angenommen werden,

daß in einer solchen Situation nicht alle Zeitanforderungen der Subtransaktionen des unterlagerten Teilbaumes vorliegen.

Das Verhalten bei Ablehnung eines TP-Dialoges mit TPTiming wird der dialoganfordernden Subtransaktion überlassen. Nach dem Dialogaufbau ohne TPTiming stehen ihr alle durch TP-Dienste bzw. andere ausgehandelte Dienste angebotenen Reaktionsmöglichkeiten zur Verfügung. Insbesondere kann sie die Transaktion zurücksetzen.

Denkbar wäre auch eine Lösung, die die Kommunikationsmanager anderer Transaktionsprozesse einbezieht. Ein Kommunikationsmanager, der eine **Timing-RI-PDU** empfängt und erkennt, daß nicht alle Transaktionsprozesse seines unterlagerten Teilbaumes TPTiming-Unterstützung erhalten, kann einen provider-initiated-Rollback einleiten. Dieses Verhalten des Kommunikationsmanagers könnte auch beim Eintritt in das Commit-Phase-1-Intervall durch die Subtransaktion angeregt werden. Dazu wäre eine Erweiterung des TP-Schnittstellenereignisses TP-Commit-Req um einen entsprechenden Parameter nötig.

Für die in Kapitel 9.2 vorgestellte lokale Behandlung von Zeitanforderungen einer Subtransaktion im Kommunikationsmanager wurde das Schnittstellenereignis **Timedata-Req** angeboten. Eine Subtransaktion, die dieses Ereignis nicht generiert, nimmt die angebotene Dienstleistung nicht wahr. Im Zusammenhang mit lokaler Zeitbehandlung hat dies keinen Einfluß auf andere Subtransaktionen. Für die Abwicklung des TPTiming-Protokolls ist die Beteiligung aller Subtransaktionen dagegen notwendig. Eine Subtransaktion, die keine Information über ihr Zeitverhalten im Commit-Phase-2-Intervall an den Kommunikationsmanager übergibt, verhindert die Abschätzung der Dauer des Commit-Phase-2-Intervalls für ihren Vorgängerprozeß und damit für alle ihr übergeordneten Transaktionsprozesse. Die Gegenüberstellung von Anforderungen bezüglich des Kommunikationsendzeitpunktes des unterlagerten Teilbaumes mit dem Zeitbedarf für die Commit-Phase-2 ist deshalb nicht mehr aussagekräftig. Sie wird in den Entscheidungsprozeß vor der Weiterleitung einer Ready-RI-PDU an den Vorgängerprozeß bzw. vor der Ermittlung der globalen Entscheidung im Wurzelprozeß nicht einbezogen.

In dieser Situation werden nur noch die Zeitanforderungen eingehalten, die sich auf das Eintreffen der globalen Entscheidung beziehen. Der Zeitin-

formationsfluß ist baumaufwärts gerichtet, Nur die Transaktionsprozess, zwischen dem Transaktionsprozess, der die "schweigsame Suht.ransaktion" enthält, und dem Wurzelprozess erfahren von der wduzierten TPTiming-Auswertung. Ebenso wie im Fall der Ablehnung einPs Dialoges mit TPTiming wäre auch hier eine durch die Subtransaktionen ang.,regte Vorgabe für das Verhalten der Kommunikationsmanager denkbar. Durch das Einleiten eines provider-initiated-Rollback können die vorliegenden Zeitanforderungen eingehalten werden.

Wenn ein Kommunikationsmanager für einen Transaktionsprozess trotz der Zustimmung zum TPTiming beim Dialogaufbau der Ready-RI-PDU keine *TPTiming-RI-PDU* voranstellt, gehen für den Vorgängerprozess wesentliche Zeitinformationen verloren. Diese Situation entspricht der erstgenannten, in der bereits beim Dialogaufbau bekannt wird, daß ein Transaktionsprozess keine TPTiming-Unterstützung erhält.

In allen drei Fällen wurde eine Konfliktlösung vorgeschlagen, die keine Wartesituationen hervorruft. Das TPTiming-Protokoll verändert das Kommunikationszeitverhalten der verteilten Transaktion nur durch zusätzliche Berechnungen in den Kommunikationsmanagern und die Übertragung einer weiteren PDU pro Transaktionsverbindung. Durch die Konkatenation wird der zusätzliche Übertragungsaufwand gering gehalten.

9.3.3 Einbindung der TPTiming-Implementation in den Kommunikationsmanager

Zur Realisierung des TPTiming-Protokolls waren Erweiterungen der Multi-TPPM-Task und der Verbindungstasks notwendig. Die Zusammenführung aller von einem Transaktionsprozess zu berücksichtigenden Zeitanforderungen und Zeitbedarfsangaben stellt eine verbindungsübergreifende Aufgabe dar und wird daher in der Multi-TPPM-Task erbracht. Die Ermittlung der Sende- und Empfangszeitpunkte der PDUs, die die Berechnungsgrundlage für die in einer *TPTimin_q-RI-PDU* kodierten Informationen bilden, ist dagegen eine verbindungspezifische Aufgabe. Sie wird in den Verbindungstasks erbracht.

Die Parameter der *TPTimin_q-RI-PDU* werden entsprechend den in Kapitel

8 beschriebenen Verfahren kodiert. Als Referenzzeitraum für die Kodierung wird die Dauer zwischen dem Empfang der Prepare-RI-PDU und dem Senden der Iteady-RI-PDU herangezogen. Die maximale Zeitdauer bis zum Eintreffen der Entscheidung bzw. zum Abschluß der Kommunikation im unterlager-ten Teilbaum werden als Verhältniswerte zum Referenzzeitraum angegeben. Die Voraussetzungen für bekannte maximaler und minimaler Übertragungszeit, n und der Grenzfunktionen der lokalen Uhren gehen in die Berechnung und Auswertung der Parameter der TPTiming-RI-PDU mit ein.

Außerdem wird in die *TPTiming-RI-PDU* das erwartete Zeitverhalten des Kommunikationsmanagers für die Abwicklung des Commit-Verfahrens einbezogen. Die Implementierung muß daher für die beiden Commit-Intervalle definiertes Zeitverhalten garantieren. In der vorliegenden Implementierung sind alle Abläufe innerhalb einer Task sequentiell und gehen daher als Konstante in die Berechnungen ein. Bei der Nachrichtenübermittlung zwischen der Multi-TPPM-Task und einer Verbindungstask darf die Nachricht nur eine definierte Zeit ruhen. Dieses Problem läßt sich durch Änderung der Prioritäten der Verbindungstasks lösen.

Die Weiterleitung der TPTiming-RI-PDU und der Ready-RI-PDU an den Vorgängerprozeß eines Transaktionsprozesses wird von der Multi-TPPM-Task initiiert. Sie erhöht die Priorität der betreffenden Verbindungstask so, daß sie höher als die eigene Priorität ist. Dadurch wird die unmittelbare Fortsetzung der Bearbeitung in der Verbindungstask erreicht. Nach dem Absenden der beiden PDUs durch die Verbindungstask setzt diese ihre eigene Priorität zurück. Beim Eintreffen der Entscheidung (Empfang der Commit-RI-PDU) in derselben Verbindungstask wird sie nach allen verbindungs-spezifischen Aktionen an die Multi-TPPM-Task weitergereicht. Die Behandlung wird dort ohne Unterbrechung fortgesetzt. Die Vervielfältigung der Entscheidung zur Weiterleitung an die Nachfolgerprozesse erfolgt in der Multi-TPPM-Task. Sie erhöht die Prioritäten aller Verbindungstasks, die den Dialogen zu den Nachfolgerprozessen des aktuell unterstützten Transaktionsprozesses zugeordnet sind. Es bietet sich dabei an, die Prioritäten so zu vergeben, daß der Nachfolgerprozeß mit dem größten Zeitbedarf zuerst bedient wird. Nach dem Absenden der Entscheidung nimmt jede Verbindungstask ihre Priorität wieder zurück. Das Zurücksetzen der Priorität bewirkt,

daß alle Verbindungen min wieder gleichberechtigt verwaltet werden. Durch eine Ordnung der Prioritäten nach einer Strategie der Echtzeitplanung könn-, eine bevorzugte Bedienung der auf die Commit-Bestätigung wartenden V.,r. bindungstasks erreicht werden. Wenn die Bestätigungen ültr di. K"nnntnis der globalen Entscheidung (Empfang der Commit-RC-PDU) von allen Nachfolgerprozessen eingetroffen sind und die Bestätigung der lokalen Subtransaktion (TP-Done-Req) vorliegt, initiiert die Multi-TPPM-Task die Weiterleitung der Bestätigung an den Vorgängerprozeß und erhöht die Priorität der zustäl:\digen Verbindungstask. Diese nimmt unmittelbar anschließend alle notwendigen verbindungs-spezifischen Operationen vor, die das Absenden der Commit-RC-PDU an den Vorgängerprozeß enthalten und 'nimmt ihre Priorität wieder zurück.

Mit dem beschriebenen Vorgehen wird gewährleistet, daß in der Commit-Abwicklung keine unbestimmten Verzögerungen bei der Nachrichtenübermittlung zwischen Multi-TPPM-Task und den Verbindungstasks entstehen.

9.3.4 Bewertung des TPTiming-Protokolls

Das TPTiming-Protokoll realisiert ein Verfahren zur schwachen Synchronisation der Transaktionsprozesse eines Transaktionsbereiches. Durch das zwei-Phasen-Commit-Protokoll wird eine kausale Ordnung von Kommunikationsereignissen vorgegeben. Aus der Einordnung dieser Kommunikationsereignisse in die einem Transaktionsprozeß zugängliche lokale Zeitachse werden Informationen über Bearbeitungszeiträume in den Nachfolgerprozessen gewonnen. Für einen Transaktionsbereich, dessen Kommunikationsmanager auf stark synchronisierte Uhren zurückgreifen, läßt sich die Übermittlung der zeitbezogenen Daten vereinfachen.

Für den Zeitbedarf bezüglich der Commit-Phase-2 führt der zur Fehlerbewertung betrachtete Pfad in jedem Transaktionsprozeß zu dem Nachfolgerprozeß, der den höchsten Zeitbedarf angemeldet hat. Für die Zeitanforderungen verläuft der Pfad zur Fehlerbewertung von einem Transaktionsprozeß zu dem Nachfolger, der die härteste Zeitanforderung aufgestellt hat. Diese Pfade müssen nicht identisch sein und entsprechen nicht unbedingt dem Pfad mit den meisten Transaktionsprozessen. In die Fehlerabschätzung eines Zeitda-

tums gehen die Schwankungen der Uhren und der Übertragungszeiten multiplikativ ein. Die Abweichung der abgeschätzten Zeitdaten von ihren realistischen Werten hängt von der Anzahl der Transaktionsprozesse auf den entsprechenden Pfaden und den verbindenden Dialogen ab.

10 Ausblick

Im Rahmen dieser Arbeit wurde der ISO-TP-Dienst auf seine Tauglichkeit für den Einsatz in der verteilten Transaktionsverarbeitung unter Echtzeitbedingungen geprüft. Als Ergebnis zeigte sich, daß

- die TP-Schnittstelle keinen Transfer zeitbezogener Informationen vorsieht,
- das Verzögern einer zu sendenden Nachricht durch den Kommunikationsmanager während der Datentransferphase zulässig ist und keine maximale Aufenthaltsdauer vorgesehen ist und daß
- die während der Abwicklung des 2-Phasen-Commit-Protokolls entstehenden Wartezeiten auf Nachrichten nicht vorhersehbar sind und daher die Echtzeitverarbeitung einer verteilten Transaktion verhindern.

Die Bereitstellung von Kommunikationsmechanismen, die die Berücksichtigung von verteilten Zeitbedingungen ermöglichen, hatte die zeitliche Eingrenzung einer verteilten Transaktion bzw. zeitkritischer Abschnitte der einzelnen Subtransaktionen einer verteilten Transaktion zum Ziel. Das 2-Phasen-Commit-Protokoll kommt innerhalb jeder positiv abzuschließenden verteilten Transaktion zum Einsatz und unterliegt daher auch den betrachteten Zeitbedingungen.

Aus den Erkenntnissen der Untersuchung ergibt sich, daß der TP-Standard keine echtzeitfähigen Kommunikationsmechanismen definiert. Die Ergebnisse flossen in den in dieser Arbeit vorgestellten Ansatz zur Unterstützung zeitkritischer verteilter Transaktionsverarbeitung ein. Die Anbindung des zur Realisierung des Ansatzes notwendigen Protokolls erhält die transaktionsunterstützende Funktionalität des ISO-TP-Protokolls in vollem Umfang.

Der Ansatz beruht auf der von den Blattknoten eines Transaktionsbereiches ausgehenden sukzessiven Konzentration der Zeitbedingungen, die mit der Übertragung der Verantwortung für deren Überwachung an den Wurzelknoten einhergeht. Die Möglichkeiten der Zeitbehandlung in einem verteilten System mit unsynchronisierten Uhren wurden weitgehend ausgeschöpft.

Ein Ziel war die Minimierung des neu hinzukommenden Informationsaustausches auf wenige Nachrichten. So wurde z.B. auf die Einführung der in [Bü89] angedeuteten Möglichkeit einer weiteren Transaktionsabschlußphase zur Übermittlung zeitbezogener Daten verzichtet. Ein weiteres Ziel bestand in der Gewährleistung einer zeitlich korrekten Abwicklung einer verteilten Transaktion. Daher wurde als Reaktion auf eine verletzte Zeitbedingung der Transaktionsabbruch ins Auge gefaßt. Unter der Voraussetzung intakter Verbindungen und intakter Rechner erreicht der Transaktionsabbruch die Subtransaktion, welche die verletzte Zeitbedingung vorgegeben hat, rechtzeitig.

Mit dem Transaktionsabbruch wurde die einzige Reaktionsmöglichkeit aufgegriffen, die alle an der verteilten Transaktion mitwirkenden Subtransaktionen erreicht. Die Signalisierung einer verletzten Zeitbedingung durch den Kommunikationsmanager läßt auch anwendungsspezifische Reaktionen zu, die dann durch die lokale Subtransaktion initiiert werden. Solche Reaktionen können jedoch nicht mehr alle Knoten des Transaktionsbereiches einbeziehen, da einige Subtransaktionen, insbesondere die Subtransaktion, deren Zeitbedingung verletzt wurde, bereits in die Commit-Phase-1 eingetreten sind und nur die Mitteilung des positiven oder des negativen Transaktionsabschlusses akzeptieren.

Die unterschiedliche Behandlung von harten und weichen Zeitanforderungen wurde in dieser Arbeit nicht weiter vertieft. Eine Verfeinerung dieser Klassifizierung könnte als Grundlage für die Vorgabe von anwendungsspezifischen Reaktionen dienen und die Einleitung eines Transaktionsabbruches nur als letzte Konsequenz auf die härteste Klasse von Zeitanforderungen beschränken. Hier sind weitere Untersuchungen notwendig, die auch Einfluß auf die geeignete Strukturierung der verteilten Anwendung haben.

Kommunikationsdienste, die unabhängig von der durch TP vorgegebenen Baumstruktur über eine eigene Kommunikationsstruktur realisiert werden, sind in dieser Arbeit nicht berücksichtigt. Als Beispiel sei hier die Abwicklung eines Synchronisationsprotokolles für die Uhren eines verteilten Systems über einen zentralen Zeit-Server genannt. Eine solche Konstellation schränkt derzeit den Kreis der verfügbaren Systeme zur Unterstützung einer verteilten Transaktion weiter ein, gewinnt aber im Zuge der fortschreitenden Er-

schließung von verteilten System mit Synchronisationsverfahren für Uhren an Bedeutung. Der Einsatz von Uhren, deren Gleichlauf durch ein Synchronisationsprotokoll erreicht wird, ist mit dem vorgestellten Ansatz verträglich.

Die Grenzen des Ansatzes zeigen sich beim Ausfall von Transaktionsverbindungen oder Rechnern des verteilten Systems. Der ISO-TP-Standard sieht hier Recovery-Maßnahmen vor, die nach der Reparatur der ausgefallenen Komponente einsetzen. Im Zusammenhang mit Echtzeitverarbeitung ist eine, z.B. durch den Ausfall einer Komponente verursachte, nicht definierbare Verzögerung nicht tragbar. Durch redundante Strukturen wird in Echtzeitsystemen definiertes Zeitverhalten garantiert. Die von ISO-TP vorgegebene Baumstruktur als Basis für die Kommunikation zwischen den in eine verteilte Transaktion eingebundenen Subtransaktionen enthält keine Redundanz. Der Ausfall einer Verbindung muß zwangsläufig zu einer Partitionierung der Baumstruktur führen. Eine Erweiterung der Kommunikationsstruktur zu einem gerichteten azyklischen Graphen mit einer definierten Wurzel wäre eine Möglichkeit zur Lösung solcher Situationen. Hier sind weitere Untersuchungen nötig, die zum einen den Aufbau redundanter Strukturen, zum anderen die Abwicklung des Commit-Protokolles auf diesen Strukturen betreffen. In [MSF83] wurde ein Ansatz vorgestellt, der die Existenz von Prozeß-Clustern innerhalb eines Transaktionsbereiches ausnutzt, um die Abwicklung des Commit-Protokolls zu beschleunigen. Im allgemeinen laufen die Prozesse einer verteilten Transaktion auf verschiedenen Rechnern ab, sodaß die Kommunikation nur über die Transaktionsverbindungen möglich ist.

Für die Gewährleistung von Echtzeitverhalten in einem verteilten System sind umfassende Analysen der entstehenden Konkurrenzsituation nötig. In [ADH91] wird ein Modell vorgestellt, welches sich mit der Ankunftsrate von Aufträgen und dem Queuing-Verhalten des Systems beschäftigt. Die vorliegende Arbeit setzt begrenztes Queuing-Verhalten für alle Rechner des verteilten Systems voraus. Damit wird die Prozessorzuteilung nach zeitlichen Kriterien gefordert. Von dem Queuing-Verhalten des Kommunikationssystems wurde abstrahiert. Das in Abschnitt 2.3 erläuterte Verfahren zur Ermittlung minimaler und maximaler Übertragungszeiten beinhaltet die durch Queuing bedingten Verzögerungen, liefert jedoch nur statistische Werte. Eine

weiterführende Untersuchung dieser Aspekte wäre auch für verteilte Transaktionsverarbeitung unter Berücksichtigung von Echtzeitanforderungen interessant.

vi3

Literatur

- (ACS88] „Service Definition for the Association Control Service Element (ACSE)“. ISO, Dezember 1988. IS 8649.
- (ADH91] David P. Anderson, Luca Delgrossi und Ralf G. Herrtwich. „Process Structure and Scheduling in Real-Time Protocol Implementations“. In: *Kommunikation in verteilten Systemen*, Informatik-Fachberichte, S. 83-95. Springer, Februar 1991. GI/ITG-Fachtagung.
- (ALS89] „Application Layer Structure“. ISO, März 1989. DIS 9545.
- (And89] Christian Andres. *Ein graphentheoretischer Ansatz zur rechnergestützten parallelen Komposition von Prozessen für verteilte Echtzeitsysteme*. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, Februar 1989. Fachbereich Informatik, Lehrstuhl für.
- (Apt85] Krzysztof R. Apt. „Real Time Clocks Versus Virtual Clocks“. *NATO AS/ Series*, F14:475-501, 1985 1985.
- (Bü89] Uwe Bürger. „A Flexible Two-Phase Commit Protocol“. *Computer Networks and ISDN Systems*, (17):175-185, 1989.
- (BW90] Alan Burns und Andy Wellings. *Real-Time Systems and their Programming Languages*. Addison-Wesley Publishing Company, 1990.
- (CCR89] „Commitment, Concurrency, Recovery (CCR), Service Definition DIS 9804.3, Protocol Specification DIS 9805.3“. ISO, September 1989.
- (CL85] K. Mani Chandy und Leslie Lamport. „Distributed Snapshots: Determining Global States of Distributed Systems“. *ACM Transactions on Computer Systems*, 3(1):63-75, Februar 1985.
- (Dan89] Jörg Dannenberg. „Entwicklung von Grundfunktionen für den ISO-Transaktionsdienst“. Studienarbeit, Friedrich Alexander Universität Erlangen-Nürnberg, Oktober 1989. Fachbereich Informatik, Lehrstuhl für Betriebssysteme.

- [DF89] Joseph D. Touch und David J. Farber. „Mirage: a Model for Ultra-High-Speed Protocol Analysis and Design". In: Harry Rudin und Robin Williamson, Hrsg., *Protocols for High-Speed Networks*, S. 115-133, Zürich, Mai 1989. University of Pennsylvania.
- [Dol86] Danny Dolev. „On the Possibility and Impossibility of Achieving Clock Synchronisation". *Journal of Computer and System Sciences*, 32:230-250, 1986.
- [FHK83] A. Fleischmann, P. Holleczeck und G. Klebes. „Synchronisation und Kommunikation verteilter Automatisierungsprogramme". *Angewandte Informatik*, 7, 1983.
- [Fle84] Albert Fleischmann. *Ein Konzept zur Darstellung und Realisierung von verteilten Prozeßautomatisierungssystemen*. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, April 1984.
- [GFE93] PEARL GI-Fachgruppe 4.4.2 Echtzeitprogrammierung. „PEARL90". Sprachreport Version 1.0, Gesellschaft für Informatik e.V., Bonn, Januar 1993.
- [GR93] Jim Gray und Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
- [Her91] Dr.-Ing R. G. Herrtwich. „Betriebsmittelvergabe unter Echtzeitsichtspunkten". *Informatik Spektrum*, 14:123-136, 1991.
- [Hof91] Prof. Dr. Fridolin Hofmann. *Betriebssysteme: Grundkonzepte und Modellvorstellungen*. Leitfäden der angewandten Informatik. Teubner, Edition, 1991.
- [inf88] „Informationsverarbeitung". DIN 334000, November 1988. Teil 9.
- [JZ90] David B. Johnson und Willy Zwaenepoel. „Recovery in Distributed Systems Using Optimistic Message Logging and Checkpointing". *Journal of Algorithms*, (11):462-491, 1990.
- [Kar93] Frank Karden. „Verteilte Zeiten". *iX*, S. 144-151, Februar 1993.
- [Koe92] H. Koepetz. „Sparse Time versus Dense Time in Distributed Real-Time Systems". *IEEE*, 460-467, 1992.

- [Lam78] Leslie Lamport. „Time, Clocks, and the Ordering of Events in a Distributed System". *Communications of the ACM*, 21(7):558-565, July 1978.
- [Lam86] Leslie Lamport. *TEX: A Document Preparation System*. Digital Equipment Corporation, Addison-Wesley Publishing Company, Edition, September 1986. User's Guide & Reference Manual.
- [Las90] Jürgen Laskowsky. „Entwicklung einer Testumgebung für den ISO-Transaktionsdienst". Studienarbeit, Friedrich Alexander Universität Erlangen-Nürnberg, Juni 1990. Fachbereich Informatik, Lehrstuhl für Betriebssysteme.
- [LMC87] Thomas J. Leblanc und John M. Mellor-Crummey. „Debugging Parallel Programs with Instant Replay". *Transactions on Computers*, C-36(4):471-482, April 1987.
- [LSP82] Leslie Lamport, Robert Shostak und Marshall Pease. „The Byzantine Generals Problem". *ACM Transactions on Programming Languages and Systems*, 4(3):382-401, Juli 1982.
- [Mac88] Lothar Mackert. „Verteilte Datenbanken". Vorlesungsskript, Heidelberg, Sommersemester 1988.
- [MSF83] C. Mohan, R. Strong und S. Finkelstein. „Method for Distributed Transaction Commit and Recovery Using Byzantine Agreement Within Clusters of Processors". *ACM*, 1983.
- [MW86] Klaus Meyer-Wegener. *Transaktionssysteme*. Dissertation, Universität, Kaiserslautern, Juni 1986.
- [Nel81] Bruce J. Nelson. *Remote Procedure Call*. Dissertation, Carnegie-Mellon-University, Mai 1981.
- [Obe90] Andreas Oberwies. *Zeitstrukturen für Informationssysteme*. Dissertation, Universität, Mannheim, Juli 1990.
- [Ram93] Krithi Ramamritham. „Real-Time Databases". *Distributed and Parallel Databases*, 1:199-226, 1993.

- [ROS88] „Remote Operations, Part 1: Model, Notation, and Service Definition, Part 2: Protocol Specification". ISO, Dezember 1988. DIS 9072.
- [RP90] K. Rothermel und S. Papp. „Open Commit Protocols for the Tree of Processes Modell". *IEEE*, S. 236-244, 1990.
- [RTB89] Michel Rynal; Gerard Tel und Jean-Claude Bermond. *Distributed Algorithms*, Kapitel 3, S. 179-250. Elsevier Science Publishers B.V., North Holland, 1989.
- [Sim92] Klaus Simon. *Effiziente Algorithmen für perfekte Graphen*. B.G.Teubner, Stuttgart, 1992. ISBN 3-519-02940-5.
- [SLW89] S.Davidson, I. Lee und V. Wolfe. „A Protocol for Timed Atomic Commitment". *IEEE*, S. 199-206, 1989.
- [SM91] Reinhard Schwarz und Friedemann Mattern. „Detecting Causal Relationships in Distributed Computations: In Search of the Holy Grail!". Interner Bericht 215/91, Universität Kaiserslautern, November 1991. Fachbereich Informatik.
- [SSP85] Boleslaw Szymanski, Yuan Shi und Noah S. Prywes. „Synchronized Distributed Termination". *IEEE Transactions on Software Engineering*, SE-11(10):1136-1140, Oktober 1985.
- [Tan92] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall International, 1992.
- [Tin76] Gottfried Tinhofer. *Methoden der angewandten Graphentheorie*. Springer, 1976.
- [Top84] Rodney W. Topor. „Termination Detection for Distributed Computations". *Information Processing Letters*, 18:33-36, Januar 1984.
- [TPM90] „Distributed Transaction Processing, Part 1: Model". ISO, September 1990. DIS 10026.
- [TPP90] „Distributed Transaction Processing, Part 3: Protocol". ISO, September 1990. DIS 10026.

- [TPS90] "Distributed Transaction Processing Part 2: Servicing". ISO, September 1990. DIS 0026.
- [Wer91] Werum Datenverarbeitungssysteme GmbH, Erbstorfer Landstrasse 14, 2120 Lüneburg. f[i]A RE, 90 Serien/import, Nov<ml">r 1991 1 kg. 2.2.1/9111/FB.
- [YM92] Yong I Yoon und Song C. Moon. , Hdi;ihJ,, Translating the Iron-ssin', for Real-Time Distributed Database Systems". Microprocessors 1992. Microprogrammin_q, 31:61-66, 1992.

Index

- 2-Phasen-Commit-Protokoll, 21
- ACID-Eigenschaften, 16, 35
- ACID-Rules, 16, 35
- Aktivitat, 32
- Anwendungsprozeß, 36
- Atomicity, 16
- Bezugssystem, 30, 51
- Bezugszeitpunkt, 30
- byzantinisches
 - Einigungsverfahren, 28
- Commit-Initiator, 21, 23
- Commit-Koordinator, 21, 22
- Commit-Phase-1, 44
- Commit-Phase-1-Intervall, 68, 71
- Commit-Phase-2, 45
- Commit-Phase-2-Intervall, 68, 72
- Consistency, 16
- Datentransferintervall, 68, 69
- Datentransferphase, 44
- Dialog, 38
- Dialogbaum, 39
- Durability, 16
- Echtzeitanwendung, 24
- Ereignismenge, 69
- heuristische Entscheidung, 104
- internes Ereignis, 62
- Isolation, 16
- kausale Relationen, 31
- kausale Zeitstruktur, 31
- kausales Zeitintervall, 31
- Kommunikationsmanager, 18
- Kommunikationsphasen, 21, 22
- Kommunikationssystem, 18
- Konkatenation, 64
- Konkurrenz, 17
- logische Uhr, 27, 33
- lokale Uhr, 24, 90
- Nachfolger, 42, 43
- PDU, 62
- physikalische Uhr, 27
- Queuing, 76
- Rolle, 40
- schwache Synchronisation, 27
- starke Synchronisation, 28
- Startereignis, 32
- Subsystem, 18
- Synchronisationsverfahren, 24
- Terminierung, 19, 20
- Terminierungsabschnitte, 20
- TP-Bezugszeitpunkte, 51
- TP-Dialog, 38
- TP-Dienst, 36
- TP-Dienst-Primitiv, 36
- TP-Schnittstelle, 36
- TP-Service-Provider, 36
- Transaktion, 15, 35
- Transaktionsbereich, 40
- Transaktionskette, 46

Transaktionskette, 46
Transaktionsprozeß, 40
Transaktionssystem, 18
Transaktionsverbindung, 40

unterlagerter Teilbaum, 43
unkettete Transaktion, 47

Vektorzeit, 34
verkettete Transaktion, 46
Vorgänger, 42

Warten, 77

Zeitobjekt, 30

